

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СПИСОК-2013

МАТЕРИАЛЫ ВСЕРОССИЙСКОЙ НАУЧНОЙ КОНФЕРЕНЦИИ ПО ПРОБЛЕМАМ ИНФОРМАТИКИ

23–26 апреля 2013 г.,
Санкт-Петербург

Санкт-Петербург
2013

УДК 004(063)
С72

*Печатается по рекомендации
кафедры системного программирования
Санкт-Петербургского государственного университета*

С72 СПИСОК-2013: Материалы всероссийской научной конференции по проблемам информатики. 23–26 апр. 2013 г., Санкт-Петербург. — СПб.: "DDO", 2013 — 792 с.

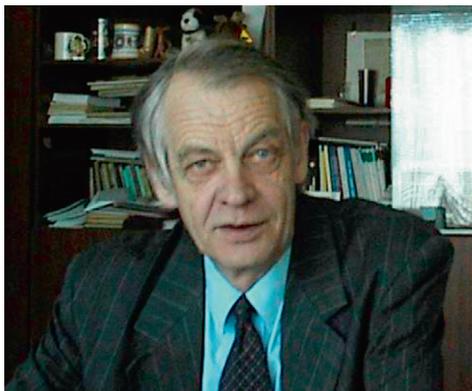
ISBN ; 9: /7/; 873/299; /9

Тематика сборника затрагивает широкий круг актуальных проблем теоретической прикладной математики и информатики. Слово «СПИСОК», ставшее названием конференции, — это не только обозначение фундаментальной структуры данных, но и сокращение от названий трех направлений исследований: Системное Программирование, Интеллектуальные Системы, Обеспечение Качества. Результаты исследований в этих областях являются значительной частью того знания, которое в настоящее время можно назвать словом «информатика».

Для студентов и аспирантов естественно-научных специальностей.

ISBN ; 9: /7/; 873/299; /9

© Санкт-Петербургский государственный университет, 2013
© Издательство DDO, 2013



Прекрасно, что конференция по информатике проходит в Питере.
Прекрасно, что она ориентирована на молодежь.
Однако, сделать конференцию полезной и интересной — очень трудная задача.

Желаю участникам и особенно организаторам удачи.

A handwritten signature in blue ink, consisting of stylized, cursive letters that appear to be 'В.П.' followed by a horizontal line.

академик **Иванников В.П.**

Директор Института системного программирования РАН
Зав. кафедрой системного программирования МФТИ
Зав. кафедрой системного программирования ВМК МГУ



Информационные технологии все шире проникают во все сферы производственной и общественной жизни. Здесь важно и умело ими пользоваться, и создавать новые подходы и решения. Без научного подхода эффективность и того, и другого будет невысокой. Научный подход заостряет внимание не столько на преимуществах существующих продуктов, сколько на ключевых трудностях и нерешенных задачах. Мало сказать, что решение лучшее (оптимальное, эффективное, быстрое и т. д.), надо еще это и доказать. Иначе все мы будем слепо следовать рекламе богатых фирм с установившейся репутацией, а новое будет пробивать себе дорогу очень трудно, возможно безуспешно. У вас есть шанс добиться признания своих замечательных идей если вы сумеете четко поставить задачу, обосновать и изложить решение, выстроить доказательную базу, подтвердить свои построения экспериментом, грамотно и содержательно опубликовать полученные результаты.

Санкт-Петербург был и остается одним из лидеров отечественной информатики и программной инженерии. Очень важно сохранять традиции лидерства и продолжать дело своих учителей. В рамках Лавровских чтений у молодых исследователей есть уникальная возможность заслушать доклады ученых более старшего поколения, познакомиться с новыми подходами и обобщениями.

Желаю успеха очередной конференции СПИСОК, а ее участникам — новых идей, нового творчества!

A handwritten signature in black ink, appearing to read 'Marchuk'.

Марчук А.Г.

Директор Института систем информатики им. А.П.Ершова СО РАН,
Заведующий кафедрой программирования, профессор НГУ

Системное программирование



**Терехов
Андрей Николаевич**

председатель программного комитета конференции
д.ф.-м.н., профессор
заведующий кафедрой системного программирования СПбГУ
директор НИИ Информационных технологий СПбГУ
генеральный директор ЗАО Ланит-Терком

ВОССТАНОВЛЕНИЕ ПАМЯТИ ВИРТУАЛЬНОЙ МАШИНЫ ИЗ РАСШИРЕННОГО ОБРАЗА ПАМЯТИ БАЗОВОЙ СИСТЕМЫ НА ПЛАТФОРМЕ WINDOWS

Овчинников А.А.

*студент 5 курса Математико-механического факультета СПбГУ,
anton.ovchi2nikov@gmail.com*

Аннотация: Важной частью компьютерного криминалистического анализа (digital forensics) является исследование образов памяти. Вместе с массовым распространением средств виртуализации появляется необходимость в новых методах анализа образов памяти, снятых с систем, на которых запущены виртуальные машины. В работе предложен подход к получению физической памяти гостевой системы из образа памяти базовой системы (host system), что позволяет в дальнейшем применить стандартные техники анализа памяти.

Введение

Основными задачами компьютерного криминалистического анализа является изучение вычислительных устройств и носителей информации в целях поиска и фиксирования доказательств (например, в ходе расследования гражданских или уголовных дел).

Типичный процесс экспертизы состоит из следующих основных шагов:

- **Захват (acquisition) образа данных на цифровом устройстве**
Результатом этого шага может быть образ жесткого диска, файловой системы, образ памяти и т. п.
- **Анализ данных**
Информация, представляющая интерес, может быть различной по виду и содержанию: история посещений веб-браузеров, журналы программ обмена мгновенными сообщениями, удаленные файлы различного типа и многое другое. Для извлечения и анализа данных могут применяться такие методы, как карвинг([3]) или стохастическое моделирование ([5]).
- **Подготовка отчёта о собранных артефактах**
По окончании этапа анализа формируется отчёт с подробным описанием всех использованных средств, методов и полученными результатами.

Использование расширенного образа памяти

Анализ образа памяти (memory forensics, live RAM forensics) в данный момент является важной частью криминалистического анализа. Образ (дамп) памяти захватывается на работающей системе, после чего анализируется. В памяти может быть найдено большое количество информации,

которую трудно или невозможно получить, располагая только жёстким диском: расшифрованные файлы или пароли, распакованные вредоносные программы, открытые в момент захвата образа сетевые подключения.

Особый интерес при исследовании процесса представляет его адресное пространство, восстановление которого позволяет приступить к более тщательному анализу. Но обладая исключительно образом памяти, значительная часть адресного пространства зачастую не может быть восстановлена, так как процесс может также использовать файлы подкачки и отображаемые в память файлы. Таким образом, некоторые части виртуального адресного пространства не хранятся в физической памяти операционной системы. Поэтому предлагается использовать т. н. «расширенный» образ памяти, то есть образ, дополненный файлом подкачки системы, а также исполняемым файлом и файлами динамических библиотек, которые использует процесс. Таким образом, появляется возможность восстановить значительную часть пользовательского адресного пространства процесса.

Исследование запущенных виртуальных машин

Современные технологии виртуализации позволяют создавать и использовать независимые системы, которые в то же время будут работать на одном физическом оборудовании. Гипервизоры, работающие на основе базовой (хостовой) ОС, также называемые гипервизорами «второго типа», позволяют запускать произвольные операционные системы внутри уже запущенных. Данные средства могут быть использованы злоумышленниками для создания «чистой», легко настраиваемой среды, в которой можно осуществлять преступную деятельность, не оставляя улик на хостовой системе. А используя зашифрованный раздел для хранения файлов виртуальной машины на жестком диске можно полностью скрыть ее наличие на компьютере.

Также средства виртуализации широко применяются в серверном сегменте. Использование исключительно офлайн-методов исследования (как анализ жестких дисков и журналов) в случае таких инцидентов как взлом и распространение вредоносного кода серьезно ограничивает эффективность анализа. А использованию анализа памяти в таких случаях препятствует как раз наличие технологий виртуализации, которые не позволяют применить стандартные техники и утилиты.

На данный момент ни одна из имеющихся утилит для работы с образами памяти не умеет извлекать информацию о виртуальных машинах, запущенных на хост-системе.

Предлагается осуществлять анализ образа памяти на наличие запущенных виртуальных машин, и в случае их наличия производить извлечение данных о гостевой системе. В частности, предлагается извлекать оперативную память гостевой системы для дальнейшего применения стандартных методов анализа памяти.

Восстановление физической памяти гостевой системы на примере QEMU

QEMU ([9]) — свободная программа с открытым исходным кодом для эмуляции аппаратного обеспечения различных платформ (x86, x86_64, MIPS, SPARC и т. д.).

Далее предполагаем, что виртуальное адресное пространство процессора QEMU было восстановлено из расширенного образа памяти системы Windows 7 x64, и дальнейшие действия будут производиться именно с восстановленным адресным пространством ([1]).

Структура RAMBlock

Различные фрагменты адресуемой памяти в QEMU хранятся в виде двусвязного списка `ram_list`, описываемого структурой `RAMList`. Каждый элемент в списке представляет собой структуру `RAMBlock` (её описание находится в файле `include/cpu-all.h` дерева исходных кодов QEMU), описывающую конкретный фрагмент памяти: это может быть и физическая память, и видеопамять, и системный BIOS.

```
typedef struct RAMBlock {
    struct MemoryRegion *mr;
    uint8_t *host;
    ram_addr_t offset;
    ram_addr_t length;
    uint32_t flags;
    char idstr[256];
    struct {
        struct RAMBlock *tqe_next;
        struct RAMBlock * *tqe_prev;
    } next;
} RAMBlock;
```

Рис. 1. Структура `RAMBlock`

Требуется найти структуру `RAMBlock` (см. Рис. 1), которая отвечает за отображение физической памяти.

После исследования исходного кода было установлено, что у блока, описывающего физическую память гостевой машины идентификатор `idstr` равен «`pc.ram`». Поэтому логичным шагом является поиск в образе памяти процесса данной строки. После нахождения смещения идентификатора, можно получить смещение предполагаемой структуры типа `RAMBlock`.

Получив очередное смещение, требуется проверка (валидация) того, что по данному смещению находится действительная структура `RAMBlock`.

Проверка потенциальных структур RAMBlock

Можно выделить несколько правил для валидации структур RAMBlock:

1. Поле `mg` должно являться действительным ненулевым указателем в адресном пространстве пользователя процесса. То есть для Windows на платформе $\times 64$ должно выполняться соотношение $0 < mg < 0 \times 8000'0000'0000$ при общем размере поля в 8 байт.

2. Для поля `host` должно выполняться аналогичное соотношение: $0 < host < 0 \times 8000'0000'0000$.

3. Поле `offset` должно быть равно 0 для физической памяти.

4. Как уже было сказано, поле, описываемое массивом символов `idstr` должно содержать идентификатор «`rs.gam`». Еще одной особенностью данного поля является то, что кроме первых шести байтов, представляющих собой строку «`rs.gam`», оставшиеся 250 байт являются нулевыми байтами. Это связано с тем, что память для блока RAMBlock выделяется с помощью функции `mmap` с параметром `MAP_ANON`, который приводит к принудительной инициализации нулями выделяемой памяти.

5. Поле `next` должно содержать корректные ссылки на другие структуры RAMBlock, описывающие регионы памяти, отличные от физической.

Данные правила позволяют достаточно точно определить, какие из найденных смещений являются действительными смещениями структур RAMBlock.

Общий алгоритм восстановления физической памяти

Алгоритм восстановления физической памяти из гостевой машины под управлением QEMU в общем виде выглядит следующим образом:

1. Поиск процессов QEMU
2. Восстановление адресного пространства процесса QEMU из расширенного образа памяти
3. Поиск идентификатора «`rs.gam`»
4. Нахождение смещения предполагаемой структуры RAMBlock
5. Проверка валидности структуры
6. Для каждой валидной структуры RAMBlock — получение значений полей `host` (виртуальный адрес блока памяти в адресном пространстве QEMU) и `length` (размер данного блока).
7. Чтение блока длины `length` начиная с виртуального адреса `host` в адресном пространстве процесса QEMU.

После восстановления памяти описанным образом следует провести проверку корректности восстановленной памяти. Для этих целей был использован `Volatility Framework` — известное программное средство для ана-

лиза образов памяти, снятых с систем под управлением Windows и Linux. Восстановленные данные были опознаны Volatility как образ памяти, а также были успешно найдены все запущенные в гостевой машины процессы.

Заключение

В работе описан метод извлечения физической памяти гостевой машины, запущенной под управлением QEMU, из образа памяти базовой системы. После успешного извлечения появляется возможность применить стандартные методы анализа образов памяти для исследования гостевой системы.

Описанный метод можно обобщить на другие средства виртуализации с исходным кодом (VirtualBox, KVM, Xen), что позволит проводить эффективный анализ инцидентов, произошедших на виртуализированных и облачных платформах.

Литература

1. *А. Овчинников*, Расширенный анализ образа памяти на платформе Windows // Материалы всероссийской научной конференции по проблемам информатики «СПИСОК-2012», 2012, С. 38–43.
2. *М. Руссинович*, Внутреннее устройство Microsoft Windows (4-ое издание), 2008
3. *А. Merola*, Data Carving Concepts // SANS Institute, 2008
4. *А. Schuster*. Searching for Processes and Threads in Microsoft Windows Memory Dumps // Digital Forensic Research Workshop, Digital Investigation, 2006.
5. *J. Grier*, Detecting data theft using stochastic forensics // Digital Investigation, 2011.
6. *J. Kornblum*. Using Every Part of the Buffalo in Windows Memory Analysis // Digital Investigation Journal, 2007.
7. *J.S. Okolica, G.L. Peterson*. Windows operating systems agnostic memory analysis // Digital Forensic Research Workshop, 2010.
8. *M. Burdach*. An Introduction to Windows Memory Forensic, 2005, URL: <http://bandwidthco.com/whitepapers/compforensics/memory/Introduction%20To%20Windows%20Memory%20Forensic.pdf>
9. QEMU — open source processor emulator, URL: <http://qemu.org>

ОБЩИЙ ПОДХОД К ВОССТАНОВЛЕНИЮ АДРЕСНОГО ПРОСТРАНСТВА ПРОЦЕССА ИЗ ОБРАЗА ПАМЯТИ ОС ANDROID

П.Ю. Свидерский

*студент 5 курса кафедры системного программирования СПбГУ,
pavel@psviderski.name*

Аннотация: Массовое распространения ОС Android вызвало необходимость разработки надежных методов проведения цифрового криминалистического анализа (digital forensics) устройств, работающих на базе этой платформы. В работе рассмотрены основные цели и преимущества анализа образа оперативной памяти мобильных устройств, а также описан общий подход к анализу памяти ОС Android.

Введение

Цифровой криминалистический анализ (Digital Forensic Analysis) — это комплекс экспертных мероприятий, направленных на сбор, хранение и анализ информации, находящейся на компьютерах и цифровых устройствах. Полученная в результате экспертизы устройства цифровая информация, например, может быть предоставлена в суде как потенциальное доказательство, в случае если данное устройство имело отношение к рассматриваемому судебному делу.

В цифровом криминалистическом анализе выделяют отдельную область анализа мобильных цифровых устройств (мобильные телефоны, смартфоны, КПК, планшетные компьютеры, устройства GPS). С активным ростом доступности этих мощных устройств увеличилась частота использования их преступниками, в том числе для помощи в совершении преступлений.

Преступники могут использовать мобильные устройства для ряда действий, таких как совершение мошенничества через интернет и электронную почту, вымогательство с помощью SMS-сообщений, коммуникации, связанные с торговлей наркотиками или детской порнографией и др. Мобильные устройства содержат большой объём доказательной информации, которая связана с личностью владельца этого устройства, например, история звонков, контакты (телефонная книга) и SMS-сообщения. Современные же мобильные устройства (к примеру, смартфоны) могут содержать ещё больше полезной для анализа информации, такой как фотографии, видео, электронные календари и заметки, электронная почта (отправленные и принятые письма), история посещения страниц в Web-браузере, история сообщений и контакты в социальных сетях.

До недавнего времени процедура создания точной и достоверной копии данных с мобильного устройства в ходе цифрового криминалистиче-

ского анализа ограничивалась только устройством хранения информации, таким как встроенная флеш-память или внешняя карта памяти. Это означает, что процесс анализа полагался только на файловую систему накопителя. У такого подхода есть большой минус: файловая система, снятая с флеш-накопителя мобильного устройства, может быть зашифрована. Если неизвестен ключ шифрования, процедура расшифровки файловой системы становится практически невозможной.

Для проведения более полной экспертизы мобильного устройства, помимо флеш-накопителя необходимо анализировать оперативную память устройства. Анализ оперативной памяти имеет огромное значение для исследования, поскольку она содержит большое количество информации, которую трудно или невозможно получить, располагая только флеш-накопителем.

На данный момент уже существует большое количество исследовательских работ, в которых обсуждается анализ файловой системы ОС Android. Также успешно развивается область снятия и исследования оперативной памяти мобильных устройств на базе ОС Android [7].

В этой работе представлены универсальные, независимые от версий ОС Android и моделей устройств, методы для восстановления виртуального адресного пространства процессов из снятого образа памяти.

Образ оперативной памяти

В оперативной памяти мобильного устройства хранится актуальная на данный момент информация о работающей на устройстве операционной системе, запущенных приложениях (в том числе вредоносных), открытых файлах, установленных сетевых соединениях с другими устройствами или веб-узлами, расшифрованных данных и многое другое. Более того, может храниться и ранее используемая вышеописанная информация, например, список недавно завершённых приложений. Отсутствие такой информации может сделать некоторые исследования невозможными, например, обнаружение и анализ современных вредоносных программ, которые не взаимодействуют с энергонезависимой памятью.

Для анализа оперативной памяти с работающего мобильного устройства снимается её копия (образ) и записывается в файл. Далее вся работа ведется только с образом памяти. Для снятия такого образа может быть использован, например, специальный программный модуль LiME¹, который загружается в ядро операционной системы [7].

Одна из ключевых целей анализа образа памяти — находить и идентифицировать в нём процессы, включая завершённые или потенциально скрытые. Каждый процесс имеет свое виртуальное адресное пространство,

¹<http://code.google.com/p/lime-forensics/>

которое отображается в оперативную память, не обязательно прямолинейно [3]. В ОС Android всеми пользовательскими приложениями управляет виртуальная машина Dalvik. Восстановленное адресное пространство процесса из образа памяти позволяет проводить дальнейший анализ структур виртуальной машины Dalvik и работающего на ней приложения. В конечном итоге из пользовательских приложений могут быть извлечены такие полезные данные, как список контактов владельца мобильного устройства, отправленные и принятые SMS-сообщения, список недавних звонков и т.п.

Нахождение процессов

Сложность поиска процессов в образе обусловлена существованием большого количества различных версий и сборок ядра Linux, которое используется в семействе ОС Android. В разных версиях и сборках ядра структуры, которые описывают процесс могут существенно различаться: иметь разные размеры, разные смещения полей, некоторые поля могут отсутствовать. Поэтому для нахождения процессов в образе будут использоваться особенности реализации аппаратного обеспечения мобильных устройств, в частности, архитектуры ARM.

Для каждого пользовательского процесса в системе ядро создаёт структуру каталог страниц (Page Directory). Эта структура используется аппаратным обеспечением для управления страницами памяти процесса и трансляции виртуального адреса в физический [2]. Следовательно, для нахождения процессов в образе нужно научиться искать в нём структуры каталог страниц. Эти же структуры используются для восстановления адресного пространства процессов.

Каталог страниц как структура создаётся ядром, но, так как используется аппаратным обеспечением, её формат должен удовлетворять строгим спецификациям архитектуры ARM. Например, она должна иметь размер 16 Кбайт и адрес, выровненный по границе в 16 Кбайт, а также содержать 4096 записей размером 4 Кбайт. Было замечено, что ядро Linux для отображения адресов пространства пользователя использует только страницы размером 4 Кбайт, а для адресов пространства ядра — только секции размером 1 Мбайт, поэтому соответствующие записи в каталоге страниц (0–3071 для пространства пользователя и 3072–4095 для пространства ядра) должны иметь надлежащий формат. Найдя в памяти места, где выполняются эти условия, можно провести дополнительные проверки, чтобы быть уверенным, что найден именно каталог страниц. Например, проверить флаги прав доступа для записей пространства ядра, они должны иметь только привилегированный доступ [1].

Такой метод нахождения процессов не зависит от версии ОС Android, ядра Linux и модели мобильного устройства. Дополнительным преимуществом этого подхода является способность находить недавно завершённые

процессы, а также скрытые вредоносными программами (см. *rootkit*²), так как даже скрытый процесс в системе должен иметь каталог страниц и удовлетворять вышеописанным условиям программно-аппаратного взаимодействия.

Восстановление адресного пространства процесса

После того, как найден каталог страниц, можно приступить к восстановлению адресного пространства процесса. Для этого перебираются записи таблицы страниц первого уровня (найденный каталог страниц). В зависимости от того, какой формат имеет запись, производятся соответствующие действия. Если запись имеет формат секции, то из образа извлекается область данных размером 1 Мбайт, на которую указывает запись. Иначе запись содержит адрес таблицы страниц второго уровня. В таком случае перебираются записи соответствующей таблицы страниц второго уровня. Для каждой такой записи из образа извлекается область данных размером 4 Кбайт, на которую указывает эта запись. В стандартной конфигурации ОС Android не используется раздел подкачки для перемещения в него страниц, которые не помещаются в оперативной памяти, что значительно упрощает процесс восстановления адресного пространства.

Таким образом, для каждого процесса создаётся отдельный файл (*dump*), в который последовательно записываются извлечённые области. Этот файл содержит упорядоченную последовательность областей памяти, относящихся к виртуальному адресному пространству процесса. Для того, чтобы иметь возможность по виртуальному адресу определять соответствующее ему смещение в *dump*-файле, необходимо поддерживать дополнительную индексную структуру. Для каждой записанной в *dump*-файл области в ней хранится соотношение «диапазон виртуальных адресов — смещение».

Заключение

В работе описаны основные преимущества использования анализа оперативной памяти в ходе проведения экспертизы мобильного устройства. Рассмотрен универсальный метод нахождения процессов и восстановления их адресного пространства, который не зависит от версии ОС Android и модели мобильного устройства. Восстановленное данным методом адресное пространство может быть полезным для дальнейшего анализа и восстановления из него структур виртуальной машины Dalvik, а также такой информации, как изображения, текстовые документы, исполняемые файлы, зашифрованные или запакованные вредоносные программы.

²*rootkit* — программа, основной задачей которой является скрытие следов присутствия в системе себя или другой программы.

Литература

1. ARM Architecture Reference Manual. ARM Limited. 2005.
 2. *DP Bovet, M. Cesati*. Understanding the Linux kernel. 3rd ed. Sebastopol, CA: O'Reilly Media, Inc. 2006.
 3. *M. Gorman*. Understanding the Linux Virtual Memory Manager. 2007
 4. *M. Burdach*. Physical memory forensics. USA: Black Hat. 2006.
 5. *I. Kollar*. Forensic RAM dump image analyser. Charles University in Prague. 2010.
 6. *K. Saur, J.B. Grizzard*. Locating x86 paging structures in memory images. Digital Investigation Journal. 2010.
 7. *J. Sylve, A. Case, L. Marziale, G.G. Richard*. Acquisition and analysis of volatile memory from android devices. Digital Investigation. 2011.
-

КОРПОРАТИВНАЯ СИСТЕМА ОРГАНИЗАЦИИ ИНФОРМАЦИИ И СОБЫТИЙ

Карбушев Дмитрий Олегович

Студент 4 курса института ИСУиИТ ЮГУ, expkdo@gmail.com

Волошин Семен Викторович,

Студент 4 курса института ИСУиИТ ЮГУ, wanttomit@mail.ru

Аннотация: Для большинства государственных учреждений и предприятий существует проблема информирования сотрудников о новостях и предстоящих событиях. Для этого, администрация использует либо свои информационные ресурсы, сторонние ресурсы, связь либо непосредственный личный контакт.

Введение

Для большинства государственных учреждений и предприятий существует проблема информирования сотрудников о новостях и предстоящих событиях. Для этого, администрация использует либо свои информационные ресурсы, сторонние ресурсы, связь либо непосредственный личный контакт.

Существующие сервисы для решения таких проблем не учитывают специфику структуры организации, так как направлены, прежде всего, на использование сервиса отдельными группами людей, не имеющих привязки к месту работы, учебы и должностям.

Предлагаемое решение позволяет сотруднику создавать, просматривать и управлять событиями внутри корпоративной сети. В качестве примера работы сервиса была выбрана ИС Югорского государственного университета

Структура

Чтобы удовлетворять всем требованиям, сервис организации информации и событий имеет блочную структуру. Так как некоторые функциональные блоки могут быть опционально отключены, то необходимо спроектировать модуль так, чтобы система адаптировалась под него — определяла его наличие или отсутствие.

Система реализована с помощью технологий ASP.NET (C#) и MySQL (Также реализованы классы, позволяющие работать с Oracle и MSSql).

В составе имеет два основных элемента:

- Блок «Battleship» (анг. Линкор) — Ядро системы. CMS сайта. Отвечает за все манипуляции с данными.
- Блок «Sail» (анг. Паруса) — Интерфейсная оболочка.

И два дополнительных:

- Блок «Dock» (анг. Пристань) — Модуль отвечает за привязку сервиса к внешней системе, источнику данных о пользователях и их ролях, в нашем случае ИС ЮГУ.
- Блок «Mariner» (анг. Матрос) — Синтаксические анализаторы внешних информационных ресурсов для формирования списка событий, не принадлежащих корпоративной сети.

Структура сервиса организации информации и событий представлена на рисунке 1.

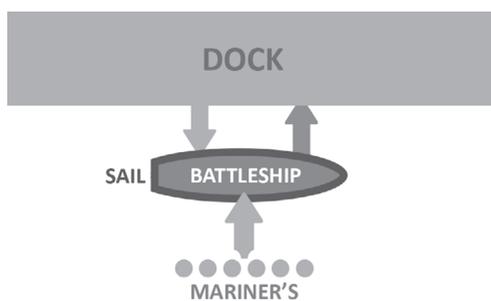


Рис. 1. Структура сервиса

Технологии

Для обмена информацией между блоками используются различные технологии передачи данных (рисунок 2).

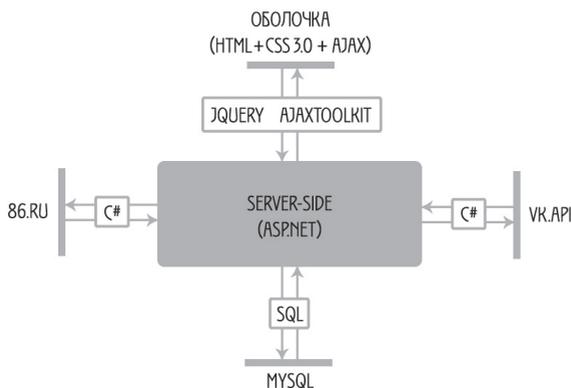


Рис. 2. Используемые технологии

Сложность проектирования модулей состоит в том, чтобы согласно используемым технологиям и структуре сервиса организации событий соблюдать порядок и очередность действий, так как внедрение отдельных частей может повлечь за собой большую нагрузку на сервер или мешать остальным блокам работать в штатном режиме.

Заключение

Особенностью предложенного решения является то, что разработанная система может использоваться, как интегрировано во внешний источник данных о пользователях, так и отдельно от него.

Полная независимость компонентов друг от друга, позволяет работать системе практически безошибочно.

Модульная структура приложения делает сервис расширяемым и облегчает его последующую модернизацию, а наличие готовых интерфейсов позволяет адаптировать систему практически к любым внешним и внутренним условиям.

ОБЗОР СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ¹

Агапова Татьяна

*математико-механический факультет, 2 курс,
tatjana.agarova@gmail.com*

Аннотация: Управление бизнес-процессами является важной стороной деятельности организации. В настоящее время существуют системы управления бизнес-процессами (BPM-системы), предоставляющие возможности моделирования, исполнения и мониторинга бизнес-процессов. В данной статье рассмотрены наиболее популярные BPM-системы, определены перспективы дальнейших исследований в области управления бизнес-процессами, представлено новое средство моделирования бизнес-процессов.

Введение

Одним из ключевых понятий бизнес-моделирования является бизнес-процесс — совокупность взаимосвязанных мероприятий или задач, направленных на создание определенного продукта или услуги для потребителей. Процесс включает в себя связанные между собой процедуры и функции, решающие совместно некоторую задачу (примером бизнес-процесса может являться описание последовательности действий клиента и продавца при покупке автомобиля). Бизнес-модель в таком случае представляет собой формализованное описание бизнес-процессов, отражающих деятельность организации, реальную или предполагаемую, и даёт ответы на следующие вопросы:

- какие процедуры и в какой последовательности нужно выполнить для получения результата;
- кем они должны быть выполнены;
- что требует на входе и что генерирует на выходе процедура процесса;
- какие ресурсы необходимы для выполнения процесса;
- каким образом документирован и параметризован процесс.

Спектр применения моделей бизнес-процессов довольно широк и охватывает самые различные аспекты управления деятельностью организации[1]. Модели являются мощным инструментом бизнес-анализа, поскольку позволяют в формализованной форме представить структуру и динамику организации и упрощают выявление текущих проблем, и, в связи с этим, моделирование имеет большую ценность при реинжиниринге бизнес-процессов. Также модель процесса может определять требования к программному обе-

¹ Работа поддержана грантом университета №6.39.1054.2012

спечению, автоматизирующему данный процесс, обеспечивая взаимопонимание между менеджерами, бизнес-аналитиками и разработчиками. Более того, если семантика модели строго определена, представляется возможным исполнять представленный ей процесс напрямую в специализированном движке. В дополнение к этой возможности, строго специфицированная семантика позволяет осуществлять автоматическую верификацию процесса до того, как начнётся его исполнение.

Подобная широта применения породила множество методологий и языков, реализующих различные подходы к моделированию бизнес-процессов. Наиболее распространёнными из них являются методологии IDEF, ARIS и BPM. Первые две из них ориентированы прежде всего на человека, то есть на создание моделей процессов как таковое, без связи с последующим их исполнением. Как следствие, в процессе автоматизации бизнес-процессов модели, созданные с помощью соответствующих нотаций, играют скорее пассивную роль (описывают требования к автоматизирующей процесс системе), нежели активную (сами представляют собой процесс, пригодный для исполнения машиной). Более подробное сравнение данных методологий приведено в [4].

В BPM-системах, напротив, с моделирования комплексное мероприятие по управлению бизнес-процессами только начинается. Как правило, наряду со средствами моделирования, BPM-система также предоставляет средства симуляции, мониторинга, анализа и среду исполнения бизнес-процессов, при этом предоставляя пользователю набор интерфейсов для взаимодействия с исполняющимся процессом. В отличие от реинжиниринга, который представляет собой однократное радикальное преобразование бизнес-процессов и, как следствие, связан с трудоёмкой перестройкой используемого ПО, использование BPM предполагает непрерывные усовершенствования бизнес-процессов с использованием специальных систем, позволяющих править процессы «на лету», значительно снижая стоимость их оптимизации[2].

Для моделирования и описания исполняемых бизнес-процессов BPM-системы используют следующие языки и нотации[5]:

- BPMN (Business Process Model and Notation) — визуальная нотация моделирования бизнес-процессов. Основная цель BPMN — создание моделей, которые будут одинаково доступны для всех участников бизнеса. Для исполнения модели в нотации BPMN она транслируется в описание процесса на BPEL, которое затем загружается в движок BPM-системы;
- BPEL (Business Process Execution Language) — XML-язык исполнения бизнес-процессов, описывает бизнес-процесс в виде последовательности взаимосвязанных веб-сервисов;
- XPDL (XML Process Definition Language) — формат обмена данными между BPM-системами, может рассматриваться как XML-сериализация BPMN.

Обзор BPM-решений

Среди возможных BPM-решений можно выделить отдельный класс BPM-систем. BPM-системы предназначены для управления бизнес-процессом на всех стадиях его жизненного цикла, в связи с чем включают в себя в обязательном порядке следующие компоненты[3]:

- дизайнер для моделирования бизнес-процессов;
- движок исполнения;
- средства контроля и мониторинга выполнения бизнес-процессов.

Исходя из предназначения BPM-систем и общих требований к прикладному ПО, можно сформулировать критерии оценки BPM-решений, перечисленные ниже.

1. Простота и удобство использования.
2. Поддержка общепринятых стандартов (XPDL, BPEL).
3. Полнота поддержки нотации BPMN.
4. Степень участия программистов в автоматизации процесса.
5. Наличие средств верификации бизнес-процессов.
6. Лицензия и стоимость.
7. Интеграция с прочими корпоративными приложениями.
8. Возможность динамического изменения бизнес-процесса.

ELMA BPM Suite

ELMA BPM Suite — разработка российской компании Elewise, предназначенная для управления бизнес-процессами.

Моделирование бизнес-процессов в ELMA осуществляется в нотации BPMN. После создания диаграммы, задания параметров процесса и определения данных, с которыми работает бизнес-процесс, он публикуется на сервере ELMA и становится доступным для запуска посредством веб-интерфейса системы. Для каждого запущенного экземпляра бизнес-процесса система создаёт его карточку, что позволяет отследить и проконтролировать исполнение процесса. Через карточку процесса пользователи, обладающие соответствующими правами, могут управлять его ходом.

Одной из целей BPM является минимизация необходимости участия программистов в автоматизации исполнения бизнес-процессов. В ELMA весь исполняемый код и интерфейсы к шагам бизнес-процессов генерируются автоматически. Программирование вручную требуется при создании сценариев к шагам процесса типа «скрипт». Сценарии пишутся на языке C# и могут быть оформлены в виде плагинов к системе и переиспользованы.

ELMA поддерживает принцип непрерывного усовершенствования процессов, предоставляя возможность изменять их «на лету», без приостановки исполнения.

Существуют средства интеграции с основными корпоративными приложениями (SOA, CRM, почтовые сервисы). Российский пользователь может отнести к плюсам ELMA также тесную интеграцию с «1С: Предприятие».

Недостатками системы являются весьма скромная и далеко не полная реализация BPMN, отсутствие поддержки XPDЛ и BPEL, в связи с чем не представляется возможным использовать в ELMA процессы, разработанные в какой-либо другой BPMS. Отсутствуют какие-либо средства верификации бизнес-процессов. Кроме того, ELMA обладает довольно сложным интерфейсом.

ELMA распространяется в трёх версиях, предназначенных для организаций с различными потребностями и численностью сотрудников. Помимо этого, существует бесплатная версия на пять рабочих мест.

Bizagi BPM Suite

Bizagi BPM Suite состоит из следующих компонентов:

- Bizagi Process Modeler — дизайнер бизнес-процессов. Распространяется бесплатно;
- Bizagi Studio — модуль, предназначенный для автоматизации бизнес-процесса;
- Bizagi BPM Server — сервер, исполняющий процессы.

Моделирование в Bizagi осуществляется в нотации BPMN. По сравнению с прочими BPM-решениями Bizagi поддерживает наиболее полный уровень соответствия спецификации BPMN. Также существует возможность коллективного проектирования.

После создания модели процесса она загружается в Bizagi Studio. Здесь пользователь может определить всю информацию, необходимую для автоматизации процесса (определение данных, исполнителей процесса, пользовательских интерфейсов, задание бизнес-правил и т. д.). Посредством данного модуля можно осуществить интеграцию с прочими корпоративными приложениями. Автоматизация осуществляется исключительно с помощью графических интерфейсов, программирование необходимо только для спецификации скриптовых задач.

Автоматизированный посредством Bizagi Studio процесс загружается на сервер и становится доступным для исполнения. С помощью веб-интерфейса осуществляется выполнение пользовательских задач и контроль исполнения процесса. Система формирует довольно подробные отчёты по процессу в целом и по каждой задаче в рамках процесса. Отчёты также доступны через веб-интерфейс.

Изменение процесса в Studio ведёт к его немедленному изменению на сервере, все запущенные экземпляры процесса сразу же начинают работать по новой модели, не требуя приостановки.

Bizagi BPM Suite предоставляет возможности обмена моделями между приложениями, поддерживая импорт и экспорт в форматы XPDL и MS Visio. Однако, наряду с этим, генерация моделей исполняемых процессов на языке BPEL отсутствует.

В Bizagi нет никаких средств верификации процессов, что является существенным недостатком, принимая во внимание, что данная система позволяет моделировать довольно сложные процессы, требующие проверки на корректность. Также отсутствуют какие-либо возможности симуляции бизнес-процессов — увидеть, как выполняется процесс, можно только исполнив его.

Отдельно следует отметить очень удобный, простой и понятный интерфейс всех компонентов системы.

Bonita Open Solution

Отличительной особенностью Bonita Open Solution является наличие opensource-версии. Строго говоря, данная версия не является полноценной BPM-системой, так как в ней отсутствуют средства мониторинга процессов, доступные в коммерческом варианте BOS. Opensource-версия предоставляет лишь самую базовую функциональность, необходимую для управления бизнес-процессами, позволяя разрабатывать процессы и исполнять их.

BOS состоит из трёх компонентов:

- Studio — среда для моделирования и автоматизации процессов;
- Execution Engine — движок исполнения процессов;
- User Experience — интерфейс для взаимодействия пользователя с ассоциированными с ним процессами.

Для моделирования процессов BOS использует нотацию BPMN. Можно отметить довольно полную поддержку нотации. BOS поддерживает импорт диаграмм из формата XPDL, однако возможности экспорта в него отсутствуют. Кроме того, нет поддержки BPEL.

В Bonita Studio автоматизация процесса неотделима от моделирования. Определение данных, участников, создание интерфейсов к шагам процесса происходит в том же окне, что и создание визуальной модели. Отсутствие необходимости часто переключаться между окнами редакторов ведёт к ускорению разработки процесса. В целом можно сказать, что Studio удобна и эргономична.

Отличительной особенностью BOS является возможность её взаимодействия с большим количеством приложений и сервисов посредством концепции коннекторов. В число поддерживаемых приложений входят различные базы данных, почтовая служба, веб-сервисы и многие другие. С помощью коннекторов можно исполнять скрипты и Java-классы. Существуют средства разработки своих собственных коннекторов. Автоматизация скриптовых задач осуществляется также посредством коннекторов, а не путём прямого программирования.

В BOS отсутствуют средства статической верификации бизнес-процессов, однако есть возможность их симуляции с различными параметрами и формирования отчётов по результатам, что может оказать значительную помощь в тестировании и отладке процесса.

Недостатком BOS является отсутствие поддержки динамического изменения бизнес-процесса, затрудняющее оптимизацию процессов. Данное упущение является существенным для BPM-системы, поскольку возможность изменения процесса во время его исполнения входит в число ключевых концепций BPM.

Выводы

Проведём сравнение рассмотренных систем в соответствии со сформулированными выше критериями.

Показатель	ELMA	Bizagi	Bonita
Простота и удобство использования	Довольно сложный интерфейс	Удобна и проста	Проста и эргономична
Поддержка общепринятых стандартов (XPDL, BPEL)	Нет	XPDL поддерживается, BPEL — нет	Только импорт из XPDL
Полнота поддержки нотации BPMN	Только базовые элементы	Одна из наиболее полных реализаций	Довольно полная поддержка
Степень участия программистов в автоматизации процесса	Реализация скриптовых задач	Реализация скриптовых задач	Реализация коннекторов
Наличие средств верификации бизнес-процессов	Нет	Нет	Только симуляция, статической верификации нет
Лицензия и стоимость	Платная, существует бесплатная версия на 5 мест	Платная	Платная, существует opensource-версия
Интеграция с прочими корпоративными приложениями	Основные приложения, есть возможность интеграции с «1С: Предприятие»	Доступна интеграция с большинством корпоративных приложений	Широкие возможности интеграции
Возможность динамического изменения бизнес-процесса	Есть	Есть	Нет

Таблица 1. Сравнение BPM-систем

При рассмотрении наиболее типичных BPM-систем выявились следующие их недостатки: отсутствие средств статической верификации бизнес-процессов и ограниченные возможности обмена моделями между системами. Opensource-вариант есть только у Bonita, однако его функциональность сильно сокращена по сравнению с полной версией. В связи с этим, существует потребность в дальнейших исследованиях в области статической верификации процессов и, в идеале, реализации их результатов в opensource BPM-системе.

Реализация на основе QReal

Вследствие обозначенных выше перспектив исследований в области BPM-систем было принято решение реализовать редактор бизнес-процессов на базе DSM-платформы QReal, разрабатывающейся на кафедре системного программирования СПбГУ[6]. Помимо того, что разработка данного редактора послужила бы базой для дальнейших экспериментов в области моделирования бизнес-процессов, это также стало бы ещё одной апробацией QReal в реальном проекте. В данный момент редактор поддерживает некоторые основные элементы BPMN: события, условные операторы и задания. Пример процесса представлен на рис. 1.

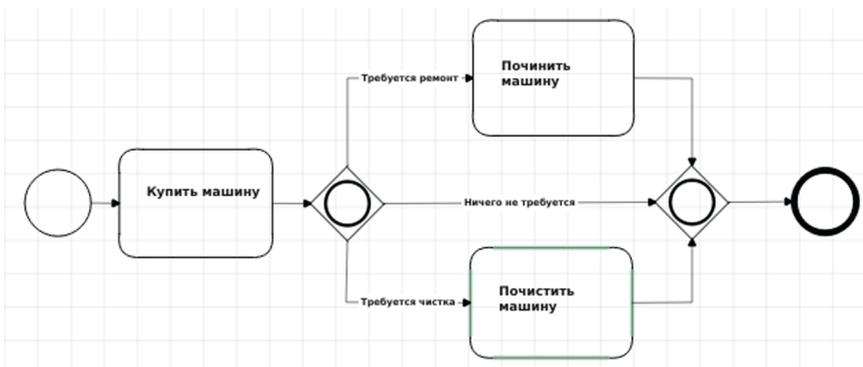


Рис. 1. Пример процесса, созданного средствами QReal

В процессе разработки редактора потребовалось расширить движок QReal, добавив в него возможность отображения элемента на сцене по-разному в зависимости от свойств элемента. К примеру, в случае редактора BPMN это используется для определения разных пиктограмм для разных типов событий и условных операторов.

В будущем планируется добавить поддержку прочих элементов BPMN, в частности, пулов, подпроцессов и граничных событий (boundary events). Также следует учесть, что BPMN строго определяет корректность диа-

грамм. К примеру, существуют ограничения на последовательность элементов в потоке управления при использовании составных событий и условных операторов. Также, согласно спецификации BPMN, потоки управления не могут пересекать границы пулов, а потоки сообщений — связывать элементы внутри одного пула. Возможность контроля подобных правил требует развитых средств задания ограничений на визуальные языки.

Заключение

В данной статье рассмотрены три распространённые BPM-системы, выявлены их основные недостатки, определены перспективы дальнейших исследований в области моделирования бизнес-процессов. В частности, перспективной областью является статическая верификация бизнес-процессов. Рассмотренные системы данной возможностью практически не обладают. В связи с этим, было принято решение разработать собственный редактор для моделирования бизнес-процессов, что может стать хорошей базой для дальнейших экспериментов в данной области.

Литература

1. *А.Г. Зуева, Б.В. Носков, Е.В. Сидоренко, Е.И. Всяких, С.П. Киселев.* Практика и проблематика моделирования бизнес-процессов // ДМК Пресс, М., 2008, 246 С.
2. *Ismael Chang Ghalimi.* Будущее BPMN, URL: <http://nvoynov.blogspot.ru/2007/08/bpm-future-of-bpm-by-by-ismael-chang.html>
3. *BPMS.RU,* Введение, URL: <http://bpms.ru/intro/index.html>
4. *Федоров И.* Сравнительный анализ нотаций моделирования бизнес-процессов, URL: <http://www.osp.ru/os/2011/08/13011140/>
5. *Артамонов И.В.* Современные стандарты описания и исполнения бизнес-процессов, URL: <http://ecm-journal.ru/post/Sovremennye-standarty-opisanija-i-ispolnenija-biznes-processov.aspx>
6. *Кузенкова А.С., Дерипаска А.О., Таран К.С., Подкопаев А.В., Литвинов Ю. В., Брыксин Т.А.,* Средства быстрой разработки предметно-ориентированных решений в metaCASE-средстве QReal // Научно-технические ведомости СПбГПУ, Информатика, телекоммуникации, управление. Вып. 4 (128). СПб.: Изд-во Политехнического Университета. 2011, С. 142–145.

РАЗРАБОТКА МЕТАМОДЕЛИРОВАНИЯ «НА ЛЕТУ» В СИСТЕМЕ QREAL

Птахина А.И.

*студентка 4 курса кафедры системного программирования СПбГУ,
alinaptakhina@gmail.com*

Аннотация: В статье описывается подход к разработке ПО, основанный на использовании визуального программирования и позволяющий быстро и легко модифицировать визуальный язык программирования. Под метамоделированием «на лету» подразумевается возможность добавлять новые элементы языка, удалять и изменять существующие прямо в процессе разработки, а также возможность создать свой собственный язык «с нуля». Приводится описание реализации такого подхода в системе QReal — проекте научно-исследовательской группы кафедры системного программирования СПбГУ, в виде прототипа, включающего в себя все перечисленные возможности модификации визуального языка.

Введение

Визуальное программирование [1] позволяет ускорить процесс разработки приложений и сделать его простым и удобным. При таком подходе программа представляется в виде набора диаграмм, формально не связанных друг с другом. Визуальные модели описывают отдельные аспекты ПО, что позволяет не рассматривать все многообразие предметной области, а сосредоточиться лишь на некоторых ее свойствах, и моделировать ПО с разных точек зрения.

На практике часто возникают задачи, при автоматизации решений которых естественным образом возникает необходимость описывать сложные, нетривиальные структуры. Для этого довольно часто возникает потребность видоизменения существующих или добавления новых сущностей языка. Однако визуальные редакторы, в основном, предоставляют пользователю фиксированный набор диаграмм и сущностей. Это ограничивает пользователя, заставляя думать в терминах конкретной диаграммы, а не в терминах предметной области. И, как следствие, приводит к тому, что пользователь отдает предпочтение простому листу бумаги и ручке и сначала пытается нарисовать эскиз на бумаге с помощью абстрактных кругов и стрелок, а потом думает, как его полученное решение перенести в визуальный редактор. Как правило, этот процесс длится долго: нужных элементов языка нет, а добавление новых и изменение существующих затруднительно. Существуют специальные инструментальные средства, позволяющие быстро создавать визуальные языки и редакторы для них, они называются DSM-платформами.

Безусловно, свести работу с CASE-средством до уровня бумаги и ручки не получится, однако хочется предоставить пользователю большую свободу действий. Хочется, чтобы пользователь мог сам создать нужный для его предметной области язык без особых усилий. Ведь решение любой задачи проще всегда начинать с эскиза, с простых элементов, которые в дальнейшем можно детализировать и уточнять в зависимости от потребностей предметной области. Предлагаемый нами подход, позволяющий быстро и легко расширять систему, мы будем называть метамоделированием «на лету». Данная работа базируется на идеях и результатах, полученных в рамках дипломной работы Е.И.Такун [3].

Интерпретация метамodelей

Визуальные языки в современных DSM-платформах, как правило, задаются с помощью метамodelей, то есть моделей синтаксиса языка. Они описывают, какие элементы могут находиться на диаграмме, какие свойства есть у этих элементов, и как элементы могут быть связаны друг с другом. Существует два принципиально разных подхода по работе с метамodelью. Один из них заключается в том, что описание языка может быть сгенерировано как код на текстовом языке программирования по метамodelи. Второй — метамodelь может интерпретироваться во время работы DSM-платформы. Реализация метамodelирования «на лету» возможна только в случае интерпретируемой метамodelи. Основное отличие от генеративного подхода заключается в том, что интерпретация моделей происходит непосредственно во время работы приложения, в то время как генерация обязана завершиться до начала работы. Интерпретатор метамodelей в системе QReal был реализован в рамках моей курсовой работы 3-го курса [2].

Класс интерпретатора для получения информации о сущностях языка имеет доступ к репозиторию с сохраненной метамodelью. Он позволяет извлекать из репозитория все необходимые данные. Для реализации метамodelирования «на лету», класс интерпретатора был расширен функциями добавления и изменения информации.

Режим интерпретируемой диаграммы

В QReal интерфейс пользователя состоит из:

- сцены — на ней рисуются диаграммы;
- палитры — на ней располагаются элементы визуальных языков;
- обозревателя логической модели — в его древовидном представлении отображается логическая структура системы;
- обозревателя графической модели — он древовидно показывает структуру диаграмм, которыми логическая структура визуализируется;
- редактора свойств элементов.

Метамодели в системе QReal можно создавать и изменять при помощи встроенного метаредактора. Он представляет собой удобный инструмент, позволяющий задавать сущности языка и устанавливать между ними отношения. Созданную в метаредакторе метамодель можно сохранить в формате сохранения QReal.

Чтобы перейти в режим интерпретируемой диаграммы, достаточно при запуске системы QReal выбрать в стартовом диалоге пункт меню «Открыть интерпретируемую диаграмму» (см. рис. 4) и загрузить файл с сохраненной метамоделью.

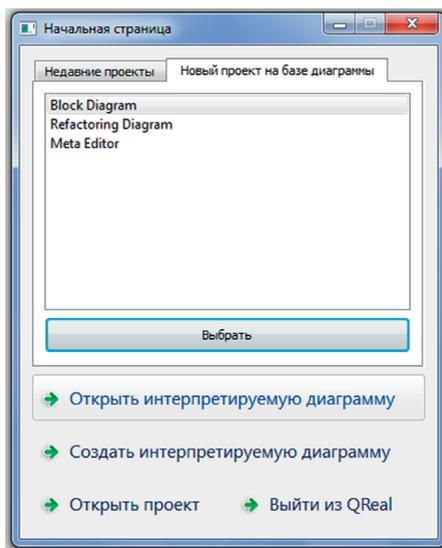


Рис. 4. Стартовый диалог QReal

В случае, если не имеется сохраненной метамодели, или хочется создать свой собственный визуальный язык, можно воспользоваться пунктом меню «Создать интерпретируемую диаграмму». В открывшемся окне ввода необходимо ввести имя создаваемой диаграммы. После этого будет создана диаграмма и редактор с одноименным названием. После этого диаграмма не будет содержать больше никакой информации. Палитра элементов и обозреватели логической и графической модели, а соответственно и редактор свойств QReal, будут пусты, поскольку они пока не содержат элементов. Далее можно переходить к созданию новых элементов диаграммы. Полученную с помощью метамоделирования «на лету» метамодель можно сохранить и далее уточнять и детализировать в метаредакторе. В нем можно задать более сложные отношения между элементами: наследование, связи и

пр. После внесения изменений в метамодель ее можно снова использовать в режиме метамоделирования «на лету». Напишем подробнее об операциях, позволяющих изменять метамодель «на лету».

Добавление элемента производится путём указания вида элемента (сущность или связь), заполнения всех обязательных свойств (таких как имя, тип линии связи и т.д.) и задания графического представления (для этого используется встроенный в QReal редактор формы фигур, который используется и в метаредакторе). При этом в репозитории с метамоделью находится нужный редактор и диаграмма в нём, и ей в качестве сына добавляется новый узел, представляющий созданный элемент, со всеми заданными свойствами.

Удаление элемента несколько сложнее технически, поскольку уже созданная диаграмма может использовать удаляемый элемент. В этом случае все экземпляры элемента удаляются с диаграммы, при этом, если элемент содержал в себе какие-то другие элементы, то и они тоже будут удалены. Кроме того, удаляемый элемент может находиться в некоторых отношениях с другими элементами с точки зрения метамодели, например, от него могут наследоваться другие элементы. В этом случае наследующиеся от него элементы теряют все унаследованные свойства удалённого элемента. При этом сам элемент удаляется из репозитория, перед этим интерпретатор анализирует все связи этого элемента в метамодели и вносит необходимые изменения.

Редактирование свойств элемента доступно в окне, изображенном на рисунке 5. С его помощью элементу можно добавлять новые свойства, удалять и изменять существующие. Отметим, что модификация унаследованного свойства запрещена, а при редактировании свойств элемента-родителя соответствующие изменения автоматически происходят и в свойствах элемента-наследника.

Как ни странно, в связи с особенностью архитектуры системы QReal, свойство у элемента проще удалить, чем добавить новое. Это связано с тем, что существующие элементы нового свойства не имеют. При последующих обращениях к элементу система попытается получить значение этого свойства, возникнет ошибка. В связи с этим было принято решение запретить пользователю добавлять свойства элементу в случае, если хотя бы одно его представление или представление его элементов-наследников присутствует на сцене, либо в обозревателях графической или логической модели. Однако добавлять новые свойства и изменять существующие в такой ситуации, конечно, можно.

При добавлении нового свойства или редактировании существующего пользователю необходимо указать его имя, тип и значение по умолчанию (см рис. 6). Особое внимание следует уделить операции изменения типа, поскольку возможны ситуации, когда приведение типа произвести невозможно, например, не все значения string можно перевести в int. В таком случае

пользователю будет предоставлено предупреждение, что в случае некорректного преобразования имеющееся значение обнулится.

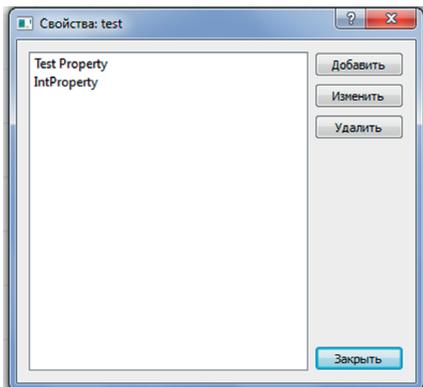


Рис. 5. Список свойств элемента

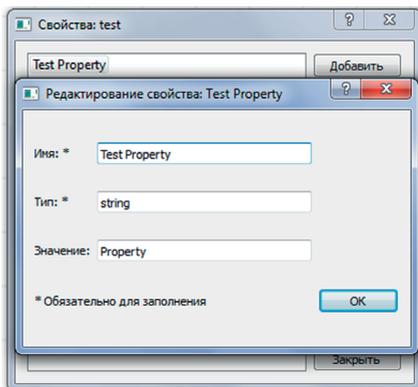


Рис. 6. Редактирование выбранного свойства

У пользователя есть возможность изменить графическое изображение элемента: загрузить существующий рисунок, сохраненный в графическом формате SVG, создать свой собственный с помощью имеющегося в QReal инструментария. Новое изображение будет применено как ко всем представителям данного элемента на сцене, так и к его иконкам в палитре и в обозревателе графической и логической модели.

Применение метамоделирования «на лету»

Рассмотрим, как языки визуального моделирования применяются в разработке аппаратного обеспечения. Создадим визуальный язык для HaSCoI [4]. Проведем апробацию метамоделирования «на лету» на задаче об арбитра динамических приоритетов 4 в 1. Сама задача формулируется следующим образом: «На один из четырех входов поступают данные, первый параметр пришедших данных — приоритет. Если в одном такте данные поступили на несколько входов, на выход выдаются данные с наибольшим приоритетом, остальные входы объявляются неготовыми. Если приходит только одно сообщение, оно отправляется на выход».

Решать поставленную задачу будем в соответствии с оригинальным решением: сначала напишем арбитр динамических приоритетов 2 в 1 (с двумя входами и одним выходом), затем создадим арбитр-функтор 4 в 1, использующий 3 арбитра 2 в 1, который и решит задачу. Для нас принципиально то, что мы взяли известную задачу, с существующим уже решением и не

придумываем нового, а просто реализуем известное при помощи метамоделирования «на лету». Анализируем сущности, необходимые для получения решения и строим их прямо в процессе разработки.

В HaScol для моделирования аппаратных систем используется понятие процесс — сущность, инкапсулирующая в себе ресурсы (данные), обработчики сигналов и другие процессы, и имеющая входы и выходы (порты). Процесс имеет тип, характеризующий его входы и выходы, процессы могут наследоваться друг от друга. Перенесем данное понятие в визуальный язык, добавив сущности «процесс» и «порт». И начнем рисовать диаграмму типов процессов. В ходе рисования мы понимаем, что нам нужно выразить наследование процессов, кроме того оказывается удобным понятие языка HaScol «функтор», т. е. процесс параметризованный другим процессом. Добавим эти понятия в язык. Для того, чтобы получить полноценную работающую программу, надо добавить реализацию процессам. Поскольку мы пока не хотим создавать отдельный язык для задания логики реализации, будем писать реализацию в текстовом виде, для чего добавим соответствующий элемент. Результаты этой работы приведены на рисунке 7.

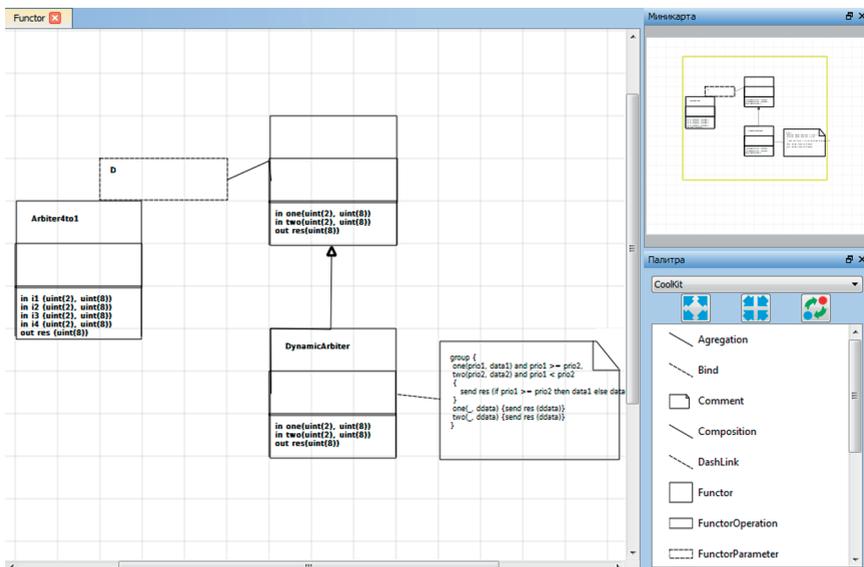


Рис. 7. Диаграмма типов процессов для задачи «Арбитр 4 в 1»

Аналогичным образом нарисуем диаграмму отображения портов для задачи «Арбитр 4 в 1», указанную на рисунке 8.

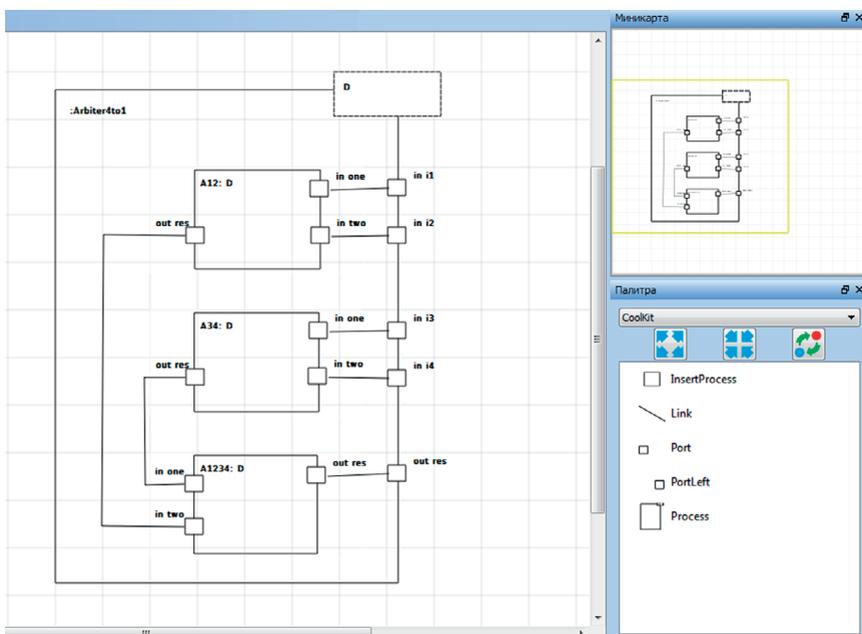


Рис. 8. Диаграмма отображения портов для задачи «Арбитр 4 в 1»

Здесь мы видим процесс верхнего уровня `Arbiter4to1` — без имени, но с типом и с формальным параметром функтора. `Arbiter4to1` имеет 4 входных порта и 1 выходной, их типы здесь уже можно не указывать, они были на диаграмме типов процессов, на предыдущем слайде. Процесс имеет три вложенных процесса `A12`, `A34` и `A1234` типа `D`, который, как следует из диаграммы типов процессов, описывает процессы, имеющие два входа и один выход. Если мы считаем, что в качестве параметра функтора передается арбитр 2 в 1, изображенное на рисунке соединение портов решает задачу.

Заключение

В результате данной работы было реализовано метамоделирование «на лету», предоставляющее пользователю возможность быстро и легко вносить изменения в визуальный язык программирования: добавлять новые элементы языка, удалять ненужные и изменять существующие. При этом все новые и измененные сущности сразу доступны для построения диаграммы, и в случае возможных конфликтов и некорректности системы пользователю предоставляется соответствующая информация о возможных последствиях.

В рамках данной концепции было произведено объединение редактирования модели и метамодели, что избавило пользователя от необходимости мыслить в терминах двух абстракций. Теперь уровень метамодели скрыт от пользователя, что позволяет ему полностью сконцентрироваться на задаче, а не на создании инструментария. Полученное решение было апробировано на реальной задаче.

Литература

1. *Д. Кознов*, Основы визуального моделирования. // Бином. Лаборатория Знаний, Интернет-Университет Информационных Технологий — 008.
 2. *А.И. Птахина*. Интерпретация метамodelей в metaCASE-системе QReal, курсовая работа, СПбГУ, кафедра системного программирования, 2012. http://se.math.spbu.ru/SE/YearlyProjects/2012/YearlyProjects/2012/345/345_Ptakhina_report.pdf [дата просмотра: 08.04.2013]
 3. *Е.И. Такун*. Реализация режима быстрого прототипирования в CASE-системе QReal, дипломная работа, СПбГУ, кафедра системного программирования, 2011 <http://se.math.spbu.ru/SE/diploma/2011/Takun%20Evgenija%20-%20text.doc> [дата просмотра: 08.04.2013]
 4. *Dmitri Boulytchev, Oleg Medvedev*. Hardware Description Language Based on Message Passing and Implicit Pipelining // EWDTS '10 Proceedings of the 2010 East-West Design & Test Symposium. — 2010. — Pages 438–441
-

РАЗРАБОТКА ВИЗУАЛЬНОГО КОНСТРУКТОРА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Белокуров Д.Н.

*4 курс, кафедра системного программирования СПбГУ,
eliriand@math.spbu.ru,*

Бумаков Н.В.

*4 курс, кафедра системного программирования СПбГУ,
nikita.bumakov@gmail.com,*

Захаров В.А.

*2 курс, математико-механический факультет СПбГУ,
vladimir.zakharov93@gmail.com,*

Чижова Н.А.

*5 курс, кафедра системного программирования СПбГУ,
nadya.chizhova@gmail.com.*

Аннотация: В статье рассматривается задача реализации визуального онлайн конструктора клиент-серверных мобильных приложений для платформ Android и Windows Phone. Представлен способ задания клиентской и серверной логики приложения при помощи визуального дизайнера. Приводится описание реализации прототипа, включающего в себя все стадии работы конструктора: задание логики клиентской части приложения, взаимодействие с сервером и с сервером приложений и генерация конечного файла.

Введение

В связи с распространенностью мобильных телефонов все более очевидной становится необходимость наличия средств для быстрой разработки мобильных приложений. На данный момент большинство подобных конструкторов, предложенных на рынке, поддерживают только создание приложений, не требующих клиентской или серверной логики. Примером такого приложения является приложение-визитка — приложение, отображающее общую информацию об организации, контакты и местоположение предприятия на карте. В статье описана реализация конструктора мобильных приложений, с помощью которого возможно задавать более сложную, в том числе серверную, логику работы приложения.

1. Обзор существующих решений

На данный момент на рынке существует ряд онлайн-сервисов, позволяющих создавать мобильные приложения с помощью визуального редактора:

AppInventor, iBuildApp, Apps.ru и другие. Такие средства предлагают пользователю инструмент для задания внешнего вида приложения и простейшей логики в виде указания переходов между экранами приложения.

Недостатком таких сервисов является то, что они не поддерживают возможность задания нетривиальной логики приложения, например: агрегацию данных, работу с внешними источниками данных, авторизацию и т. д. Таким образом, функционал большинства существующих сервисов ограничивается созданием информационных приложений, которые не способны на взаимодействие с сервером или внешними источниками данных.

Кроме того, многие из представленных на рынке сервисов позволяют создавать приложения только для одной определённой мобильной платформы, например, проект iBuildApp позволяет создавать приложения исключительно для платформы iOS.

2. Реализация

По результатам исследования и анализа существующих конструкторов для создания мобильных приложений был предложен собственный подход.

Общая архитектура разработанной системы состоит из нескольких компонент. Клиент обеспечивает моделирование интерфейса приложения, задание необходимой логики, а также возможность проверить корректность работы создаваемого мобильного приложения с помощью эмулятора мобильного телефона. Серверная часть выполняет агрегацию необходимых данных и генерацию итогового файла, который затем передается на клиент и сохраняется на компьютере пользователя. За саму генерацию приложений отвечает набор специально написанных генераторов приложений под различные платформы.

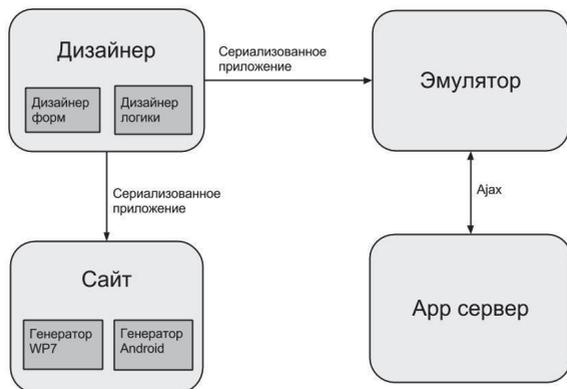


Рис. 1. Основные модули конструктора

Создание приложения происходит путем перетаскивания на рабочую область элементов палитры, таких как кнопка, изображение, текстовое поле и другие. Аналогичным образом происходит задание логики приложения. Созданное приложение можно протестировать с помощью эмулятора и затем сгенерировать соответствующий файл. Сформированный файл автоматически сохраняется на компьютере пользователя.

Для возможности свободного доступа пользователей к разработанному конструктору, прототип дизайнера и эмулятора выложен в облако Microsoft: Windows Azure.

2.1. Клиент

Клиентская часть сервиса состоит из двух компонент: дизайнера приложений и эмулятора. Дизайнер отвечает за создание интерфейса приложения и описание его бизнес-логики. Эмулятор позволяет запустить создаваемое приложение прямо в окне интернет-браузера.

Дизайнер приложений включает в себя такие компоненты, как палитра элементов, редактор свойств, основная рабочая область, менеджер экранов и редактор клиентской логики приложения. Для задания клиентской логики на данный момент используется событийно-триггерная система, позволяющая задавать поведение приложения при реализации событий определённых типов, например «пришёл ответ на запрос об авторизации», «сработал таймер», «нажата кнопка» и т. д. На данный момент клиентская логика описывается путём компоновки элементарных блоков-действий. Действия могут быть как общими для всех типов приложений, например условное ветвление или переход на другой экран, так и специфичными для конкретных типов создаваемых приложений, например отправка запроса об авторизации на сервер. В дальнейшем планируется использование графических языков для задания и клиентской, и серверной логики.

Для запуска онлайн-эмулятора или генерации мобильного приложения, подлежащего установке на смартфон, разрабатываемое приложение экспортируется в XML специального вида, который передаётся эмулятору или на сервер в зависимости от того, запускается эмулятор или же генерируется готовое приложение.

Эмулятор служит для запуска разрабатываемого приложения прямо в окне браузера. Это позволяет существенно облегчить отладку приложения благодаря тому, что не требуется каждый раз генерировать приложение и устанавливать его на смартфон.

Так как реализация сложного приложения на языке JavaScript — довольно нетривиальный процесс, то в качестве основного средства для разработки был выбран язык TypeScript¹. TypeScript представляет собой надстройку

¹ Домашняя страница языка Typescript, URL: <http://www.typescriptlang.org/>

над JavaScript и транслируется в него. Выбранное средство выгодно отличается наличием типизации, удобной в использовании объектной моделью. Кроме того, для TypeScript имеется возможность отладки приложений, что существенно облегчает процесс разработки.

В качестве средства, которое используется для построения интерфейса разрабатываемого приложения как в дизайнера, так и в эмуляторе используется библиотека JQuery Mobile². Данная библиотека содержит набор готовых мобильных виджетов, что позволило избежать необходимости реализовывать многие элементы интерфейса вручную.

2.2. Сервер разработки и сервер приложений

Сервер разработки обеспечивает связь между клиентской частью и генератором. Также на этом сервере предполагается работа репозитория для хранения диаграмм пользователя, благодаря которому каждый клиент в личном кабинете сможет редактировать ранее созданные диаграммы.

В качестве основной платформы для реализации сервера был выбрана Microsoft.NET: язык программирования F# и технология web-разработки WebSharper для взаимодействия с клиентской частью. Функциональный язык F# обеспечивает высокую производительность, и снижает объем написанного кода, в сравнении, например, с C#.

Для возможности задания серверной логики приложения реализован сервер приложений, который имеет возможность в режиме онлайн передавать пользователю данные, запрошенные из сторонних источников. Также сервер приложений может быть использован для агрегации данных из различных внешних систем, выполнения сложных вычислений и других задач. Для разработки данного сервера приложений также использовались технологии Microsoft®.NET и язык программирования C#. Технология работы сервера приложений была апробирована при создании ряда приложений, отображающих на карте ряд точек, координаты которых передаются на клиент по его запросу.

2.3. Генератор

Генератор по предоставленным ему данным создает файл приложения под необходимую мобильную платформу. Сформированный файл в дальнейшем сервер передает клиенту. Полученный файл пользователь может загрузить в телефон и установить приложение.

В данный момент реализована генерация в две популярные мобильные платформы: Android и Windows Phone 7/8. Планируется возможность генерации и в другие мобильные платформы.

² Домашняя страница jQuery Mobile, URL: <http://jquerymobile.com/>

Принцип работы генератора следующий. Сначала сервер передает ему созданный клиентом XML файл специального формата единого для всех мобильных платформ. Для генерации приложения под любую платформу достаточно лишь одного этого файла. В полученном файле для всех элементов пользовательского интерфейса генерируются соответствующие узлы в файлах XML для Android и XAML для Windows Phone. Необходимые ресурсы скачиваются из сети интернет. Для узлов, отображающих логику, генерируются необходимые строки в файлы исходного кода. При необходимости также добавляются нужные для работы приложения подключения библиотек. После прохождения всех узлов происходит автоматическая сборка приложения. Для сборки в Android необходимы утилита Apache Ant и android SDK. Для Windows Phone необходимы Windows Phone SDK и msbuild. На выходе для Android создается файл с расширением apk, а для Windows Phone — хар-файл. На этом работа генератора заканчивается.

Для сохранения единого стиля проекта генераторную часть было решено также писать на языке F#. Сопоставление шаблонов в этом языке позволяет удобно работать со стандартными парсерами библиотеки классов платформы .NET и дает возможность быстро добавить разбор новых узлов. Кроме этого, скачивание нужных ресурсов из сети интернет и запись данных в файл описывается всего в несколько строчек.

Заключение

В результате данной работы были изучены существующие подходы к визуальному созданию мобильных приложений. В результате проведенного анализа было разработано собственное решение, описанное в данной работе. Также был реализован прототип онлайн конструктора приложений для мобильных телефонов, требующих возможности задания клиентской логики.

В дальнейшем планируется реализация возможности агрегации данных в приложении из различных источников, поддержка задания более сложной клиентской и серверной логики и использование специализированных графических языков.

Литература

1. *Мартынов Д.Н., Федотов А.Г.* Windows Azure: Облачная платформа Microsoft, 2010. URL: http://download.microsoft.com/documents/rus/msdn/Windows_Azure_web.pdf
2. *Мартин Фаулер*, Архитектура корпоративных программных приложений, Вильямс, 2007. 544С

РАЗРАБОТКА И РЕАЛИЗАЦИЯ X-CLIENT ДЛЯ ОС WINDOWS

Малыгин А.О.

*студент 5-го курса, кафедры Информатики,
математико-механического факультета СПбГУ, carbon2320@gmail.com*

Научный руководитель

Баклановский М.В.

Аннотация: X11 — протокол позволяющий передавать окна приложений, которые запускаются на удаленном компьютере с помощью X-Client'a, а так же принимать и пробрасывать события на локальном X-Server'e.

Данный документ представляет собой описание реализации X-Client'a для Windows OS, эвристические методы решения и рассмотрение случаев необходимости использования тонкого клиента в целом.

Введение

X Window System — открытое, платформо-независимое, «серверно-клиентское» программное обеспечение (более известное как X11[1]), которое позволяет управлять оконным графическим интерфейсом пользователя в распределенных сетях по протоколу X11.

Хочется акцентировать внимание на том, что использование концепции клиент-серверной модели отлично от обычного представления. В X11 используются следующие термины:

- X-Client в концепции клиент-серверной модели, означает сервер, который, в свою очередь, запускает приложение и передает управление на X-Server;
- X-Server в концепции клиент-серверной модели, означает клиент, который принимает управление от X-Client'a и занимается следующими задачами:
- построение окна и передача оконному менеджеру для отображения;
- обработка событий мышки и клавиатуры, передача их на X-Client;

Во многих Unix подобных системах используется протокол X11, как основной. А для отображения окон используются оконные менеджеры. Например, в GNOME[6], используют «Metacity»[7], а в Mac OS используют QUARTZ[9], как одни из стандартных оконных менеджеров.

Постановка задачи

Многим пользователям доводилось работать с неизвестными, только что загруженными, непроверенными приложениями, которые часто вы-

зывают недоверие и заставляют беспокоиться о конфиденциальности тех данных, которые хранятся у них на компьютере. При установке таких приложений, стоит задуматься о целостности информации. Даже, если у вас есть антивирус, не стоит думать, что базы данных антивируса обновляются раньше, чем создается очередной, новый вирус. Отсюда возникает задача безопасности целостности системы в целом.

Предлагается запускать приложения на удаленной, (возможно, виртуальной), машине. При помощи X-Client'a передавать окно приложения на локальную машину, а при помощи X-Server'a рисовать его и передавать происходящие события мышки и клавиатуры обратно. Тем самым, мы получаем, подобие песочницы, которая гарантирует нам целостность наших данных.

На текущий момент, по протоколу X11 возможно передавать окна следующим образом в следующих операционных системах:

1. С Linux на Linux, т.к. протокол X11 лежит в основе системы;
2. С Linux на Windows OS, т.к. существуют решения X-Server'a для Windows OS (Например, MobaXTerm[8]);

Но, не были реализованы следующие варианты, поскольку в основе Windows OS не лежит протокол X11:

1. С Windows на Linux
2. С Windows OS на Windows OS

Хочется отметить, что в данном случае рассматриваются отдельные окна одного приложения, а не весь рабочий стол в целом со всеми имеющимися приложениями. Это делает клиент тонким, с точки зрения приложения.

На рис.1 схематично изображено описанное выше представление возможностей передачи окна приложения.

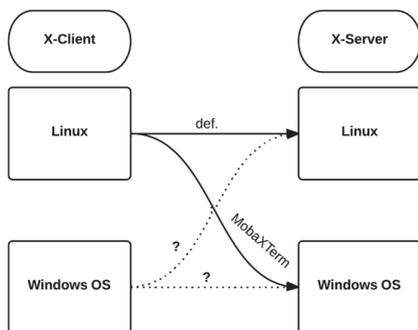


Рис. 1. Реализованные решения передачи окна по протоколу X11

Стоит отметить, что реализация такого X-Client'a для Windows OS, позволит работать с разными типами приложений, запущенных на разных операционных системах, например, таких как Windows 3.1, Windows XP, Windows 7, Linux (рис.2).

А так же, X-Client'a удобно использовать в разделенном рабочем пространстве.

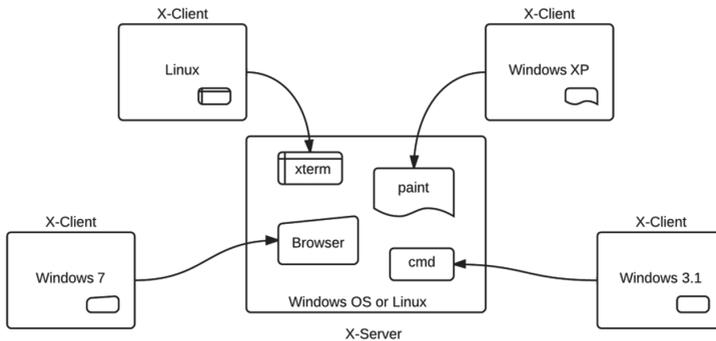


Рис. 2. Демонстрация тонкого клиента и отображения приложений с разных операционных систем

Реализация X-Client'а для Windows OS

В первую очередь, было изучено большое количество информации по существующим решениям. Вследствие чего, было найдено следующее проприетарное ПО: «Citrix MetaFrame»[2]. После этого, была получена информация из официального сообщества разработчиков X11[1], о том, что можно реализовать свое оконное приложение, используя протокол X11 и библиотеку xlib[4] для Windows OS. Отсюда возник вопрос о таком приложении, которое сможет передавать любые окна по этому протоколу и появилась идея реализации универсального X-Client для Windows OS. Поскольку, переписывание всех существующих приложений на X11 не представляется возможным, а также изменения GDI[3] на X11 без исходного кода, является трудоемкой задачей, то было принято решение передавать фреймовый буфер.

Отсюда возникли следующие задачи:

1. Захват изображения приложения;
 - a. Получение окна приложения;
 - b. Удаление рамки приложения;
 - c. Удаление прозрачности приложения;
 - d. Захват многооконного приложения;
 - e. Отображение курсора;
2. Передача фрейма по протоколу X11;
 - a. Сжатие изображения;
 - b. Создание окна из изображения;
 - c. Передача пустого курсора;
 - d. Расчет размеров окна;
3. Получение и обработка событий X-Server'а;
 - a. Получение кликов мышки;
 - b. Пересчет координат мышки;

- c. Получения символов клавиатуры;
- d. Получение escape последовательностей клавиатуры;
- e. Подбрасывание событий в приложение;

Замечу, что многие из задач трудно разрешимы, как для разных версий Windows OS, так и для одной. Многие методы решения являются эвристическими. Например, для разных приложений Windows OS используются разные стили окна, скажем, тройная рамка или одинарная, а может и двойная. На помощь приходит Windows API[5], который позволяет настроить стили приложения. А также отсутствие иерархии окон в Windows OS задают неразрешимость и решаются только эвристическими методами.

На данный момент был реализован тестовый X-Client для Windows OS на C#, который позволяет находить все открытые, не свернутые, оконные приложения, представлять их в виде списка, для удобства просмотра и выбора окна для передачи на X-Server. А так же, приложение позволяет менять стили окна и убирать рамку и сохранять фрейм. На языке C было реализовано сжатие фрейма, передача его на X-Server и обработка событий клавиатуры. При помощи очереди сообщений, события подбрасываются в буфер приложения с текущим хендлером и обрабатываются им. Так же, реализован полуавтоматический алгоритм передачи фреймов, который в дальнейшем будет полностью автоматизирован.

В ближайших планах, реализовать автоматизацию обновления фреймов и обработку escape последовательностей клавиатуры, а так же, обработку событий мышки и пересчет размера окна.

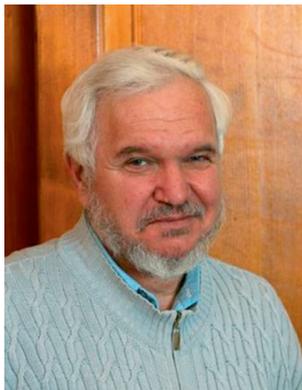
Заключение

В документе были представлены общие сведения о X Windows System, основные понятия этой системы, задачи и возможные решение безопасности конфиденциальных данных, а так же, план реализации X-Client'а для Windows OS.

Литература

1. Сообщество разработчиков X11 <http://www.x.org>
2. Citrix MetaFrame <http://www.citrix.com/>
3. GDI <http://msdn.microsoft.com/en-us/library/ms533797.aspx>
4. Xlib <http://tronche.com/gui/x/xlib/>
5. Windows API
[http://msdn.microsoft.com/en-us/library/windows/desktop/ff818516\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ff818516(v=vs.85).aspx)
6. GNOME <http://www.gnome.org/>
7. Metacity <http://xwinman.org/metacity.php>
8. MobaXTerm <http://mobaxterm.mobatek.net/>
9. QUARTZ <http://quartz-scheduler.org/>

Фундаментальная информатика



**Косовский
Николай Кириллович**

д.ф.-м.н., профессор
заведующий кафедрой информатики СПбГУ



**Герасимов
Михаил Александрович**

к.ф.-м.н., доцент кафедры информатики СПбГУ

ДОСТАТОЧНЫЕ УСЛОВИЯ СХОДИМОСТИ НЕСИНХРОНИЗИРУЕМЫХ ИТЕРАЦИОННЫХ МЕТОДОВ

Буянов С.В.

*Православный Свято-Тихоновский гуманитарный университет,
svbuyanov@mail.ru*

Одной из важных характеристик параллельных вычислительных систем является их масштабируемость, т. е. способность системы наращивать свою производительность при добавлении ресурсов. Однако при увеличении числа процессоров в параллельных системах часто наблюдается снижение роста производительности, вызванное увеличением непродуктивных (накладных) расходов при решении задач параллельными методами. Эксплуатация параллельных вычислительных систем показывает, что при организации параллельных вычислений существует критическое число процессоров, которые работают одновременно, в то время как другие простаивают из-за ожидания данных. Для того, чтобы избежать снижения роста производительности можно использовать несинхронизируемые вычисления. Идея несинхронизируемых вычислений пришла из проектирования цифровых систем управления.

Для осуществления управления объектом реальном масштабе времени необходимо формировать управляющий сигнал с заданным периодом, который задается, исходя из требуемого качества управления объектом. В управляющей вычислительной системе процессоры работают циклически и независимо друг от друга. Например, на борту летательного аппарата имеется несколько параллельно работающих процессоров. Они независимо друг от друга обрабатывают сигналы, получаемые от собственных датчиков. При этом каждый процессор сравнивает свои сигналы и результаты их обработки с сигналами других, контролируя их работу. Каждый процессор в начале каждого цикла обработки данных использует те данные, которые он имеет в момент начала вычислений. Если новых данных нет, то он используют старые. Таким образом исключаются простои из-за ожидания данных.

Как показано ранее несинхронизируемые вычисления могут быть успешно использованы для организации параллельных итерационных вычислений в гетерогенных вычислительных системах, когда производительность процессоров существенно различается.

В докладе обсуждаются подходы к разработке математических моделей для описания и анализа несинхронизируемых вычислений, а также для поиска достаточных условий сходимости несинхронизируемых итерационных методов.

РАЗРАБОТКА И РЕАЛИЗАЦИЯ БЫСТРЫХ АЛГОРИТМОВ КЛАССИФИКАЦИИ ТЕКСТОВ ОПУБЛИКОВАННЫХ ONLINE

Заболотский Е.Д.

Задача классификации текстов — одна из задач информационного поиска, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа.

Задача классификации документов является подзадачей задачи Data Mining. **Data Mining** (добыча данных, интеллектуальный анализ данных) — собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности. Термин введен Григорием Пятецким-Шапиро в 1989 году.

Определения классификации:

Классификация — системное распределение изучаемых предметов, явлений, процессов по родам, видам, типам, по каким-либо существенным признакам для удобства их исследования; группировка исходных понятий и расположение их в определенном порядке, отражающем степень этого сходства.

Классификация — упорядоченное по некоторому принципу множество объектов, которые имеют сходные классификационные признаки (одно или несколько свойств), выбранных для определения сходства или различия между этими объектами.

Существующие алгоритмы классификации текстов, которые чаще всего используются:

- Ранжирование и четкая классификация;
- Метод Роше (Rocchio method);
- Деревья решений (Decision Tree);
- Метод опорных векторов (Support Vector Machine, SVM);
- Метод k-ближайших соседей (k-Nearest Neighbors, k-NN);
- Метод «случайный лес» (Random Forest);
- Метод Байеса (Naive Bayes).

Для решения задачи классификации текстов был выбран метод частотного анализа. Частотный анализ основывается на предположении о существовании нетривиального статистического распределения отдельных слов и их последовательностей в тексте. Частотный анализ предполагает, что частота появления заданного слова алфавита в достаточно длинных текстах одна и та же для разных текстов одного языка.

Постановка задачи.

- Существует множество категорий;
- Существует множества текстов по каждой категории;
- Имеется текстовый файл;
- Необходимо определить категорию текстового файла.



Определение категорий

В качестве категорий выступают 16 основных направлений деятельности Правительства Российской Федерации:

- Здравоохранение;
- Культура;
- Природные ресурсы и экология;
- Внутренние дела (МВД);
- Иностранные дела (МИД);
- Образование и наука;
- Промышленность и торговля;
- Спорт;
- Связь и массовые коммуникации;
- Энергетика;
- Финансы;
- Транспорт;
- Труд и социальная защита;
- Сельское хозяйство;
- Оборона;
- Экономическое развитие.

В качестве текстов, определяющих категории, используются нормативно-правовые акты Российской Федерации, регулирующие отношения в ответственном направлении деятельности.

Таким образом, построение категорий происходит на основании нормативно-правовых актов Российской Федерации.

Нормативно-правовой акт.

- официальный документ установленной формы, принятый в пределах компетенции уполномоченного государственного органа с соблюдением установленной законодательством процедуры, содержащий общеобязательные правила поведения, рассчитанные на неопределённый круг лиц и неоднократное применение;
- имеют определённый вид и облакаются в документальную форму;

- составляются по правилам юридической техники;
- нормативные правовые акты, действующие в стране, образуют единую систему.

Определение категорий происходит в соответствии с выборкой нормативно-правовых актов, относящихся к соответствующим категориям.



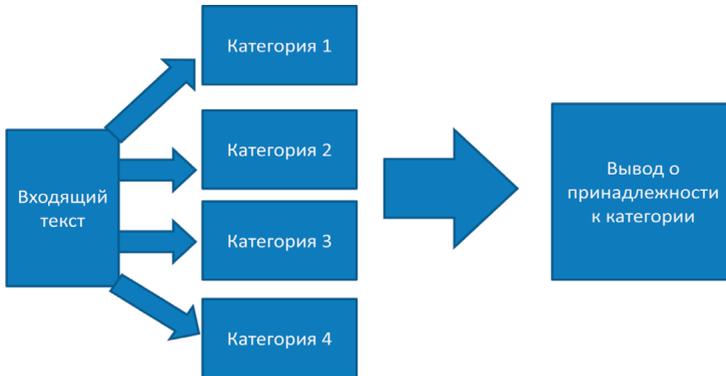
Анализ входящего текста

- Исключение частиц, предлогов, местоимений и слов длиной менее трех символов
- Нормализация слов
- Определение частоты появления каждого слова в тексте



Сравнение

- Используется метод частотного анализа;
- Каждая категория — список слов с соответствующими частотами вхождения;
- Определяется вес каждого слова во входящем текстовом файле;
- Сравнение происходит при совпадении более пяти слов.



Алгоритм сравнения

$$X = \frac{a_{ij} * b_j}{\sqrt{\sum (a_{sj} * b_j)^2}}$$

$$a_{ij} = \frac{t_i}{\sum_{l=1}^T t_l} \quad b_j = \log \frac{|D|}{n_i}$$

- X — функция соответствия
- a — отношение числа вхождения слова к общему количеству слов
- b — инверсия частоты, с которой слово встречается в документах категорий
- D — количество документов в категории
- n — количество документов, в которых встречается слово

Используемые средства

- В качестве программных средств для решения задачи были выбраны язык Java и средства разработки Eclipse;
- Для хранения и обработки текстовой информации необходимы база данных и СУБД. База данных MySQL имеет хорошие показатели быстродействия, но ограничена в создании очень больших БД. Поскольку потенциально объем словарей лингвистической БД может быть очень большим, была выбрана СУБД MS SQL Server.

Литература

1. *Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze* An Introduction to Information Retrieval Draft
 2. *Chisholm E., Kolda T.G.* New term weighting formulas for the vector space method in information retrieval. Technical Report Number ORNL-TM-13756, Oak Ridge National Laboratory, Oak Ridge, TN, March 1999
 3. *Lan M.* (2007) A New Term Weighting Method for Text Categorization. PhD Theses
 4. *F. Sebastiani.* Machine Learning in Automated Text Categorization
 5. *Дюк В., Самойленко А.* Data Mining
 6. *Lancaster, F.W.* (2003). Indexing and abstracting in theory and practice. Library Association, London
 7. *Jiawei Han and Micheline Kamber.* Data Mining: Concepts and Techniques
 8. *Ian H. Witten, Eibe Frank.* Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations
 9. Large Scale Hierarchical Text classification (LSHTC) Pascal Challenge <http://lshtc.iit.demokritos.gr/>
 10. *Xiao-Lin Wang, Bao-Liang Lu.* Improved Hierarchical SVMs for Large-scale Hierarchical Text Classification Challenge
 11. *Masand B., Linoff G., Waltz D.* Classifying news stories using memory-based reasoning. Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 1992, pp. 59–65
 12. *Salton G., Buckley C.* (1988), Term-weighting approaches in automatic text retrieval, Information Processing & Management, Vol. 24, no. 5, pp. 513–523
 13. *Дунаев Е.В.* Автоматическая рубрикация web-страниц в интернет-каталоге с иерархической структурой / Е.В. Дунаев, А.А. Шелестов // Интернет-математика 2005. Автоматическая обработка веб-данных. — М., 2005. — С. 382–398
 14. *Dumais S., Platt J., Heckerman D., Sahami M.* Inductive learning algorithms and representations for text categorization. // In Proc. Int. Conf. on Inform. and Knowledge Manage., 1998
-

WEB-ПРИЛОЖЕНИЕ ДЛЯ РАБОТЫ С ИЗОБРАЖЕНИЯМИ БИМЕДИЦИНСКОГО ХАРАКТЕРА

Филин А.Ю.

асп. кафедры информатики СПбГУ

Соловьев И.П.

к. ф.-м. н., доцент кафедры информатики СПбГУ

Аннотация: Web-приложение предоставляет пользовательский интерфейс для обработки изображений биомедицинского характера. Приложение разработано на современной масштабируемой платформе Google App Engine, что существенно упрощает разработку и поддержку возможностей программы.

Введение

Основной задачей данной работы является разработка Web-приложения, которое позволило бы пользователям работать с изображениями биомедицинского характера, применять различные методы выделения признаков на таких изображениях и сохранять результаты своей работы. Возможность обрабатывать изображения указанного типа позволит специалистам быстрее и эффективнее решать прикладные задачи, например, постановку диагноза или выявление предрасположенности пациента к болезни [1].

Основные функции приложения

На данный момент существует множество методик и подходов для анализа изображений. Эти методы рассматривают самые разные характеристики изображений. Среди наиболее известных следует упомянуть статистические, морфологические, аналитические, спектральные.

Важной составляющей исследования любого метода обработки изображений является эффективная и доступная программная реализация. Лидирующими современными платформами для построения больших приложений являются Java и .NET. Основными преимуществами указанных технологий являются мультиплатформенность и большой выбор различных инструментов для решения широкого круга задач.

Важными особенностями любого исследования являются доступность и наглядность достигнутых результатов. В данный момент существует множество технологий, которые помогают строить мощные Web-приложения, доступные для любого пользователя через интернет.

Web-приложение для данной работы строится на платформе Google App Engine, которая предоставляет гибкую и настраиваемую инфраструктуру для

Web-приложений, написанных на языке Java. Среди основных преимуществ указанной платформы выделим возможность быстро разрабатывать и развертывать свои приложения, не заботясь об аппаратных средствах, исправлениях и резервном копировании. Google App Engine зарекомендовала себя как отличная площадка для построения Web-приложений любой сложности: от простых информационных порталов [5] до сложных корпоративных приложений.

Разрабатываемое автором приложение предоставляет пользователю следующие возможности:

- личный профиль, в котором хранится история и статистика по обработанным изображениям,
- возможность поделиться своими результатами с другими пользователями;
- ресурсы, демонстрирующие различные схемы и алгоритмы работы методов анализа и обработки биомедицинских изображений;
- возможность загрузить собственное изображение и получить необходимые, выбранные пользователем характеристики, выделяемые с помощью методов анализа изображений в удобном для пользователя виде.

Приложение позволяет загружать пользователям исследуемые изображения на сервер. После этого происходит этап обработки изображения, на котором с помощью выбранных методов анализа выделяется набор соответствующих признаков, которые в итоге отображаются пользователю на специальной странице. Полученные данные можно сохранить для дальнейшего использования.

Заключение

Разработано Web-приложение, которое предоставляет пользователю возможность загружать свои изображения, применять к ним различные методы анализа и сохранять полученные результаты. Web-ориентированность приложения позволяет добиться большей наглядности и доступности результатов каждого из пользователей.

Литература

1. *Soloviev, N. Ampilova, E. Gurevich.* On implementation of a neural network classifier for some classes of biological and medical preparation images. Proc. 7 Int. Conf. SEMA12, 8-10 Nov. 2012, Athens, Greece. p.94–97.
2. *Пытьев Ю.П., Чуличков А.И.* Методы морфологического анализа изображений, М. : ФИЗМАТЛИТ, 2010. — 342 с.: ил. ISBN 978-5-9221-1225-3.
3. *Serra J.,* Image analysis and mathematical morphology, London : Academic Press Inc., 1982 — 621 с., ISBN 0-12637240-3.
4. Документация по платформе Google App Engine [Электронный ресурс] : Режим доступа: <https://developers.google.com/appengine/docs>.
5. Страница сайта о защите и возможных типах атак Web-приложений [Электронный ресурс] : <http://google-gruycere.appspot.com/>

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ОБОБЩЕННЫХ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

М.Л. Симуни

Санкт-Петербургский государственный университет

Существует большое количество средств и формализмов для компактной записи типичных схем рекурсии в функциональных программах. В первую очередь, это свертка (функции `foldr` и `foldl` в языке Haskell) [1, 2]. Обобщением свертки для более сложных схем рекурсии являются понятия катаморфизма, анаморфизма, хиломорфизма и параморфизма [3]. Формализм Берда-Меертенса [4], позволяет не только записывать программы с использованием набора стандартных схем рекурсии, но и применять к ним различные преобразования.

Однако все эти средства не получили широкого распространения при практическом программировании на функциональных языках. Даже функция свертки на практике применяется сравнительно редко. Одной из причин этого является, видимо, недостаточное удобство и понятность записи.

В работе предлагается еще один способ представления рекурсивных схем, по своим возможностям эквивалентный понятиям катаморфизма и анаморфизма, но обеспечивающий более гибкую и компактную запись. Основной идеей рассматриваемого формализма является использование понятия регулярного выражения [5], обобщенного для применения его в функциональных программах.

Пример расширенного регулярного выражения

Рассмотрим простой пример: программу на языке Haskell [2] для вычисления суммы элементов списка:

```
sum (x:xs) = x + sum xs
sum [] = 0
```

С использованием свертки эта программа может быть записана так:

```
sum = foldr (+) 0
```

В данном случае свертка обеспечивает очень простую и удобную запись. Однако уже для чуть более сложной задачи: «Найти сумму всех положительных элементов списка» запись со сверткой уже не так интуитивно понятна, и, как показывает практика, вызывает затруднения у недостаточно опытных программистов:

```
sumpos = foldr (\x y -> if x>0 then x+y else y) 0
```

Если рассмотреть еще немного более сложную задачу: «Найти сумму всех положительных элементов списка, умноженную на сумму всех отрицательных элементов», то запись становится еще более сложной:

$$f = \text{foldr } (\backslash x \ y \ -> \text{if } x > 0 \text{ then } x + y \ \text{else } y) \ 0 + \\ \text{foldr } (\backslash x \ y \ -> \text{if } x < 0 \text{ then } x + y \ \text{else } y) \ 0$$

Рассмотрим примеры записи этих функций с использованием предлагаемого формализма:

$$\text{sum } (\{x:\} * []) = \{x+\} * 0$$

Пояснения:

- Запись $\{\text{выражение}\}^*$ означает выражение, повторенное 0 или более раз. Эта нотация является аналогом операции $*$ для регулярных выражений. В качестве выражения используется, как правило, функция от одной переменной. В данном примере это функции $(x:)$ и $(x+)$.
- Важным отличием от обычных регулярных выражений является поддержка *частичного задания* регулярных выражений. В данном примере конструкции $\{x:\}^*$ и $\{x+\}^*$, находящиеся в разных частях правила, задают в совокупности одно обобщенное регулярное выражение. Неформально можно сказать, что регулярные выражения в правой и левой части вычисляются параллельно: каждый экземпляр $x:$ появившийся при сопоставлении с образцом — регулярным выражением в левой части порождает соответствующий вызов функции $(x+)$ в правой части.

$$\text{sumpos } (\{x:\} * []) = \{\text{if } x > 0 \text{ then } (x+)\}^* 0$$

Пояснение: в данном случае полной записью в правой части было бы $\{\text{if } x < 0 \text{ then } (x+) \ \text{else } \text{id}\}^*$, однако else id можно не писать, что делает запись более простой.

$$\text{sumpos } (\{x:\} * []) = \{\text{if } x > 0 \text{ then } (x+)\}^* 0 / \\ \{\text{if } x < 0 \text{ then } (x+)\}^* 0$$

Другие примеры и более сложные конструкции

Важным специальным случаем рекурсивной схемы является рекурсия по целым числам, соответствующая понятию примитивной рекурсии в теории вычислимости. Вот простой пример:

```
repeat 0 a = []
repeat n a = a:repeat (n-1) a
```

Эта функция генерирует список данной длины, состоящий из одного и того же числа a . С использованием обобщенных регулярных выражений эту функцию можно записать очень коротко:

```
repeat ({1+}*0) a = {a:}*[]
```

Пояснение: запись $\{1+\}^*0$ в левой части правила порождает рекурсивную схему, при которой их аргумента вычитается 1 несколько раз, пока аргумент не окажется равным 0. Таким образом, обеспечивается функциональность, соответствующая понятию анаморфизма в [3].

Более сложным является определение функции факториал. Проблема заключается в том, что просто использовать выражение $\{1+\}^*0$ невозможно, поскольку нам надо использовать в рекурсивной схеме текущее значение переменной. В [3] для записи такой рекурсивной схемы вводится понятие параморфизма. В такой ситуации можно применить конструкцию `let`, расширенную для использования в регулярных выражениях.

```
fact ({let i = 1+}*0) = {i*}*1
```

Еще одним интересным примером является определение функции `zip`. Приведем рекурсивное определение:

```
zip (x:xs) (y:ys) = (x,y):zip xs ys
zip _ _ = []
```

С использованием обобщенных регулярных выражений мы получаем очень компактную запись:

```
zip ({x:}*_) ({y:}*_) = {(x,y):}*[]
```

Конструкция $\{ \ }^*$ может использоваться вложено, что позволяет компактно записывать более сложные рекурсивные программы. В качестве примера приведем запись функции, которая для данного списка списков:

- вычисляет в каждом внутреннем списке сумму положительных чисел
- вычисляет максимум сумм по всем спискам.

```
maxSumsPos ({(x:}*[]):}[] =
  {max ({if x<0 then (x+)}*0)}*0
```

Формальные определения

Обобщенным регулярным выражением мы называем набор конструкций вида { *регулярное подвыражение* }*, где *регулярное подвыражение* может быть одной из следующих конструкций:

- функция от одной переменной
- специальный вид if выражения:
if *условие* then *регулярное подвыражение*
- специальный вид let выражения:
let *выражение* = *регулярное подвыражение*

Все такие конструкции, входящие в одно правило, образуют одно определенное регулярное выражение.

Заключение

В работе получены следующие результаты:

- Введено понятие обобщенного регулярного выражения для функциональной программы.
- Показано, что обобщенные регулярные выражения можно использовать для описания рекурсивных функций. При этом достигается функциональность, аналогичная таким формализмам, как катаморфизм/анаморфизм/параморфизм и формализму Берда-Меертенса.
- Показано, что обобщенные регулярные выражения могут быть эффективно преобразованы в рекурсивную функциональную программу.
- Создан прототип такой программы, преобразующий программу с использованием расширенным регулярных выражений в программу на языке Haskell.

Литература

1. *Бёрд Р.* Жемчужины проектирования алгоритмов: функциональный подход. ДМК Пресс, 2013. 330 с.
2. *Липовача М.* Изучай Haskell во имя добра! ДМК Пресс, 2012. 490 с.
3. *Meijer E., Fokkinga M., Paterson R.* Functional Programming with Bananas, Lenses, Envelopes and Barbed Wire. // Proceedings of the 5th ACM conference on Functional programming languages and computer architecture, 1991, Cambridge, MA, pp. 124–144
4. *Bird R., de Moor O.* The Algebra of Programming. Prentice Hall, 1997, 312 p.
5. *Ахо А., Лам М, Сети Р., Ульман Д.* Компиляторы: принципы, технологии и инструментарий. М.: Вильямс, 2008. 1184 с.

РЕАЛИЗАЦИЯ ПАКЕТА ПРОГРАММ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ВИДЕОИНФОРМАЦИИ СРЕДСТВАМИ ВЗАИМОДОПОЛНЯЕМЫХ ИСТОЧНИКОВ

А.А. Овсянников

аспирант мат-мех факультета СПбГУ

Актуальность проблемы

В последние годы наблюдается стремительный рост развития мобильных технологий. При этом следует учитывать, что при большом разнообразии мобильных средств, растет и разобщённость аппаратных особенностей каждого. В таких случаях появляется необходимость создавать огромное количество форматов для видеoinформации, подходящих для определенных моделей мобильных устройств. Такой подход не вполне удовлетворителен, если учесть, что в некоторых случаях необходимо использовать информацию, полученную об одном объекте или событии, но с большого количества разнооснащенных источников. Но при этом стоит заметить, что появляется возможность использования информации из таких источников для ее качественного дополнения. Например, для повышения разрешения видеоизображения на устройствах, не оснащенных необходимыми аппаратными средствами. Либо дополнять видео геолокационными данными.

Помимо проблемы мультимедийности, существует так же проблема передачи и хранения. Одним из основных способов передачи на сегодняшний день является интернет, но его использование сопряжено с определенными проблемами:

– Отсутствие связи.

Даже сегодня существуют области, в которых отсутствует связь по причине удаленности от крупных населенных пунктов (так как рассматривается только передача средствами мобильных или домашних сетей).

– Обрывы.

Некоторые мобильные сети имеют недостатки в защите от обрывов при движении.

– Рост вероятности появления ошибок с увеличением объема передаваемых данных.

Никак нельзя гарантировать 100% совпадение файла попавшего на источник и файла полученного на приемник.

В связи с вышеизложенным, актуальна задача разработки комплекса программ для оптимизации времени обработки, дополнения и использования видеoinформации с учетом аппаратных средств устройства, а так же создание общего формата.

Постановка задачи:

Целью является разработка системы для:

- успешного решения проблемы мультиформатности;
- создание единого формата хранения видеоинформации;
- улучшение качества видеоинформации
- минимизации ошибок при передаче.

Кроме того требуется разработать алгоритмы передачи больших объемов данных средствами мобильных сетей, так как пропускная способность некоторых видов связи (например EDGE ограничена).

За счет использования разнооснащенных мобильных устройств возможно качественно улучшать видеоинформацию.

Существующие системы обработки видеоинформации

Статистика, основанная на исследовании Гатнера (Gartners) показывает, что если в 2006 г. только 48% мобильных устройств было оснащено камерами, то к 2010 г. их стало 86% [«A Camera-based Mobile Data Channel: Capacity and Analysis» Xu Liu, David Doermann, Huiping Li]. Такой рост может означать, что в скором будущем многие технологии, основанные на использовании специализированных устройств (камеры слежения, авторегистраторы и т.д.), будут интегрированы на телефоны. Следует отметить, что использование мобильных средств (устройств) в проблемах обработки и хранения видеоинформации развита на сегодняшний момент в недостаточной степени. Следовательно, для анализа можно рассматривать существующие системы создания и обработки видеоинформации, такие как системы видеонаблюдения.

VOCORD Phobos и VOCORD Tahion

В цифровых системах видеонаблюдения, в отличие от аналоговых, есть возможность повлиять на качество изображения за счет применения методов цифровой обработки изображения. Эти методы позволяют получать изображение высокого качества при использовании не слишком хороших камер или при неблагоприятных условиях съемки.

Специально для систем VOCORD Phobos и VOCORD Tahion разработан аппаратно-программный комплекс, который решает задачи автоматической регулировки параметров изображения.

APU «ЯК»

Одна из проблем, с которыми постоянно сталкиваются разработчики и пользователи цифровых систем видео-наблюдения — ограниченный динамический диапазон регистрируемого изображения. При этом для сцен с

широким динамическим диапазоном характерной чертой является потеря деталей в затемненных или в ярких участках изображения. Не решают всех проблем и встроенные в камеры стандартные средства автоматического управления уровнем сигнала (АРУ).

Более того, наличие стандартных АРУ в камере может способствовать развитию новых проблем, поскольку они только поддерживают среднее значение сигнала, что при определенных условиях может ухудшить изображение. Например, появление в поле зрения сильно освещенного объекта приводит при использовании камер со стандартными АРУ к ухудшению отображения плохо освещенных областей картинки. С этой проблемой можно справиться при помощи настройки вручную, но это, во-первых, отвлечет оператора, что особенно скажется при работе с многоканальными охраняемыми системами, а также внесет определенный субъективизм в настройку системы, а значит и в финальные показания.

Полученные результаты

На сегодняшний день разработана система Castarama. Это облачный сервис (то есть хранение информации происходит не на самом устройстве, а на сервере) позволяющий получать доступ к видеoinформации из любой точки мира. Помимо этого была реализована возможность отправки видеосообщения между пользователями.

Видео дополняется данными о расположении за счет возможности получения координат GPS модуля, встроенного в большинство мобильных устройств.

Так же реализована развлекательная часть. То есть существует возможность записи музыкальных видео. Это стало возможно благодаря стримингу музыкального потока на клиентской части системы и склеивания на серверной. На серверной стороне реализован следующий алгоритм: сервер анализирует звуковую дорожку, разбивает ее на участки, на участках анализирует локальные минимумы и максимумы АЧХ (Амплитудно-частотная характеристика) и анализирует текст исходной песни. В итоге получается смонтированный многопользовательский музыкальный видео-клип.

Таким образом, данный проект является начальным этапом, позволяющим более близко ознакомиться с работой над медиа-ресурсами и опробовать некоторые алгоритмы передачи и хранения данных. Для реализации функционала были разработаны алгоритмы успешно решающие проблемы:

– Отсутствие связи и обрывы

В такой ситуации происходит временное сохранение файла с дальнейшей отправкой при появлении устойчивого канала связи.

Для реализации был использован Приемник широкоэмитательных сообщений (Broadcast receiver). Приемник широкоэмитательных сообщений — это компонент для получения внешних событий и реакции на них. Инициали-

зирать передачи могут другие приложения или службы. Класс Broadcast Receiver является базовым для класса, в котором должны происходить получение и обработка сообщений. Таким образом, каждые 15 минут проверялось состояние сети. При благоприятном изменении сети создавалась сообщение системе, которое обрабатывалось VR и выполнялась очередная попытка отправки файла.

– Рост вероятности появления ошибок с увеличением объема передаваемых данных.

Для решения был реализован алгоритм порционной отправки файла.

Почему разбивается на части?

Если посылать данные непрерывно — любое искажение исходного файла или разрыв соединения, потерянный или не вовремя пришедший пакет потребует повторной полной отправки всего файла. В случае с фрагментами — будет перезапрошено лишь несколько пакетов. Но при разбиении файла на части увеличивается объем передаваемых данных — каждый пакет должен, кроме полезной информации, содержать данные о самом файле и адреса отправителя пакета, получателя пакета, контрольную сумму для проверки целостности файла.

Так как рассматривается мобильная система, то сохранность файла может быть нарушена. В такой ситуации применялся алгоритм, описанный выше (BR).

Данный подход очень похож на решение при использовании вместо мобильной системы ПК. Но, как говорилось ранее, Castarama является только начальным этапом работы.

Рассматривая система была реализована (с клиентской стороны) посредством стандартной связки Eclipse IDE и Android SDK с использованием сторонних библиотек, нацеленных на создание и обработку по протоколу HTTP.

Обзор алгоритма

1. Отправка запроса на сервер.

На данном этапе клиент собирает данные, необходимые для идентификации устройства: Mac-адрес, идентификационный номер пользователя, геолокационные данные, язык устройства и данные о сессии. Сессия – это закодированная строка, необходимая для защиты и для сохранения при неудачной отправке файла (благодаря сессии, даже если отправка произошла гораздо позже запланированного времени, сервер сможет восстановить необходимые для корректной работы системы время создания видео-сообщения). Все это необходимо для определения типа данных, которые необходимо отправить серверу обратно. Существует три типа:

А) Обычная запись

Б) Запись музыкального видео

В) Видео-сообщение

2. Получение данных с сервера.

В зависимости от типа, сервер отправляет обратно: зарезервированный номер в Базе Данных (регистрационный ключ); регистрационный ключ и регистрационный номер музыкального файла (необходим для составления ссылки для стримминга файла); регистрационный ключ, идентификационный номер получателя соответственно. Все эти данные используются в имени создаваемого файла. Это позволит избежать проблем распознавания сервером кусков файла при передаче (будет рассмотрено в шаге 6).

3. Настройка необходимых средств для записи и отправки.

Клиентской стороной производится настройка перед началом записи. В зависимости от типа видео-сообщения происходит создание необходимого файла-хранилища, настройка стримминга музыкального файла и т. д.

4. Запись

Запись производится в зависимости от типа видео-сообщения посредством камеры (камер) мобильного устройства.

5. Временное локальное сохранение

Так как вероятность описанных выше ошибок при передаче файла является существенной проблемой, необходимо временно сохранить файл.

6. Отправка

Как говорилось ранее, файл подвергается разбиению на части и происходит асинхронная (в независимых друг от друга потоках) отправка на сервер. В случае неудачи (разрыв связи, нарушение целостности куска файла) сервер присылает ответ, что произошла ошибка, с указанием кода куска и кодом самой ошибки. В этом случае, создается задача для Broadcast Receiver: создается сервис, который благодаря таймеру повторяет попытки отправки. Так как на шаге 2 в имя файла были указаны необходимые для идентификации данные, то исключаются потери. Но возможно возникновение ситуации, когда ошибка сохранения происходит на сервере. В этом случае, после нескольких неудачных попыток отправки, сервер присылает определенный код ошибки, сигнализирующий, что необходимо повторное получение данных, описанных на шаге 2. Такая ситуация, конечно, малоприятна и требует более тщательного анализа, но разработка более удовлетворительного алгоритма еще только предстоит.

ОБРАТНЫЙ МЕТОД МАСЛОВА И МУРАВЬИНАЯ ТАКТИКА РЕШЕНИЯ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Петухова Н.Д.

*аспирант первого года обучения кафедры математики СПб ГМТУ,
ninka_mat@mail.ru*

Аннотация: Статья посвящена изложению разработанного алгоритма решения задач искусственного интеллекта, допускающих формализацию в исчислении предикатов, с помощью модификации обратного метода Маслова. А так же муравьиной тактике применения этого алгоритма.

Введение

NP-трудность решения задач искусственного интеллекта, в частности, допускающих формализацию в исчислении предикатов [2], накладывает на алгоритмы их решения требования по оптимизации перебора, возникающего при поиске подходящей подстановки, обеспечивающей выполнение целевой формулы. Обратный метод Маслова разработан для эффективизации процедуры построения вывода в исчислении предикатов.

Использование нескольких агентов, одновременно занимающихся решением одной и той же задачи, позволяет найти аналогию их деятельности в действиях муравьёв-разведчиков. Использование такой аналогии в литературе получило название Алгоритм муравья [1]. Эта тактика позволяет «распараллеливать» процесс сбора информации и отсеивать тупиковые ветви поиска решения. В этой статье, на основе применения муравьиной тактики к процессу поиска вывода с помощью обратного метода Маслова, приведена тактика решения задач логико-предметного распознавания образов.

Решение многих задач искусственного интеллекта, допускающих формализацию средствами языка исчисления предикатов, сводится к доказательству формул вида:

$$S(\omega) \Rightarrow \exists \bar{x} \neq A(\bar{x}),$$

где $\omega = \{a_1, \dots, a_k\}$ — набор констант, $S(\omega)$ — набор постоянных атомарных формул или их отрицаний, $A(\bar{x})$ — элементарная конъюнкция вида $P_{k_i}(\bar{x})$ [2]. Такое логическое следование равносильно истинности формулы:

$$(\&S(\omega)) \Rightarrow \exists \bar{x} \neq A(\bar{x}),$$

при любых наборах значений ω , которую, используя равносильные преобразования аппарата математической логики, можно свести к формуле вида:

$$\forall a_1, \dots, a_k \exists (x_1, \dots, x_n) \neq \left(\bigwedge_{i=1}^{\delta} (\bigvee \neg S(a_1, \dots, a_k) \vee P_{ki}(x_1, \dots, x_{ni})) \right) \quad (1)$$

Эта формула выводима тогда и только тогда, когда существуют подстановки термов t_1, \dots, t_{ni} вместо переменных x_1, \dots, x_{ni} такие, что в каждом конъюнктивном члене найдется контрарная пара. То есть требуется найти значения термов, являющихся решениями системы уравнений следующего вида

$$i = \overline{1, \delta} \left\{ \begin{array}{l} \overline{1, s} \left[\begin{array}{l} \{ \\ \vdots \\ \{ \end{array} \right. \\ \vdots \\ \overline{1, s} \left[\begin{array}{l} \{ \\ \vdots \\ \{ \begin{array}{l} t_1 = a_1 \\ \vdots \\ t_{ni} = a_{ni} \end{array} \} \\ \vdots \\ \{ \end{array} \right. \\ \vdots \\ \overline{1, s} \left[\begin{array}{l} \{ \\ \vdots \\ \{ \end{array} \right. \end{array} \right. ,$$

где s — количество атомарных формул в $\neg S(a_1, \dots, a_k)$.

Решение системы такого вида требует экспоненциального числа шагов. Обратный метод ориентирован на существенное сокращение числа шагов решения такой системы.

В дополнение к идеям метода Маслова упорядочим все формулы $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{ki}(x_1, \dots, x_{ni})$ в (1) следующим образом. Так как в каждой $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{ki}(x_1, \dots, x_{ni})$ всего одна формула $P_{ki}(x_1, \dots, x_{ni})$, имеющая переменные, сгруппируем их по одинаковым именам $P_{ki}(x_1, \dots, x_{ni})$, затем упорядочим эти группы в порядке возрастания количества вхождений одноименных предикатных символов в (1). Формулы в $\neg S(a_1, \dots, a_k)$ сначала сгруппируем по именам предикатных символов, затем упорядочим в том же порядке по именам предикатных символов, что и в (1).

Ищем подстановку, которая позволит присвоить значения переменным в наиболее редко встречающемся предикатном символе. Если таковой не на-

ходится, то формула (1) не выводима. Если подстановка найдена, то осуществляем её во всех $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{k_i}(x_1, \dots, x_{n_i})$. Повторяем процедуру до тех пор, пока все значения переменных не окажутся заменены константами.

Алгоритм ИМА (inverse method algorithm) поиска вывода, основанный на Обратном методе

Определение: Пусть список Γ не повторяющихся формул вида $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{k_i}(x_1, \dots, x_{n_i})$ называется **F-набором** для формул вида (1) [3], [4].

Определение: **F-набор** будем называть **пустым** \square , если все формулы, входящие в него не имеют переменных и тавтологичны [3], [4].

Определение: **F-набор** будем называть **тупиковым**, если в него входит хотя бы одна формула, не имеющая переменных и являющаяся ложной или не являющаяся ни тавтологией ни противоречием [3].

1. Строим δ -членный F-набор, формулы в котором не повторяются. То есть переписываем без конъюнкций все дизъюнкты $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{k_i}(x_1, \dots, x_{n_i})$.

2. Присваиваем значения переменным следующим образом:

2.1. Отменяем все пометки об удалениях предикатных формул из $S(\omega)$.

2.2. Берем формулу $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{k_i}(t_1, \dots, t_{n_i})$ из рассматриваемого F-набора, содержащую m -местный предикатный символ $P_{k_i}(t_1, \dots, t_m)$ такой, что набор содержит хотя бы одну переменную.

2.3. Ищем среди предикатов в $S(\omega)$ отрицание $\neg P_{k_i}(v_1, \dots, v_m)$ предикатного символа $P_{k_i}(t_1, \dots, t_m)$ при наборе констант v_1, \dots, v_m , если нашли подходящее отрицание — помечаем его как удаленное и переходим к пункту 2.4., если его не существует — переходим к пункту 3.

2.4. Решаем систему уравнений, отождествляющую списки переменных и констант t_1, \dots, t_m и v_1, \dots, v_m . В случае, если эта система имеет решение¹, перейти к пункту 2.5., если система решений не имеет — перейти к пункту 2.3.

2.5. Заменяем во всем F-наборе переменные из списка t_1, \dots, t_m на их значение, полученные в пункте 2.4.

2.6. В полученном F-наборе удаляем повторения формул.

2.7. Если получился пустой F-набор, то алгоритм заканчивает работу.

2.8. Если получился тупиковый F-набор — перейти к пункту 3.

2.9. Если в F-наборе все формулы, имеющие переменные помечены как удаленные — переходим к пункту 4.

2.10. Если в F-наборе существуют формулы $\bigvee \neg S(a_1, \dots, a_k) \bigvee P_{k_i}(x_1, \dots, x_{n_i})$, имеющие переменные, которым еще не присвоено значение — перейти к пункту 2.2.

¹ В данном случае, помимо прочих, система уравнений не имеет решений, если какое-то из присваиваемых переменным значений, было присвоено ранее другой переменной.

3. Возвратная часть алгоритма.

3.1. Отменяем последнее действие пункта 2.5., если это возможно, и переходим к пункту 2.3.

3.2. Если отмена последнего действия пункта 2.5. невозможна — помещаем $P_{k_i}(t_1, \dots, t_m)$ как удаленный и переходим к пункту 2.

4. Если все формулы помечены как удаленные значит, формула не выводится. Алгоритм заканчивает работу.

Алгоритм муравья

Алгоритм муравья является относительно новым подходом к проблеме решения задач искусственного интеллекта. Действия муравьев послужили вдохновением для ряда методик, среди которых наиболее успешным является техника, известная как оптимизация колонии муравьев [1].

Настоящие муравьи-разведчики наносят феромон на землю для того, чтобы отметить благоприятные пути, по которым должны следовать другие члены колонии. Самое удивительное в данном процессе — это то, что муравьи умеют находить самый оптимальный путь между муравейником и внешними точками. Чем больше муравьев используют один и тот же путь, тем выше концентрация ферментов на этом пути. Чем выше концентрация ферментов на пути, тем предпочтительнее он для муравьев по сравнению с другими доступными.

Так муравьиная «логика» позволяет выбирать более короткий путь между конечными точками. Алгоритм муравья использует такой механизм для решения оптимизационных задач. Муравьи легко вступают в сотрудничество и работают вместе для достижения общей цели. Алгоритмы муравья работают так же, как муравьи. Это выражается в том, что смоделированные муравьи совместно решают проблему и помогают другим муравьям в дальнейшем поиске оптимального решения.

Алгоритмы муравья являются итерационными. На каждой итерации учитываются действия искусственных муравьев. Каждый из них строит решение, совершая различные действия и не повторяя дважды одно и то же действие. На каждом шаге муравей выбирает следующее действие в зависимости от количества феромонов, которым помечены действия перед их выполнением.

Применение муравьиной тактики

Муравей — это программный агент, который является членом большой колонии и используется для решения какой-либо проблемы. Муравей снабжается набором простых правил, которые позволяют ему выбирать действие, которое он должен совершить. Он поддерживает список табу,

то есть список действий, которые он уже совершил или вовсе не может совершить.

Настоящий муравей во время перемещения по пути будет оставлять за собой фермент. В алгоритме муравья агент оставляет фермент, помечая действия, которые уже совершил.

После создания популяция искусственных муравьев поровну распределяет между собой действия. Необходимо равное разделение муравьев между действиями, чтобы каждое действие имело возможность стать первым.

Нам необходимо доказать или опровергнуть выводимость формулы, то есть предъявить набор значений переменных \bar{x} не равных между собой, при котором формула выводима, или доказать отсутствие такого набора. В связи с этим мы будем применять муравьиную тактику, назначая приоритет выбора действий, следующим образом:

На начальном этапе количество феромона отождествления формул $P_t(t_1, \dots, t_m)$ и $\neg P(a_1, \dots, a_m)$ при любом наборе констант a_1, \dots, a_m равно 1. Если отождествление проходит успешно и происходит присвоение значений переменным, то количество феромона для такого отождествления возрастает на 1. Если решаемая при отождествлении система уравнений не имеет решений, то количество феромона убывает на 1. Если отождествление приводит к выведению тупикового набора, то количество феромона убывает на 1.

Так как мы имеем i штук дизъюнктов в (1) и формулы $P_{ki}(x_1, \dots, x_{n_i})$ не повторяются, то муравьев тоже i штук. Каждый муравей начинает свой итерационный цикл со своего дизъюнкта, и действует согласно алгоритму ИМА поиска вывода, приведенному выше. Но при этом при каждом присвоении переменным значений муравьи связываются друг с другом, и сравнивают результаты.

Сравнение результатов между различными муравьями состоит в проверке этих результатов на непротиворечивость друг другу. Если муравьи одновременно присваивают одним и тем же переменным разные значения, то такие результаты считаются противоречивыми. Если результаты действий двух муравьев не противоречат друг другу, то присвоения полученных значений переменных осуществляется в дизъюнктах обоих муравьев.

Одинаковое присвоение значений переменным разными муравьями допустимо. Если один из муравьев не может присвоить ни одного нового значения (на всех замыканиях количество феромона равно 0), этот муравей перестает участвовать в итерациях. Алгоритм заканчивает работу, если доказана выводимость или на всех замыканиях количество феромона равно 0.

Литература

1. *Dorigo M., Birattari M., Stutzle T.* Ant Colony Optimization. Artificial Ants as a Computational Intelligence Technique. // IRIDIA – Technical Report Series, 2006, № 23, pp 1–2.

2. *Kosovskaya T.* Discrete Artificial Intelligence Problems and Number of Steps of their Solution // International Journal on Information Theories and Applications, Vol. 18, Number 1, 2011. P. 93–99.
 3. *Kosovskaya T., Petukhova N.* The Inverse Method for Solving Artificial Intelligence Problems in the Frameworks of Logic-Objective Approach and Bounds of its Number of Steps // International Journal «Information Models and Analyses», Vol. 1. 2012. P. 84–93
 4. *Оревков В.П.*, Обратный метод поиска вывода // В кн.: Адаменко А.Н., Кучуков А. М., Логическое программирование и Visual Prolog — Санкт-Петербург, БХВ, 2003, с. 952–965
-

СХЕМА УНИВЕРСАЛЬНОЙ ФУНКЦИИ ДЛЯ ЯЗЫКА РЕФАЛ-5

Гошев В.А.

аспирант кафедры информатики СПбГУ, sunx@sunx.cc

Аннотация: В данной статье предлагается схема реализации универсальной функции для языка программирования РЕФАЛ-5.

Введение

Универсальная функция, по аналогии с универсальной машиной Тьюринга — функция, которая может заменить собой любую функцию. Получив на вход код функции и входные данные (аргументы функции), она вычисляет ответ, который вычислила бы по входным данным функция, чья программа была дана на вход. Такая универсальная функция свойственна интерпретируемым языкам, таким как PHP, Perl, JavaScript и другие, и как правило называется *eval* (от *англ.* evaluate — выполнять). Язык РЕФАЛ-5[1] не является интерпретируемым языком в чистом виде, так как код программы компилируется в код РЕФАЛ-машины и потом выполняется этой машиной, но это не значит, что для него невозможно реализовать такую функцию.

Универсальная функция позволит выполнять динамически генерируемый код, в частности — введенный пользователем, так же универсальная функция позволяет реализовать динамическую загрузку РЕФАЛ-модулей. Кроме своего прямого предназначения — исполнения пользовательского кода, данная функция позволит реализовать такие конструкции, как замыкание, функции высших порядков и анонимные функции, которые, к сожалению не реализованы в наиболее распространенных диалектах языка РЕФАЛ: РЕФАЛ-5 и РЕФАЛ+.

Замыкание — это подпрограмма (процедура или функция), которая использует переменные объявленные вне этой функции, не являющиеся глобальными и не являющихся ее аргументами. Как правило — это функция, которая определена внутри другой функции и использует ее локальные переменные. Стандартными средствами языка РЕФАЛ-5 легко реализовать такую конструкцию невозможно.

Функция высшего порядка — функция, которая принимает в качестве аргумента другую функцию или же возвращает функцию. Этот механизм был реализован в диалекте языка РЕФАЛ, который его авторы назвали РЕФАЛ-7[2]; так же частично данный функционал можно реализовать с помощью Метафункции *Mu*[3].

Анонимная функция, так же иногда называемая лямбда функцией — функция, которая объявляется в месте использования и не имеет идентификатора, по которому может быть вызвана традиционным способом. Та-

кая функция либо используется на месте либо ссылка на нее присваивается переменной и в дальнейшем вызов этой функции происходит через эту переменную. Создание такой конструкции стандартными средствами языка РЕФАЛ-5 так же сложно.

Реализация

Выбор способа реализации

Есть 2 способа реализовать универсальную функцию: как встроенную функцию РЕФАЛ-машины или же средствами самого языка.

Реализация как встроенная функция РЕФАЛ-машины имеет ряд плюсов:

1. Скорость выполнения — функции выполняемые универсальной функцией будут выполняться почти с такой же скоростью, как если бы это были обычные функции, плюс возможно кешировать такие функции, то есть при повторном вызове не компилировать код заново, а выполнять уже скомпилированный.
2. Простота реализации — РЕФАЛ-машине нужно будет лишь запустить компилятор для переданного кода и поместить функцию в область видимости с каким-нибудь уникальным именем.

Однако есть и минусы:

1. Возможно возникновение проблем с лицензией при изменении оригинального компилятора и РЕФАЛ-машины для языка РЕФАЛ-5.
2. После изменения получится не язык РЕФАЛ-5, а язык РЕФАЛ-5 с дополнениями, который не будет полностью совместим с остальными РЕФАЛ-5 машинами, как следствие для выполнения программ, написанных с использованием дополнений, будет необходимо использовать измененную РЕФАЛ-машину.

Если реализовать универсальную функцию средствами языка РЕФАЛ-5, то плюсом будет то, что программа, использующая данную функцию будет работать на всех РЕФАЛ-машинах языка РЕФАЛ-5. Минусы, по сравнению с реализацией функции внутри РЕФАЛ-машины, следующие:

1. Низкая скорость выполнения — универсальная функция будет вынуждена самостоятельно разбирать переданный код и выполнять его, таким образом будет необходимо реализовать интерпретатор РЕФАЛа.
2. Сложность реализации — будет необходимо реализовать разбор переданного кода и его выполнения, то есть полностью реализовать интерпретатор языка РЕФАЛ-5 на РЕФАЛе, что является трудоемкой задачей.

Не смотря на минусы, был решено реализовать универсальную функцию средствами языка РЕФАЛ-5, так как такой способ позволяет использовать стандартную РЕФАЛ-5 машину.

Разбор кода функции

Для упрощения разбора переданного универсальной функции кода будем считать, что этот код написан на базисном РЕФАЛЕ, то есть в коде отсутствуют условия и вложенные блоки. При необходимости данный функционал можно будет добавить позже. С учетом описанных ограничений переданный код будет выглядеть следующим образом:

```
{
    <Образец_1> = <Действия_1>;
    ...
    <Образец_n> = <Действия_n>;
}
```

Таким образом:

- Весь код заключен в фигурные скобки.
- Каждое предложение содержит образец и действия, отделенные от образца символом « \Rightarrow », все предложение заканчивается символом « \Leftarrow ». Однако стоит учесть, что как образец, так и действия, могут содержать литералы, которые в себе содержат эти символы (« \Rightarrow » и « \Leftarrow »).

Для упрощения дальнейшего разбора кода, переданный код преобразуется в поток лексем. Полученный поток лексем разбивается на отдельные предложения (по лексеме « \Leftarrow »), после чего каждое предложение по лексеме « \Rightarrow » разбивается на образец и действия

Выполнение полученного кода

После того, как переданный код разобран необходимо его выполнить. Для этого универсальная функция просматривает образец предложения, в котором могут быть литералы, РЕФАЛ-слова, структурные скобки и переменные. Значения переменных необходимо будет где-то хранить, для этого можно использовать стек, так как программа на языке РЕФАЛ-5 является однопоточной, то можно быть уверенным, что во время выполнения универсальной функции никто не изменит его содержимое. В зависимости от встреченного в образце элемента действия функции могут быть следующими:

- 1 Встретился литерал или РЕФАЛ-слово — проверяется, совпадает ли он с литералом (РЕФАЛ-словом) в переданном аргументе, если нет, то просмотр образца прерывается.

- 2 Встретилась переменная — здесь действия зависят от того, связана ли переменная и от ее типа:
 - 2.1) Переменная связана — проверить, если значение переменной совпадает с начальными символами аргумента, то продолжать дальше, если нет, то прервать просмотр.
 - 2.2) Переменная не связана — для t и s переменной записать в нее следующий терм или символ соответственно, для e переменной необходимо начать внутренний цикл, который поочередно помещает в эту переменную $0, 1, 2, \dots$ и так далее символов, пока весь просмотр образца не завершится успехом или не будут исчерпаны все варианты.
- 3 Встретились структурные скобки — проверить наличие структурных скобок в аргументе, при их наличии внутри скобок произвести просмотр и сопоставление образца, при отсутствии — прервать просмотр.
- 4 Обнаружен конец образца — если при этом в аргументе остались символы, то сопоставление завершено не удачно, если не осталось, то сопоставление завершилось удачей.

Если просмотр по вышеописанным правилам заканчивается удачей, то выполняются соответствующие ему действия, если же нет, то функция переходит к следующему образцу, если все образцы просмотрены, то функция завершается неудачей и вызывает ошибку.

При выполнении действий, соответствующих совпавшему образцу для вызова функций используется метафункция Mu .

Заключение

В статье показано как средствами РЕФАЛ-5 реализовать универсальную функцию, способную выполнить переданный ей код на базисном РЕФАЛЕ.

Литература

1. *Бабаев И.О., Герасимов М.А., Косовский Н.К.* Интеллектуальное программирование. Турбо Пролог и Рефал-5 на персональных компьютерах. — СПб.: Издательство СПбУ, 1992.
2. *С.Ю. Скоробогатов, А.М. Чеповский* — Язык Refal с функциями высшего порядка. <http://iu9.bmstu.ru/science/refal.pdf> [дата просмотра: 02.04.2013].
3. *Турчин В.Ф.* — РЕФАЛ-5. Руководство по программированию и справочник. http://refal.net/rf5_frm.htm [дата просмотра: 02.04.2013]

NP-ПОЛНАЯ ЗАДАЧА О РАЗБИЕНИИ МНОЖЕСТВА НА K ПОДМНОЖЕСТВ И СУБЪЭКСПОНЕНЦИАЛЬНЫЕ ФУНКЦИИ

М.А. Герасимов

(Санкт-Петербург)

Аннотация: Рассматривается возможность определения за полиномиальное время (для детерминированной одноленточной машины Тьюринга) разрешимости задачи о разбиении на K равновесных подмножеств для заданного набора данных, где величина K больше 1. Доказано, что возможен алгоритм, который при определенных ограничениях на входной набор данных решает эту задачу за полиномиальное время.

Детерминированная машина Тьюринга

Для анализа алгоритма рассматривается одноленточная, одноголовочная машина Тьюринга со входной и выходной лентой. Предполагается, что входные данные записаны на входной ленте, обрабатываются на рабочей ленте и результат записывается на выходной ленте. При работе машины Тьюринга используется алфавит, состоящий из четырех символов $\{\#, b, 0, 1\}$. Результатом работы алгоритма считается битовая последовательность, кодирующая исходные данные и соответствующее дерево кодирования, позволяющее однозначно восстановить исходную последовательность. В дальнейшем будем считать, что входные данные (натуральные числа) записаны в виде битовой последовательности на входной ленте между двумя маркерами « $\#$ ». В качестве разделителей используется пустой символ « b ». Считывание второго маркера означает конец цепочки входных данных. Входная лента позволяет считывать входные данные произвольное количество раз. Выходная лента позволяет только записывать результат вычисления в виде последовательности символов рабочего алфавита. Каждый символ выходной цепочки записывается только один раз и больше не изменяется.

Обозначения

Далее будет использоваться линейный список входных элементов, пронумерованных натуральными числами $X = \{x_1, \dots, x_M\}$ и имеющих положительные веса $\{w_1, \dots, w_M\}$ соответственно. Обозначим через $W(X)$ суммарный вес элементов множества X :

$$W(X) = \sum_{i=1}^M w_i$$

Пусть множество X разбито на K дизъюнктивных подмножеств, $K \geq 2$, $X_1, X_2, \dots, X_K : X_i \cap X_j = \emptyset, X_1 \cup X_2 \dots \cup X_K = X$. Согласно [1] весом разбиения назовем величину

$$F(X_1, X_2, \dots, X_K) = \max\{W(X_1), W(X_2), \dots, W(X_K)\}$$

Требуется найти такое разбиение множества X на K подмножеств $X_1^*, X_2^*, \dots, X_K^*$, что

$$F(X_1^*, X_2^*, \dots, X_K^*) = \min\{F(X_1^*, X_2^*, \dots, X_K^*)\}$$

Нахождение разбиения $\{X_1^*, X_2^*, \dots, X_K^*\}$ множества X и точного значения $F(X_1^*, X_2^*, \dots, X_K^*)$ для заданного множества элементов X является NP — полной задачей [2]. Одним из направлений исследования возможностей приближенного решения этой задачи за полиномиальное время на детерминированной машине Тьюринга является поиск решения $F(X_1, X_2, \dots, X_K)$, достаточно близкого к $F(X_1^*, X_2^*, \dots, X_K^*)$ за полиномиальное время.

Субэкспоненциальные функции и их связь с классической задачей о разбиении множества.

Рассмотрим некоторый класс функций, задаваемый с помощью операций сложения, умножения и рекурсии. В качестве базиса можно рассматривать множество $\{0, +, *\}$, в качестве образующих операций выбираем операции подстановки и рекурсии.

Самые простые функции класса определяются следующим образом. Первая субэкспоненциальная функция:

$$f(0) = a$$

$$f(i+1) = f(i) + b,$$

где a, b — некоторые неотрицательные целые числа. Очевидно, при $b=0$ так заданная функция является константой: $f(i) = a$, для любого $i \geq 0$. При $b > 0$ эта функция порождает арифметическую прогрессию с основанием b .

Вторая субэкспоненциальная функция так же известна:

$$f(0) = a_0$$

$$f(1) = a_1$$

$$f(i+1) = f(i) + f(i-1) + b,$$

где $i \geq 1$, i — целое число, a_0 , a_1 и b — неотрицательные целые числа. При $a_0=0$, $a_1=1$, $b=0$, эта последовательность представляет собой числа Фибоначчи. Существует аналитическая формула для представления i -го числа Фибоначчи:

$$\varphi(i) = \left[\frac{1}{\sqrt{5}} \phi^i \right],$$

где $\phi = (1 + \sqrt{5})/2$ — «золотое сечение». При $b=0$ получаем числа Лукаса. При a_0 , a_1 и b отличных от этих значений, получаются функции, свойства которых очень похожи на свойства чисел Фибоначчи.

Определение. Субэкспоненциальная функция индекса k имеет вид:

$$f(0) = a_0$$

...

$$f(k) = a_k$$

$$f(i+1) = f(i) + f(i-1) + \dots + f(i-k) + b,$$

где $i \geq k$, $k \geq 1$. Будем считать, что первая субэкспоненциальная функция имеет индекс 0. Свойства любой последовательности, задаваемой этой рекурсией очень похожи на свойства чисел Фибоначчи, хотя имеют свои особенности. Если $b=0$, будем говорить, согласно [11], что рассматриваемая субэкспоненциальная функция однородна.

Теорема 5. Для любой субэкспоненциальной функции f существует такое $N \geq 0$, что $\forall i \geq N$ справедливо неравенство:

$$f(i) \leq 2^i.$$

Замечание. Легко видеть, что так представленные функции задаются линейной рекурсией порядка k с постоянными коэффициентами (см. [11]). Как показано в этой работе, подобное задание допускает явное выражение субэкспоненциальных функций без применения рекурсии. Так, например для второй субэкспоненциальной функции решение может быть явно записано в виде:

$$f(n) = c \left(\frac{1 + \sqrt{5}}{2} \right)^n + d \left(\frac{1 - \sqrt{5}}{2} \right)^n - b,$$

$$c = \frac{\sqrt{5}-1}{\sqrt{5}} a_0 + \frac{1}{\sqrt{5}} a_1 + \frac{3\sqrt{5}-1}{2\sqrt{5}} b,$$

$$d = \frac{1}{\sqrt{5}} a_0 + \frac{1}{\sqrt{5}} a_1 + \frac{1 - \sqrt{5}}{2\sqrt{5}} b,$$

$$n \geq 0$$

Аналогично может быть найдено явное выражение любой субэкспоненциальной функции индекса k при заданных константах $\{a_i\}_{i=1}^k$, и $b \geq 0$.

Определение. Будем говорить, что множество $X = \{x_1, \dots, x_M\}$ порождается субэкспоненциальной функцией f , если существуют такие целые неотрицательные p и q , что $p \leq q$, $M = q - p + 1$, для всех i от 0 до $M - 1$, $x_i = f(p + i)$.

Определение. Будем говорить, что $X = \{x_1, \dots, x_M\}$ имеет кратность k , если M делится на k .

Теорема 1. Для любой однородной субэкспоненциальной функции $f(b=0)$ с индексом $k \geq 1$, порождающей входной поток данных $X = \{x_1, \dots, x_M\}$ кратности $k + 1$, существует алгоритм, находящий разбиение множества X на два дизъюнктивных подмножества X_1 и X_2 равного веса на детерминированной одноленточной машине Тьюринга с квадратичным от длины входных данных временем.

Теорема 2. Для любой субэкспоненциальной функции f с индексом $k \geq 1$, порождающей входной поток данных $X = \{x_1, \dots, x_M\}$ кратности $2(k + 1)$, существует алгоритм, находящий разбиение множества X на два дизъюнктивных подмножества X_1 и X_2 равного веса на детерминированной одноленточной машине Тьюринга с квадратичным от длины входных данных временем.

Теорема 3. Если поток данных $X = \{x_1, \dots, x_M\}$ порождается некоторой функцией f , для которой справедливо неравенство:

$$f(i) \geq 2^i, i \geq 0,$$

то точное разбиение множества X на два или более дизъюнктивных подмножества невозможно.

Литература

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. Fischetti M., Martello S. Worst-case analysis of the differencing method for the partition problem. // Math. Programming. 1987. V 37, N 1 pp 117–120.
3. Минский М. Вычисления и автоматы. — М., 1971.
4. Huffman D. A method for construction of minimum redundancy codes. Proceeding of IRE, 40(9):1098–1101, 1952.
5. Horowitz E., Sahni S. Fundamentals of Computer Algorithms. Computer Science Press, 1978.
6. Shannon C.E. A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, July 1948, pp. 373–423.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕРВЕРОВ ПРИЛОЖЕНИЙ

Шульгин Д.С.

аспирант кафедры информатики СПбГУ, dan.shulgin@gmail.com

Аннотация: Сервер приложений является одной из ключевых составляющих в многозвенной архитектуре приложений. Он обеспечивают масштабируемую, высокопроизводительную инфраструктуру, предоставляет модель разработки и развертывания, а также различные программные компоненты, значительно облегчающие процесс разработки. Говоря о Java Enterprise, существует несколько вендоров, предоставляющих свои продукты с поддержкой Java EE 6. Каждый из них был по-разному спроектирован и разработан, имеет разную стратегию развития и поддерживает разные компоненты спецификации. Это позволяет проводить сравнение между серверами приложений, оценивать эффективность, надежность и скорость их работы.

В данной статье будут рассмотрены наиболее распространенные сервера приложений и проведено сравнение их характеристик, общее сравнение с целью в дальнейшем выделить критерии их работы с компонентами бизнес-логики приложений.

Введение

Вне контекста приложений, которые могут быть развернуты на серверах, сравнить их можно по некоторым общим критериям, которые будут рассмотрены в данной обзорной статье. Если сравнивать сервера, принимая во внимание зависимость от развернутого на них приложения, то основным составляющим будут бизнес-объекты, входящие в структуру приложения. В составной критерий в таком случае войдет тип объекта, его размер, количество полей и бизнес-методов, а также способы управления сервера жизненным циклом этого объекта. В данном случае речь идет об объектах Enterprise JavaBeans, входящих в состав Java EE, для работы с которыми сервер приложений должен иметь в своем составе контейнер EJB.

Рассмотрим наиболее распространенные сервера приложений и проведем их общее сравнение с целью в дальнейшем выделить критерии сравнения их работы с компонентами бизнес-логики приложений. В Таблице 1 перечислены сервера приложений, которые будут рассмотрены, и их поддержка Java EE — полная или веб-профиль.

Oracle Glassfish Server

Когда Oracle овладел всеми продуктами Sun Microsystems, активнее стала использоваться стратегия разделения на бесплатное и платное решение. Также произошло и с Glassfish [2]: Glassfish Server Open Source Edition &

Application Server	Version	Java EE Compatibility
Oracle Glassfish	3.1.2.2	Full
Oracle WebLogic	12c(12.1.1)	Full
JBoss AS	7.1.1	Web Profile
IBM WebSphere	3.0.0.3	Full
Apache TomEE	1.5.1	Web Profile

Таблица 1: JavaEE Comrability [1]

Oracle Glassfish Server. Open Source версия позиционируется как лучший бесплатный сервер приложений и полностью поддерживает Java EE 6. Коммерческая версия расширяет функционал бесплатной версии поддержкой Oracle и некоторыми интересными функциями, такими как Active Redeploy — возможность взаимодействовать с приложением во время его переразвертывания.

Примечательно, что в состав Glassfish входит Apache Tomcat в качестве контейнера сервлетов.

Говоря о продуктах Oracle, стоит упомянуть и Oracle WebLogic Server [3], зародившийся еще в 1997-м году. Последняя версия 12 также обеспечивает полную поддержку Java EE 6.

Представители Oracle утверждают, что оба этих сервера приложений важны, т.к. их предназначение несколько разнится: Glassfish больше рассчитан на легковесные приложения, в то время как WebLogic лучше подходит для приложений с большим количеством транзакций и работы с базами данных Oracle.

JBoss AS

Продукт компании Red Hat. Основанный на Java, JBoss [4] имеет лишь вэб-профиль лицензию JavaEE6: например, нет поддержки JNDI (Java Naming and Directory Interface). Однако, поддержки этой и других технологий нет в Open source версии JBoss AS. Покупая лицензию Red Hat, можно получить поддержку необходимой технологии поверх JBoss AS. JBoss Enterprise Application Platform — это платформа, с настройкой производительности под высоко нагруженные, масштабируемые приложения. Помимо поддержки Java EE, JBoss EAP предоставляет и различные инструменты для управления. Например, JBoss Operating Networks, позволяющий администрирования все окружения для тестирования и разработки.

IBM WebSphere

Флагман бренда WebSphere [5] компании IBM. Также разделен на коммерческую версию и с открытым исходным кодом. Обе версии обеспечивают полную поддержку Java EE 6. Основан на Apache Geronimo, в состав

которого входит Apache Tomcat в качестве контейнера сервлетов. Geronimo, к слову сказать, также полностью поддерживает Java EE 6.

В качестве еще одного плюса в пользу этого сервера приложений — русскоязычная документация с примерами для актуальной версии самого сервера и используемых технологий.

Apache TomEE

Осенью 2011го увидел свет новый продукт от Apache — TomEE [6]. JavaEE Web Profile on Tomcat. Поддержка EJB в нем осуществляется за счет Open EJB — контейнера EJB с открытым исходным кодом по лицензией Apache 2.0. Ранее Open EJB уже интегрировался в другие сервера приложений, например, тот же Geronimo. Open EJB в текущей версии 3.1.4 полностью поддерживает EJB 3.0 и частично 3.1. В октябре 2011 TomEE получил сертификат Oracle как совместимая реализация Java EE 6 (вэб-профиль).

TomEE выступает как надстройка на Tomcat. Стоит сказать, что решение, наподобие TomEE уже существовали. Например, TCat, также основанный на чистом Tomcat и осуществляющий поддержку EJB за счет Open EJB. Однако, это платный продукт, в то время как Apache предоставляет бесплатное решение.

Сравнение

Переходя к сравнению серверов, стоит сказать, что на данный момент существует уже немало работ, однако, зачастую они направлены на выделение плюсов того или иного сервера. В связи с этим было принято решение произвести собственные независимые исследования, а затем сравнить их с существующими. Выделим критерии, по которым можно сравнить сервера приложений без развертывания на них самих приложений: объем дистрибутива, объем занимаемого места на диске, время старта, объем занимаемой процессом *java.exe* памяти процессора.

Для измерений была взята машина со следующими характеристиками:

- ОС Windows 7 Professional, SP1, 64bit, build 6.1.7601 (Win7 RTM)
- Процессор Intel(R) Core(TM) i5-2430M, 2.40GHz
- 4GB оперативной памяти

Помимо общих следует описать и аппаратные характеристики, также влияющие на результаты измерений. Они были установлены с помощью программы диагностики и тестирования *AIDA64 Extreme Edition v2.2.1800*:

- IDE Controller Intel(R) Mobile Express Chipset SATA AHCI Controller
- Disk Drive TOSHIBA MK6476GSX (640 GB, 5400 RPM, SATA-II)
- Disk File System NTFS
- Memory Type DDR3 SDRAM
- Memory Speed DDR3-1333 (667 MHz)

- DRAM Manufacturer Samsung
- Serial Number 61E2CAC5h (3318407777)

Проведенные тесты памяти на скорость чтения и записи показали следующие результаты:

- Memory Read Speed 9817 MB/s
- Memory Write Speed 10307 MB/s
- Memory Copy Speed 9141 MB/s

Размер дистрибутива

Сервер приложений	Размер (mb)
Oracle Glassfish	54.6(32.7*)
JBoss AS	127
IBM WebShepre	106
Apache TomEE	27.1

Таблица 2. Размер дистрибутива. * — Web Profile

Объем занимаемого дискового пространства

Сервер приложений	Размер (mb)
Oracle Glassfish	105(60*)
JBoss AS	78
IBM WebShepre	2200
Apache TomEE	31

Таблица 3. Объем занимаемого дискового пространства после установки.

* — Web Profile

Время развертывания

Сервер приложений	Время старта (мс)
Oracle Glassfish	5756
JBoss AS	2218
IBM WebShepre	>40 000
Apache TomEE	734

Таблица 4. Время старта сервера

Память, используемая процессом java.exe

Сервер приложений	Время старта (мс)
Oracle Glassfish	5756
JBoss AS	2218
IBM WebShepre	>40 000
Apache TomEE	734

Таблица 5. Память, используемая процессом java.exe

Дальнейшее сравнение невозможно без попытки развернуть на сервере какое-либо приложение. Т.к. исследование проводится в контексте технологии Enterprise JavaBeans, то в рамках сравнения на каждом из серверов приложений были развернуты тестовые приложения, представляющие собой имплементации типов EJB 3.1: Singleton, Stateless и Statefull бины. Эти примеры предоставляет официальный сайт Apache TomEE[6].

Сервер приложений	Singleton (мс)	Stateless (мс)	Statefull (мс)
Oracle Glassfish	970	499	216
JBoss AS	290	281	120
Apache TomEE	234	196	115
IBM WebShepre	3.0.0.3	3.0.0.3	Full
Apache TomEE	1.5.1	1.5.1	Web Profile

Таблица 6. Время развертывания приложений (мс)

Заключение

Подводя итог сравнительному анализу, можно выделить Oracle Glassfish Server: два дистрибутива для full support и web profile, сравнительно быстрый старт и потребление ресурсов, поддержка Oracle. Также все большую популярность набирает продукт от Apache (TomEE и OpenEJB), во многом за счет открытого исходного кода и быстродействия. Безусловно, стоит отметить и тот факт, что IBM WebSphere — это единственный из представленных полностью платный сервер приложений, чтобы пользоваться им полноценно в любом случае придется приобрести лицензию.

На данном этапе выбранные критерии для сравнения серверов приложений соответствуют интуитивным представлениям об эффективности их работы с точки зрения управления бизнес-объектами. Сравнительный анализ стоит расширить результатами нагрузочного тестирования приведенных тестовых приложений под управлением рассматриваемых серверов в зависимости от их конфигурации. Проведение исследований по сравнению конфигураций серверов приложений, выделение общего множества, влияющего на быстродействие того или иного сервера, а также введение критерия оптимальности стратегии конфигурации является целью моей дальнейшей работы.

Литература

1. Поддержка JAVA EE серверами приложений JavaEE Compability [Электронный ресурс] // <http://java.sun.com/javaee/overview/compatibility.jsp> [дата просмотра: 02.04.2013]
2. Oracle Glassfish Server [Электронный ресурс] // <http://glassfish.java.net/> [дата просмотра: 02.04.2013]

3. Oracle Weblogic Server [Электронный ресурс] // <http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html> [дата просмотра: 02.04.2013]
 4. JBoss AS [Электронный ресурс] // <http://www.jboss.org/jbossas> [дата просмотра: 02.04.2012]
 5. IBM Websphere [Электронный ресурс] // <http://www-01.ibm.com/software/ru/websphere/> [дата просмотра: 02.04.2013]
 6. Apache TomEE [Электронный ресурс] // <http://openejb.apache.org/> [дата просмотра: 02.04.2013]
 7. Application Servers Benchmark [Электронный ресурс] // <http://agoncal.wordpress.com/2011/10/20/o-java-ee-6-application-servers-where-art-thou/> [дата просмотра: 02.04.2013]
 8. Survey Results: Java EE Containers [Электронный ресурс] // <http://zereturnaround.com/blog/java-ee-container-heaven-hell-survey-results/> [дата просмотра: 02.04.2013]
 9. Оценивая серверы приложений J2EE // Interface Ltd. — 2001
 10. *Буди Карниван*. Программирование web-приложений на языке Java // Лори. — 2009. — 880с.
 11. *Х.М. Дейтел, П. Дж. Дейтел, С.И. Сантри*. Технологии программирования на Java 2. Распределенные приложения // Бином-пресс. — 2009. — 466с.
-

КЛАССИФИКАЦИЯ ДРЕВНЕТИБЕТСКИХ ТЕКСТОВ С ПОМОЩЬЮ МЕТОДОВ СПЕКТРАЛЬНОГО АНАЛИЗА

М.А. Герасимов, А.А. Носовец

*Санкт-Петербург, каф. Информатики СПбГУ
ge@star.math.spbu.ru, nosovecc@gmail.com*

Аннотация: Работа посвящена анализу древнетибетских текстов. Используется аппарат статистического анализа множества исходных текстов на древнетибетском языке, записанных в кодировке UTF-8. Делается попытка использования технологий анализа больших объемов данных для извлечения смысла (семантики) из исходных текстов.

Данная работа направлена на классификацию древнетибетских текстов методом спектрального анализа данных.

О методах анализа текстов

Исследование языка в первую очередь вызвано интересом к естественному интеллекту человека, поэтому такая форма существования языка как письменный текст, также является объектом пристального внимания. Сегодня наблюдается устойчивый интерес к работам, проводимым в этой области.

Задачи по автоматической обработке текста, включают в себя задачи информационного поиска, связанные с нахождением информации в электронных базах знаний, автоматическим извлечением данных и фактов по заданной тематике, автоматическим реферированием электронных документов. Имеются и другие проблемы обработки естественного языка с использованием вычислительной техники: понимание естественного языка, лингвистическое обеспечение информационных систем разных типов и машинный перевод.

При автоматической обработке текстов среди других задач возникают две близкие задачи: классификации и кластеризации текстов. Первая состоит в том, чтобы отнести текст к определенной категории. Вторая — разбить набор текстов на группы, близкие в некотором смысле. При кластеризации текстов происходит попытка выяснить естественное выделение в группах текстов определенных групп с использованием соответствующих алгоритмов и визуализации. Для каждого текста в любом случае строится набор признаков.

Задача автоматической классификации и кластеризации текстов имеет практическое значение. Соответствующие процедуры применяются при обработке информационных потоков.

Тесно связана с этим задача атрибуции текстов. Атрибуция (*от лат. attributio — приписывание*) — определение атрибутов. Существуют мето-

ды, позволяющие проводить атрибуцию текста. Это — отнесение текста к определенному жанру, стилю, времени написания и т. п.

Большинство современных методов определения жанра, стиля, времени написания, так или иначе, основаны на анализе лингвостилистических (числовых) характеристик, вычисляемых по тексту.

Можно выделить следующие характеристики, которые могут быть получены для письменного текста и доступны для последующего анализа:

1. атрибуты, легко вычисляемые по тексту: длина предложений и слов, средняя длина предложений и слов;
2. однородность текста (распределение по тексту составляющих единиц текста);
3. грамматические конструкции языка;
4. морфологические конструкции языка;
5. синтаксические конструкции языка;
6. лексика (богатство лексики, частотные словари, наличие определенных слов);
7. переходы между составляющими единицами текста;
8. анализ дополнительных признаков текста (сокращений, пунктуации, «смайликов», ошибок).

Полученные лингвостилистические характеристики подвергаются анализу с использованием различных математических методов, среди которых можно выделить следующие:

1. статистические методы;
2. изучение переходов между составляющими единицами текста;
3. арифметические методы;
4. методы распознавания образов и искусственного интеллекта.

Анализ больших объемов данных

Развитие методов записи и хранения данных привело к бурному росту объемов собираемой и анализируемой информации. Для того чтобы провести автоматический анализ данных, используется Data Mining.

Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Суть и цель технологии Data Mining можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей.

Неочевидных — это значит, что найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем.

Объективных — это значит, что обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным.

Практически полезных — это значит, что выводы имеют конкретное значение, которому можно найти практическое применение.

Нередко Data Mining отождествляют с Knowledge Discovery in Databases, хотя более правильно считать Data Mining одним из шагов этого процесса.

Data Mining — это не один, а совокупность большого числа различных методов обнаружения знаний. Все задачи, решаемые методами Data Mining, можно условно разбить на пять классов: классификация, регрессия, кластеризация, ассоциация и последовательные шаблоны.

Data Mining имеет мульти дисциплинарный характер.



Классификация

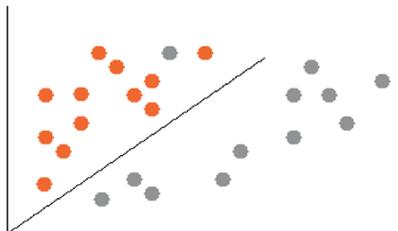
Задача разбиения множества объектов или наблюдений на априорно заданные группы, называемые классами, внутри каждой из которых они предполагаются похожими друг на друга, имеющими примерно одинаковые свойства и признаки. При этом решение получается на основе анализа значений атрибутов (признаков).

Классификация является одной из важнейших задач Data Mining.

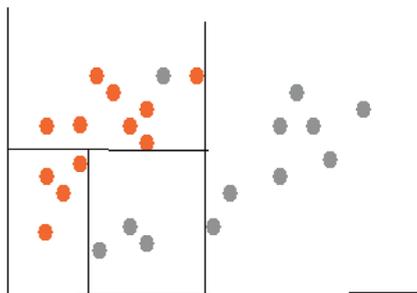
Если число классов ограничено двумя, то имеет место бинарная классификация, к которой могут быть сведены многие более сложные задачи. Например, вместо определения таких степеней кредитного риска, как «Высокий», «Средний» или «Низкий», можно использовать всего две — «Выдать» или «Отказать».

Для классификации в Data Mining используется множество различных моделей: нейронные сети, деревья решений, машины опорных векторов,

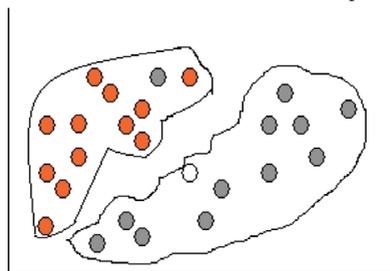
метод k-ближайших соседей, алгоритмы покрытия и др., при построении которых применяется обучение с учителем, когда выходная переменная (метка класса) задана для каждого наблюдения. Формально классификация производится на основе разбиения пространства признаков на области, в пределах каждой из которых многомерные векторы рассматриваются как идентичные. Иными словами, если объект попал в область пространства, ассоциированную с определенным классом, он к нему и относится.



Решение задачи классификации методом линейной регрессии



Решение задачи классификации методом деревьев решений



Решение задачи классификации методом нейронных сетей

Регрессия

В теории вероятностей и математической статистике это зависимость среднего значения случайной величины от некоторой другой величины или даже нескольких. В отличие от чисто функциональной зависимости $y=f(x)$, где каждому значению независимой переменной x соответствует единственное значение зависимой переменной y , регрессионная зависимость предполагает, что каждому значению переменной x могут соответствовать различные значения y , обусловленные случайной природой зависимости. Если

некоторому значению величины x_i соответствует набор значений величин $\{y_{i1}, y_{i2}, \dots, y_{in}\}$, то зависимость средних арифметических:

$$\bar{y}_i = \frac{(y_{i1}, y_{i2}, \dots, y_{in})}{n_i}$$

от x_i и является регрессией в статистическом понимании данного термина.

Изучение регрессии в теории вероятностей основано на том, что случайные величины X и Y , имеющие совместное распределение вероятностей, связаны вероятностной зависимостью: при каждом фиксированном значении $X=x$, величина Y является случайной величиной с определённым (зависящим от значения x) условным распределением вероятностей. Регрессия величины Y по величине X определяется условным математическим ожиданием Y , вычисленным при условии, что $X=x: E(Y|x) = u(x)$. Уравнение $y = u(x)$ называется уравнением регрессии, а соответствующий график — линией регрессии Y по X . Точность, с которой уравнение Y по X отражает изменение Y в среднем при изменении x , измеряется условной дисперсией D величины Y , вычисленной для каждого значения $X=x: D(Y|x) = D(x)$. Если $D(x) = 0$ при всех значениях x , то можно достоверно утверждать, что Y и X связаны строгой функциональной зависимостью $Y = u(X)$. Если $D(x) > 0$ при всех значениях x и $u(x)$ не зависит от x , то говорят, что регрессионная зависимость Y по X отсутствует.

Линии регрессии обладают следующим замечательным свойством: среди всех действительных функций $f(x)$ минимум математического ожидания $E[Y - f(X)]^2$ достигается для функции $f(x) = u(x)$. Это означает, что регрессия Y по X даёт наилучшее в указанном смысле представление величины Y по величине X . Это свойство позволяет использовать регрессию для предсказания величины Y по X . Иными словами, если значение Y непосредственно не наблюдается и эксперимент позволяет регистрировать только X , то в качестве прогнозируемого значения Y можно использовать величину $Y = u(X)$. Наиболее простым является случай, когда регрессионная зависимость Y по X линейна, т. е. $E(Y|x) = b_0 + b_1x$, где b_0 и b_1 — коэффициенты регрессии. На практике обычно коэффициенты регрессии в уравнении $y = u(x)$ неизвестны, и их оценивают по наблюдаемым данным.

Регрессия широко используется в аналитических технологиях при решении различных бизнес-задач, таких как прогнозирование (продаж, курсов валют и акций), оценивания различных бизнес-показателей по наблюдаемым значениям других показателей (скоринг), выявление зависимостей между показателями и т. д.

Кластеризация

Объединение объектов или наблюдений в непересекающиеся группы, называемые кластерами, на основе близости значений их атрибутов (при-

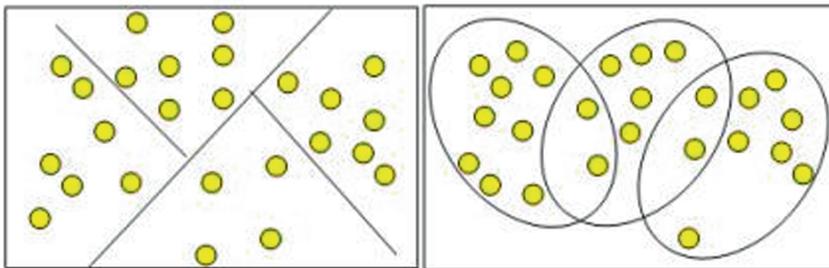
знаков). В результате в каждом кластере будут находиться объекты, похожие по своим свойствам друг на друга и отличающиеся от тех, которые расположены в других кластерах. При этом, чем больше подобие объектов внутри кластера и чем сильнее их непохожесть на объекты в других кластерах, тем лучше кластеризация.

Формальная постановка задачи кластеризации выглядит следующим образом. Пусть заданы множества объектов $X=(x_1, x_2, \dots, x_n)$ и номеров (имён, меток) кластеров $Y=(y_1, y_2, \dots, y_k)$. Для X определена некоторая функция расстояния между объектами $D(x, x')$, например, метрика $L2$. Кроме этого, имеется конечная выборка обучающих примеров $X_m=(x_1, x_2, \dots, x_m)$ из множества X , которую требуется разбить на X_m на непересекающиеся подмножества (кластеры) так, чтобы каждое из них состояло бы только из элементов, близких по метрике D . При этом каждому объекту x_i из множества X_m присваивается номер кластера y_j .

Тогда задача будет заключаться в поиске функции f , которая любому объекту x из множества X ставит в соответствие номер кластера y из множества Y , которое само по себе бывает известно заранее. Однако в большинстве случаев приходится определять оптимальное число кластеров исходя из особенностей решаемой задачи.

Кластеризация позволяет добиться следующих целей:

- Улучшает понимание данных за счет выявления структурных групп. Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятия решений, применяя к каждому кластеру свой метод анализа.
- Позволяет компактно хранить данные. Для этого вместо хранения всей выборки можно оставить по одному типичному наблюдению из каждого кластера.
- Обнаружение новых нетипичных объектов, которые не попали ни в один кластер.



Непересекающиеся и пересекающиеся кластеры

На сегодняшний день разработано более сотни различных алгоритмов кластеризации. Краткая характеристика подходов к кластеризации.

Алгоритмы, основанные на разделении данных (Partitioning algorithms), в т. ч. итеративные:

- разделение объектов на k кластеров;
- итеративное перераспределение объектов для улучшения кластеризации.

Иерархические алгоритмы (Hierarchy algorithms):

- агломерация: каждый объект первоначально является кластером, кластеры, соединяясь друг с другом, формируют больший кластер и т. д.

Методы, основанные на концентрации объектов (Density-based methods):

- основаны на возможности соединения объектов;
- игнорируют шумы, нахождение кластеров произвольной формы.

Грид-методы (Grid-based methods):

- квантование объектов в грид-структуры.

Модельные методы (Model-based):

- использование модели для нахождения кластеров, наиболее соответствующих данным.

Ассоциация

Ассоциация — выявление закономерностей между связанными событиями. Примером такой закономерности служит правило, указывающее, что из события X следует событие Y . Такие правила называются ассоциативными. Впервые эта задача была предложена для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis).

Последовательный шаблон

Последовательность вида

$$A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n, t(A_i) \leq t(A_j), t(A_i) \leq t(A_j), i \leq j,$$

где A — событие, или множество событий, t — время. Таким образом, последовательный шаблон это упорядоченная во времени последовательность событий, наблюдаемая настолько часто, что это позволяет говорить о наличии между ними связей. Анализ таких связей позволяет обнаруживать правила, что если в наблюдении появился набор одних событий из шаблона, то это с высокой долей вероятности повлечет за собой появление других событий или события из этого же шаблона. При этом события первой группы называются основными, а события, появление которых ожидается — целевыми. Основным приложением последовательных шаблонов является исследование типичных моделей поведения клиентов.

Теория последовательных шаблонов происходит из теории ассоциативных правил. Методы анализа ассоциативных правил и последовательных шаблонов во многом похожи: и в том, и в другом случае используются такие понятия как предметный набор и транзакция, такие числовые характеристики, как поддержка и достоверность, а для обнаружения частых шаблонов применяются различные модификации алгоритма Apriori. Однако, между ассоциативными правилами и последовательными шаблонами есть принципиальное различие. В ассоциативных правилах представляет интерес факт совместного появления предметов в транзакции и не рассматривается порядок и появления. В последовательных шаблонах, напротив, последовательность событий играет решающую роль, поскольку считается, что предыдущие события влияют на вероятность появления последующих.

Задача поиска последовательных шаблонов была впервые решена Р. Агравалом и Р. Срикнатом, авторами популярного алгоритма поиска ассоциативных правил Apriori. Они предложили 3 алгоритма для решения задачи открытия последовательных шаблонов на больших массивах данных — GSP, AprioriSome и AprioriAll.

Нейронная сеть

Нейронная сеть представляет собой совокупность нейроподобных элементов, определенным образом связанных друг с другом и внешней средой с помощью связей, определяемых весовыми коэффициентами. В процессе функционирования сети осуществляется преобразование входного вектора в выходной, некоторая переработка информации.

Конкретный вид выполняемого сетью преобразования данных обуславливается не только характеристиками нейроподобных элементов, но и особенностями ее архитектуры, а именно топологией межнейронных связей, выбором определенных подмножеств нейроподобных элементов для ввода и вывода информации, способами обучения сети, наличием или отсутствием конкуренции между нейронами, направлением и способами управления и синхронизации передачи информации между нейронами.

Наиболее часто нейронные сети используются для решения следующих задач:

- Классификация образов — указание принадлежности входного образа, представленного вектором признаков, одному или нескольким предварительно определенным классам.
- Кластеризация — классификация образов при отсутствии обучающей выборки с метками классов.
- Прогнозирование — предсказание значения $y(t_n + 1)$ при известной последовательности $y(t_1), y(t_2), \dots, y(t_n)$.
- Оптимизация — нахождение решения, удовлетворяющего системе ограничений и максимизирующим или минимизирующим целевую функцию.

- Память, адресуемая по содержанию (ассоциативная память) — память, доступная при указании заданного содержания.
- Управление — расчет такого входного воздействия на систему, при котором она следует по желаемой траектории.

Метод экспертного статистического анализа

Спектральный анализ текста направлен на отбор текстов по спектру подходящего текста – получение всех текстов, которые тематически связаны с исходным текстом; введение спектрального радиуса – определение спектрального расстояния между текстами.

Спектр документа представляет собой множество. При этом отфильтровываются такие части речи как предлоги, местоимения и другие не смысловые части речи, которые не несут информации о тематике данного текста. Сравнивая спектры двух текстов можно сказать, как сильно пересекаются их спектральные расстояния, что позволяет оценить их тематическую близость.

В данной работе используется метод, относящийся к кластерным методам. Перед тем как рассмотреть этот метод, рассмотрим краткий обзор существующих кластерных методов.

Иерархические агломеративные методы (Agglomerative Nesting, AGNES)

Эта группа методов характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров.

В начале работы алгоритма все объекты являются отдельными кластерами. На первом шаге наиболее похожие объекты объединяются в кластер. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер.

Иерархические дивизимные (делимые) методы (Divisive ANAlysis, DIANA)

Эти методы являются логической противоположностью агломеративным методам. В начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры, в результате образуется последовательность расщепляющих групп.

Расстояния между объектами предполагает их представление в виде точек m -мерного пространства R^m . В этом случае могут быть использованы различные подходы к вычислению расстояний.

Евклидово расстояние определяется по формуле

$$\rho_E(x_i - x_j) = \sqrt{\sum_{l=1}^k (x_{il} - x_{jl})^2},$$

где x_{i1}, x_{j1} — значения 1-го признака у i -го (j -го) объекта ($l=1, 2, \dots, k, i, j=1, 2, \dots, n$).

Расстояние по Хеммингу является просто средним разностей по координатам. Это расстояние вычисляется по формуле

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

Расстояние Чебышева может оказаться полезным, когда два объекта различаются только по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле

$$l_{ij} = \max_{1 \geq k \geq m} |x_{ik} - x_{jk}|$$

Формально данные меры можно получить из более общей формулы П. С. Урысона:

$$l_{ij} = \left(\sum_{1 \geq k \geq m} (|x_{ik} - x_{jk}|)^p \right)^{1/p}$$

при $p=0$ и $p \rightarrow 3$ соответственно.

Кластер имеет следующие математические характеристики: центр, радиус, среднеквадратическое отклонение, размер кластера.

Центр кластера — это среднее геометрическое место точек в пространстве переменных.

Радиус кластера — максимальное расстояние точек от центра кластера.

Как было отмечено в одной из предыдущих лекций, кластеры могут быть перекрывающимися. Такая ситуация возникает, когда обнаруживается перекрытие кластеров. В этом случае невозможно при помощи математических процедур однозначно отнести объект к одному из двух кластеров. Такие объекты называют спорными.

Спорный объект — это объект, который по мере сходства может быть отнесен к нескольким кластерам.

Размер кластера может быть определен либо по радиусу кластера, либо по среднеквадратичному отклонению объектов для этого кластера. Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера. Если это условие выполняется для двух и более кластеров, объект является спорным.

Алгоритм k -means

Вначале выбирается k произвольных исходных центров — точек в пространстве всех объектов. Далее итерационно выполняется операция двух шагов.

На первом шаге все объекты разбиваются на k групп, наиболее близких к одному из центров. Близость определяется расстоянием, которое вычисляется одним из описанных ранее способов.

На втором шаге вычисляются новые центры кластеров.

Рассмотренная операция повторяется рекурсивно до тех пор, пока центры кластеров не перестанут меняться.

Алгоритм BIRCH

(Balanced Iterative Reducing and Clustering using Hierarchies)

В этом алгоритме реализован двухэтапный процесс кластеризации.

В ходе первого этапа формируется предварительный набор кластеров. На втором этапе к выявленным кластерам применяются другие алгоритмы кластеризации — пригодные для работы в оперативной памяти.

Алгоритм WaveCluster

WaveCluster представляет собой алгоритм кластеризации на основе волновых преобразований. В начале работы алгоритма данные обобщаются путем наложения на пространство данных многомерной решетки. На дальнейших шагах алгоритма анализируются не отдельные точки, а обобщенные характеристики точек, попавших в одну ячейку решетки. В результате такого обобщения необходимая информация уместается в оперативной памяти. На последующих шагах для определения кластеров алгоритм применяет волновое преобразование к обобщенным данным.

Используемый метод

В дальнейшем будем называть данный метод — методом экспертного статистического анализа. Его можно описать поэтапно:

- Экспертом формируется начальный кластер (кластеры) в виде произвольного множества объектов, которые, по мнению эксперта, принадлежат кластеру;
- Для кластера строится множество точек, соответствующих выбранным объектам.
- Для данного множества вычисляется «центр» в виде математического ожидания его точек;
- Вычисляется радиус этого кластера как максимальное расстояние от точки множества до «центра».

Если объект, взятый не из кластера, имеет среднеквадратическое отклонение меньше, чем радиус кластера, то данный объект принадлежит кластеру. Если же среднеквадратичное отклонение больше спектрального радиуса, то данный объект относится к другому кластеру или образует свой собственный кластер.

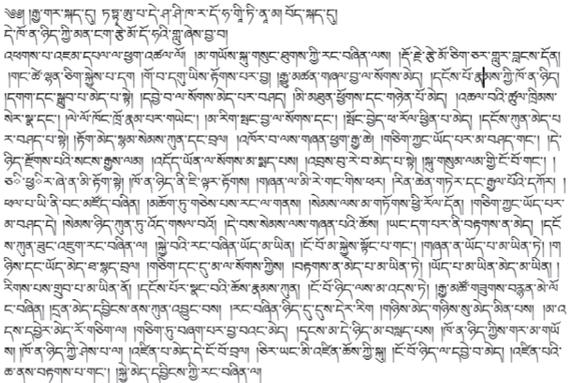
– При исследовании объектов могут образовываться пересекающиеся кластеры. Это означает, что объект (в данной работе текст) относится одновременно к разным темам.

Пример применения метода

Применив данный метод ко всем набору текстов можно получить все тексты, классифицированные по степени тематической близости с исходным текстом.

В рамках данной работы были взяты тексты, тематически схожи (смысловую схожесть текстов определял специалист по тибетским текстам). На основе этих текстов строился спектральный корпус. Затем находился радиус окрестности этого корпуса.

Было отобрано 34 текста одной тематики на древнетибетском языке.



Для дальнейшей работы эти тексты были преобразованы на латиницу.

```
[yig'mgo'mdun'ma yig'mgo'sgab'ma SHAD SHAD de TSHEG kho TSHEG nA TSHEG
nyid TSHEG kyi TSHEG man TSHEG ngag TSHEG tse TSHEG mo TSHEG do TSHEG
ha'i TSHEG glu TSHEG zhes TSHEG bya TSHEG bu TSHEG bzhugs TSHEG so SHAD
SHAD yig'mgo'mdun'ma yig'mgo'sgab'ma SHAD SHAD rgya TSHEG qar TSHEG skad
TSHEG du SHAD ta'i ta'i ta Wa TSHEG U TSHEG pu TSHEG de TSHEG sha TSHEG
shi TSHEG kha TSHEG r'a TSHEG do TSHEG ha'i TSHEG gi TSHEG ti TSHEG na
TSHEG nar SHAD bod TSHEG skad TSHEG du SHAD de TSHEG kho TSHEG nA TSHEG
nyid TSHEG kyi TSHEG man TSHEG ngag TSHEG tse TSHEG mo TSHEG do TSHEG
ha'i TSHEG glu TSHEG zhes TSHEG bya TSHEG bu SHAD 'phags TSHEG pu TSHEG
'jam TSHEG dgal TSHEG l'u TSHEG phyag TSHEG 'tshal TSHEG lo SHAD SHAD mAr
TSHEG g.yos TSHEG sku TSHEG gsung TSHEG thugs TSHEG rang TSHEG
bzhin TSHEG las SHAD SHAD rdo TSHEG rje TSHEG tse TSHEG mo TSHEG cig
TSHEG car TSHEG glur TSHEG blangs TSHEG don SHAD SHAD gang TSHEG tse
TSHEG lhan TSHEG cig TSHEG skyes TSHEG pu TSHEG dag SHAD go TSHEG bu
TSHEG dgu TSHEG yis TSHEG rtogs TSHEG par TSHEG bya SHAD SHAD rgyu TSHEG
mtshan TSHEG gzhal TSHEG bya TSHEG l'u TSHEG sogs TSHEG med SHAD SHAD
dngos TSHEG po TSHEG rnam TSHEG kyi TSHEG kho TSHEG nA TSHEG nyid SHAD
```

При обработке каждого текста был создан каталог (в итоге получили 34 каталога). В каждом каталоге находится 4 файла.

1TableFrequency.txt
101.table.val.txt
101.txt
101.val.txt

```

1 pU 70 0.0207
2 nyid 35 0.0104
3 bU 34 0.0101
4 bya 21 0.0062
5 pa'i 20 0.0060
6 med 19 0.0057
7 sems 19 0.0057
8 rab 18 0.0054
9 gsungs 18 0.0054
10 rnam 16 0.0048
11 bzhin 16 0.0048
12 l'u 15 0.0045
13 yang 15 0.0045
14 thams 15 0.0045
15 cad 15 0.0045
16 brjod 14 0.0042
17 'khor 14 0.0042
18 las 13 0.0039
19 mAr 12 0.0036
20 sogs 12 0.0036
21 bdag 11 0.0033
22 dkyil 11 0.0033
23 ba'i 11 0.0033
24 yis 11 0.0033
25 mtshan 10 0.0030
26 sangs 10 0.0030
27 rgyas 10 0.0030
28 rtdogs 10 0.0030
29 'di 9 0.0027
30 kha 9 0.0027

```

```

1 l'u 2363 0.1888
2 pa'i 2014 0.1609
3 nyid 1572 0.1256
4 med 1154 0.0922
5 mAr 1081 0.0864
6 bya 1077 0.0861
7 mi 349 0.0279
8 'gyur 307 0.0246
9 bzhin 292 0.0235
10 kun 256 0.0205
11 sems 237 0.019
12 thams 202 0.0162
13 cad 202 0.0162
14 ba'i 149 0.012
15 rnam 117 0.0094
16 pa'o 110 0.0088
17 gnyis 106 0.0085
18 rdo 98 0.0079
19 rang 87 0.0070
20 las 83 0.0067
21 rje 83 0.0067
22 phyag 66 0.0052
23 skyes 64 0.0052
24 AA' 63 0.0051
25 rgya 61 0.0049
26 lar 57 0.0046
27 lhan 55 0.0044
28 ltar 50 0.0040
29 dbang 30 0.0024
30 brjod 28 0.0023
31 gnas 24 0.0020
32 skabs 22 0.0018
33 yod 19 0.0011
34 don 12 0.0010
35 bde 12 0.0010
36 dzi 9 8.0E-4
37 phyir 9 8.0E-4
38 rtoq 4 4.0E-4
Total: 12519

```

Спектр корпуса.

В файле 101.txt находится текст на древнетибетском языке. В 101.val.txt текст, преобразованный на латиницу. В 101.table.val.txt находится таблица относительных частот для данного текста с учетом black листа, т. е. эта таблица не содержит слов, не несущих смысловую нагрузку (предлоги, частицы, местоимения, часто употребляемые слова и т. д.).

В файле 1TableFrequency.txt находится спектр исследуемого текста. Для построения этого спектра из таблицы относительных частот были взяты слова с наибольшим количеством вхождений в этот текст. Было подсчитано спектральное расстояние с использованием предварительно построенного спектра корпуса. При вычислении спектральных корпусов использовалось евклидово(квадратичное) расстояние.

Затем был найден радиус окрестности данного корпуса. В качестве радиуса окрестности было взято максимальное расстояние ≈ 0.36596 .

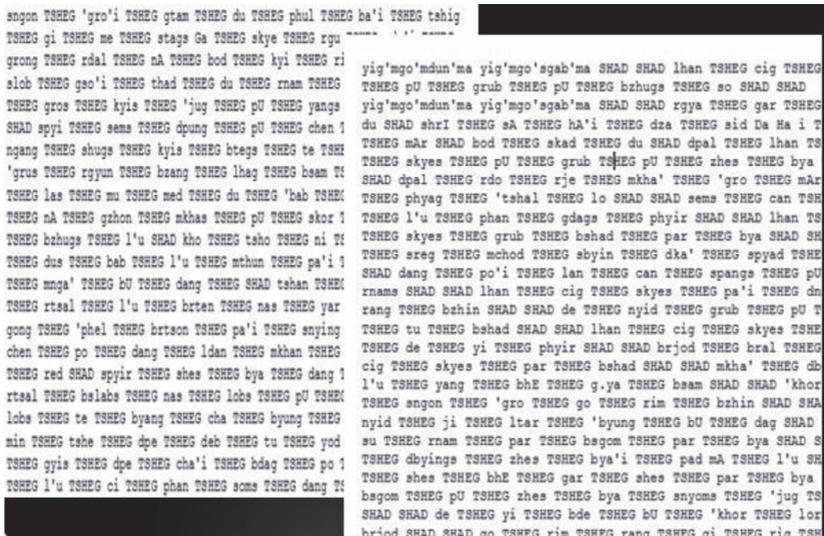
```

e[1]=0.2157934008083821
e[2]=0.24676378557815154
e[3]=0.18487447069000756
e[4]=0.1875650582663536
e[5]=0.18532588510004012
e[6]=0.16264610027083498
e[7]=0.20692598617849184
e[8]=0.20832576327485365
e[9]=0.22626804002176366
e[10]=0.2751668331877214
e[11]=0.2042132989941383
e[12]=0.2160338471713305
e[13]=0.2200566717939523
e[14]=0.17804438414762536
e[15]=0.23496302907366354
e[16]=0.26002171448666606
e[17]=0.19620370663657297
e[18]=0.2602948984924856
e[19]=0.2360659147337137
e[20]=0.24342289619363984
e[21]=0.3659656567902517
e[22]=0.2690414114680354
e[23]=0.21985444480080496
e[24]=0.3025285451796904
e[25]=0.2264300156329584
e[26]=0.2329391370131429
e[27]=0.24853617549904583
e[28]=0.20538524156308136
e[29]=0.2087648879087957
e[30]=0.2618996436264153
e[31]=0.28469038493917126
e[32]=0.26341183873424256
e[33]=0.20259178306828665
e[34]=0.34695916529420145

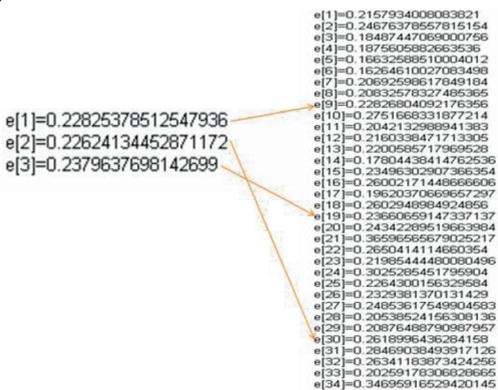
```

Также был произведен эксперимент. Были взяты тексты на древнетибетском тексте с неизвестной тематикой. Для них считались спектры. И затем смотрели, входят ли эти спектры в данный корпус. Оказалось, что все тексты с неизвестной тематикой входят в данный корпус.

Спектр первого текста, не входившего в корпус, близко расположен к 9 тексту из корпуса. Их расстояния почти равны. При визуальном сравнении двух этих текстов, видно, что эти тексты разные.



Спектры второго и третьего текста из эксперимента также находятся в исследуемом корпусе. Они расположены вблизи спектров 30 и 19 текстов соответственно.



Выводы

Данное исследование показало, что:

1. Метод экспертного статистического анализа применим к классификации древнетибетских текстов, а значит и для анализа больших объемов данных. Видно, что этот метод может быть эффективным и при использовании небольшого корпуса.
2. Небольшая временная сложность рассматриваемого алгоритма позволяет использовать его при анализе больших объемов данных для систем автоматического перевода с древних языков.

Литература

1. Барсегян А.А., Кутрянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. — СПб.:БХВ-Петербург, 2004. — 336 с.:ил.
 2. Суворцева Т.Г. Многомерный количественный анализ и классификация текстов на основе лингвостатистических характеристик
Режим доступа — Научная библиотека диссертаций и авторефератов disserCat <http://www.dissercat.com/content/mnogomernyi-kolichestvennyi-analiz-i-klassifikatsiya-tekstov-na-osnove-lingvostatisticheskikh#ixzz2TJXgpfr5>
 3. Data Mining — добыча данных. Режим доступа — http://www.basegroup.ru/library/methodology/data_mining/
 4. Чубукова И. Data Mining. Режим доступа — <http://www.intuit.ru/studies/courses/6/6/info>
-

Технологии и инструменты разработки программ и облачные вычисления



**Сафонов
Владимир Олегович**

д.т.н., профессор кафедры информатики СПбГУ
академик Американского биографического института (АБИ)
член-корреспондент РАН
заслуженный деятель науки и образования РАН

АБСТРАКТНЫЙ ЛЕКСИЧЕСКИЙ АНАЛИЗ

Вербицкая Е.А.

*студентка 3 курса кафедры системного программирования СПбГУ,
kajigor@gmail.com*

Григорьев С.В.

*магистр информационных технологий, аспирант, СПбГУ,
rsdpisuy@gmail.com*

Аннотация: Данная работа посвящена реализации генератора абстрактных лексических анализаторов, применяемых для анализа динамически формируемых выражений на встроенных языках.

Введение

Многие современные языки программирования позволяют динамически формировать в виде строк выражения на других языках и выполнять их. Наиболее распространённые примеры — встроенный SQL (частный случай — динамический SQL), генерация HTML, различные DSL. Основная проблема при работе с динамически формируемыми выражениями — отсутствие возможности статической проверки, и, как следствие, выявление ошибок лишь во время выполнения программы. Кроме того, при разработке с использованием встроенных языков полезными были бы средства для статической проверки синтаксической корректности и проверки типов, а так же упрощающие процесс разработки стандартные для IDE возможности такие как рефакторинг или автодополнение. Для реализации такой функциональности необходимым шагом является лексический анализ динамически формируемых выражений. При этом, для ускорения разработки необходимо иметь инструмент для автоматической генерации лексических анализаторов для работы со встроенными языками.

Реализация

В рамках проекта лаборатории JetBrains, целью которого является создание платформы для анализа кода на встроенных языках, был реализован генератор лексических анализаторов, который, в отличие от классических инструментов типа Lex, порождает лексический анализатор, принимающий на вход граф, который является компактным представлением множества значений динамически формируемого выражения. Результатом работы такого анализатора является граф, собранный из токенов, встречающихся в строках, содержащих встроенный код.

Шаги лексического анализа выглядят следующим образом:

- 1 на вход подаётся граф, составленный на основе дерева разбора исходного кода программы: на ребрах — фрагменты строк кода на встроенном языке, вершинам соответствуют случаи конкатенации строк;
- 2 входной граф преобразовывается к виду, удобному для лексического анализа: каждое ребро входного графа разбивается на последовательность новых ребер, метки которых содержат только по одному символу из строки исходного ребра;
- 3 запускается процедура токенизации на преобразованном графе, в результате чего получается новый граф, на ребрах которого находятся накопленные в результате работы абстрактного лексического анализатора токены, соответствующие грамматике анализируемого встроенного языка.

Рассмотрим следующий пример исходного кода:

```

IF @X = @Y
    SET @TABLE = '#tbl1'
ELSE
    SET @TABLE = 'tbl2'
SET @S = 'SELECT x FROM' + @TABLE
EXECUTE (@S)
  
```

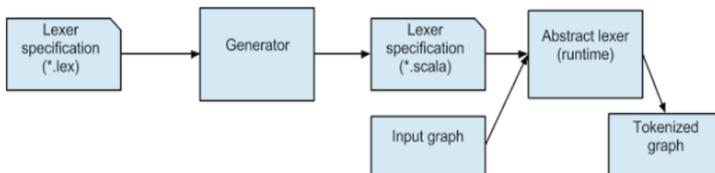
По нему будет построен следующий граф:



После применения к нему процедуры токенизации будет получен граф вида:



В качестве основы для генератора был использован генератор лексических анализаторов для Java — JFlex [1]. Для него реализована генерация управляющих данных (описание конечного автомата, пользовательский код) в язык программирования Scala [3]. При этом, непосредственно функция токенизации не генерируется, а реализована статически и параметризуется сгенерированными данными. Таким образом, полный цикл работы с инструментом выглядит следующим образом:



Абстрактный лексер основан на конечном преобразователе (finite state transducer, FST [2]), который основываясь на сгенерированном описании автомата, переводит входной граф, в граф, содержащий соответствующие спецификации токены.

Заключение

В рамках данной работы реализован абстрактный лексер на основе FST и генератор абстрактных лексических анализаторов на основе JFlex, который по стандартному описанию лексики языка генерирует набор данных для абстрактного лексера. В будущем планируется создать инструмент, предназначенный для обработки встроеного в код на Java кода на языке SQL.

Литература

- 1 JFlex — <http://jflex.de/>
 - 2 Mehryar Mohri: Finite-State Transducers in Language and Speech Processing. 1997 Association for Computational Linguistics — <http://www.cs.nyu.edu/~mohri/pub/cl1.pdf>
 - 3 Scala — <http://www.scala-lang.org/>
-

ОПТИМИЗАЦИЯ ПРЕДСТАВЛЕНИЯ БАЙТ-КОДА JVM ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ

Пилипенко А.В.

аспирант кафедры информатики

математико-механического факультета СПбГУ, artur.pilipenko@gmail.com

Аннотация: В докладе описан алгоритм сжатия байт-кода JVM, основанный на генерации новых инструкций для часто встречающихся последовательностей байт-кодов исходной программы. Этот алгоритм минимизирует суммарный размер программы и интерпретатора, необходимого для её исполнения. Наилучших результатов этот подход позволяет добиться в закрытой модели, когда весь исполняемый код доступен на этапе построения, и не стоит задачи обеспечить совместимость интерпретатора со стандартным байт-кодом.

Введение

Для встраиваемых систем выполняемые задачи не отличаются вычислительной сложностью, но общий размер программного обеспечения является критически важным параметром. Другой особенностью встраиваемых устройств является то, что в редких случаях на таких устройствах необходимо исполнять произвольные приложения. Обычно такое устройство имеет вполне конкретное применение, и конкретное приложение для исполнения. Этот факт позволяет использовать специализированное представление для Java приложений, без необходимости поддерживать совместимость со стандартным байт-кодом.

В докладе описывается алгоритм сжатия Java приложений, основанный на автоматической генерации специализированного набора инструкций на основе стандартного байт-кода JVM. [1] Специализированный набор инструкций минимизирует суммарный размер приложения и интерпретатора, необходимого для его исполнения. Он представляет собой подмножество стандартного байт-кода JVM, используемое приложением, дополнительное «суперинструкциями», кодирующими часто встречающиеся последовательности байт-кодов исходной программы. По сути, этот алгоритм является применением методов словарного сжатия к коду программ. Аналогичный подход к сжатию кода программ рассмотрен в работах [2, 3].

Генерация оптимизированного набора инструкций

Построение словаря

Процесс генерации специализированного набора инструкций начинается с построения словаря всех последовательностей байт-кодов исходной программы, которые могут быть использованы для создания «суперинструкций».

Это последовательности небольшой длины (обычно не более 10 инструкций) с одной точкой входа и выхода. Для каждой последовательности в словаре хранится то, сколько раз она встречается в исходной программе программы.

Для хранения словаря используется префиксное дерево, в узлах которого записываются байт-коды. Каждый узел такого дерева представляет элемент словаря — последовательность инструкций исходной программы, которую можно восстановить, выписав байт-коды всех вершины от узла до корня в обратном порядке.

Словарь наполняется в процессе итерации по байт-коду всех методов программы. В процессе итерации, на каждом шаге вычисляется множество последовательностей инструкций, которые оканчиваются в просматриваемой позиции. В начале просмотра каждого метода это множество содержит только пустую последовательность. На каждом последующем шаге множество состоит из пустой последовательности и всех последовательностей множества из предыдущего шага, дополненных текущим байт-кодом. В дерево добавляются последовательности, которые могут быть использованы для создания «суперинструкций». Так как все новые последовательности получаются путем добавления одного байт-кода к существующей последовательности, операции поиска и добавления в дерево сводятся к поиску и добавлению потомка для существующей вершины.

Выбор набора инструкций

Задача выбора оптимального словаря для методов словарного сжатия является NP-полной [4], поэтому на практике используются различные эвристики. Достаточно простым и эффективным является частотный подход, когда в словарь включаются самые часто встречающиеся подстроки исходного текста. Аналогичный подход используется и в описываемом алгоритме.

В байт-коде JVM все инструкции кодируются одним байтом, следовательно, набор инструкций ограничен 256 элементами. В набор включаются все инструкции исходного байт-кода, которые использует приложение, затем набор дополняется до 256 элементов последовательностями словаря с максимальным весом, где вес последовательности считается по следующей формуле:

$$\text{patternCount} * (\text{patternLength} - 1) - \text{interpreterSize}$$
, где
patternCount — сколько раз данная последовательность встречается в коде программы;
patternLength — длина последовательности в байтах;
interpreterSize — размер кода интерпретатора для данной последовательности.

Нахождение оптимального покрытия

Для заданного набора инструкций может существовать более одного представления исходной программы. Выбор оптимального представления

сводится к нахождению такого покрытия кода исходной программы последовательностями из набора инструкций, при котором суммарная длина использованных последовательностей максимальна. Оптимальное покрытие вычисляется отдельно для каждого метода.

Вычисление оптимального покрытия метода осуществляется последовательно для каждой позиции в теле метода. Для каждой позиции хранится сумма длин использованных последовательностей и указатель на последнюю использованную последовательность. Тогда, при известном оптимальном покрытии для всех предыдущих позиций нахождение оптимального покрытия для текущей позиции сводится к выбору последовательности, заканчивающейся в текущей позиции. Из множества последовательностей, заканчивающихся в данной позиции, выбирается такая последовательность, принадлежащая набору инструкций, для которой максимально значение

$$\text{length}(\text{pattern}) + \text{lengths}[\text{pos} - \text{length}(\text{pattern})],$$

где $\text{length}(\text{pattern})$ — длина последовательности;

$\text{lengths}[i]$ — сумма длин последовательностей оптимального покрытия для i -ой позиции;

pos — текущая позиция.

Представление метода в оптимизированном байт-коде восстанавливается с конца метода по указателям на использованные последовательности.

Заключение

Описанный в статье подход позволяет уменьшить размер байт-кода Java программы на 15-30% в зависимости от приложения. В отличие от многих других алгоритмов сжатия программ, описанная оптимизация не оказывает влияния на скорость исполнения программы.

Данный подход может быть применен и для открытой модели, например, для уменьшения размера системных классов. Но в открытой модели совместимость со стандартным набором инструкций существенно ограничивает количество свободных байт-кодов.

Литература

1. *Tim Lindholm, Frank Yellin, Gilad Bracha, Alex Buckley* «The Java[®] Virtual Machine Specification» <http://docs.oracle.com/javase/specs/jvms/se7/jvms7.pdf> [дата просмотра: 21.04.2013]
2. *Lars Raeder Clausen, Ulrik Pagh Schultz, Gilles Muller, Charles Conzel* «Java Bytecode Compression for Low-End Embedded Systems», ACM Transactions on Programming Languages and Systems, 2000
3. *Dimitris Saoungkos, George Manis, Konstantinos Blekas, Apostolos V. Zarras*, «Revisiting Java Bytecode Compression for Embedded and Mobile Computing Environments», IEEE Transactions on Software Engineering, 2005
4. *Michael Garey, David Johnson*, «Computers and Intractability: A Guide to the Theory of NP-Completeness», 1990

БЕСШОВНАЯ ИНТЕГРАЦИЯ АСПЕКТОВ В ОБЛАЧНЫЕ ПРИЛОЖЕНИЯ НА ПРИМЕРЕ БИБЛИОТЕКИ ENTERPRISE LIBRARY INTEGRATION PACK FOR WINDOWS AZURE И ASPECT.NET

Григорьев Д.А.

к. ф.-м. н., доцент кафедры информатики СПбГУ, gridmer@mail.ru

Григорьева А.В.

аспирант кафедры информатики СПбГУ, nastya001@mail.ru

Сафонов В.О.

д. т. н., профессор кафедры информатики СПбГУ, vosafonov@gmail.com

Аннотация: Библиотека Enterprise Library Integration Pack for Windows Azure — это решение компании Microsoft для выделения «сквозной функциональности» при разработке облачных приложений. Использование этой библиотеки подразумевает модификацию исходного кода целевого приложения. На практике возникают ситуации, когда изменение исходного кода нежелательно. Данная статья описывает методику бесшовной интеграции аспектов и целевого проекта с помощью Aspect.NET, которая позволяет не менять исходный код целевого приложения.

Введение

Любой программист сталкивался с задачей сопровождения кода. Как правило, для изменения поведения программы приходится вносить изменения в ее исходный код. Однако, в ряде случаев было бы удобно добавлять новую функциональность в систему бесшовным образом, не затрагивая исходный текст проекта. Предположим, например, что заказчик просит временно добавить новый тип отчета, либо один из пользователей сообщил о проблеме с производительностью на его нестандартной машине. Другим примером служит ситуация, когда разным пользователям требуется, чтобы одна и та же функциональность была реализована различным образом. Вместо создания множества разных веток кодовой базы в системе контроля версий, имеет смысл сохранить только одну, а желаемое поведение обеспечивать бесшовной интеграцией.

Аспектно-ориентированное программирование (АОП) предлагает решение данной проблемы — выделение «сквозной функциональности» в один модуль, аспект [3]. По своей сути, реализация «сквозной функциональности» состоит в том, чтобы выполнить некие действия в определенных точках целевой программы или же расширить существующие классы новыми методами и полями. Существенной проблемой при изменении по-

ведения программы с помощью современных АОП-инструментов является отсутствие бесшовной интеграции аспектов и целевого исходного кода системы. Причиной этого является распространенное мнение, что АОП наиболее эффективно только в том случае, когда целевая система изначально разрабатывалась с его помощью, напр. все зависимости между интерфейсами и реализующими их классами разрешаются через АОП-инструмент, а система декомпозирована таким образом, что все действия аспектов идеально соответствуют своим точкам внедрения. К сожалению, напр. для унаследованного кода этот принцип не работает, тем более при наличии требования бесшовной интеграции.

В 2011 году компания Microsoft выпустила расширение Enterprise Library (EL) Integration Pack for Windows Azure [4], которое содержит функциональные блоки для управления производительностью (Autoscaling Application Block) и ограничения функциональности под нагрузкой (Transient Fault Handling Block). Программист может реализовать ту или иную «сквозную функциональность», если в исходном коде своего проекта вызовет методы из набора классов соответствующего функционального блока. С точки зрения АОП очевидно, что местоположение этих вызовов в исходном коде целевого приложения является совокупностью точек внедрения для действий соответствующего аспекта.

Проект Aspect.NET, разрабатываемый коллективом авторов с 2004 г. в СПбГУ, представляет собой аспектно-ориентированную среду разработки программ для платформы Microsoft.NET. Аспекты определяются на метаязыке Aspect.NET ML [2], либо с помощью пользовательских атрибутов в отдельных проектах MS Visual Studio, а их слияние с целевым кодом происходит на уровне сборок статически, т.е. после этапа компиляции. Вначале компилируется сборка с аспектами из которой компоновщик аспектов (weaver) извлекает правила внедрения каждого действия аспекта, содержащиеся в его пользовательском атрибуте AspectAction(). Затем компоновщик анализирует MSIL-код откомпилированной сборки целевого проекта, находит соответствующие места внедрения (сканирование) и вставляет туда действия из аспектной сборки. Операции сканирования и внедрения действий аспектов разделены, что позволяет пользователю просматривать и фильтровать точки внедрения. Весь анализ и модификация .NET сборок производится с помощью сервисов отражения (reflection) и библиотеки MS Phoenix [11], которая декомпилирует сборку и представляет ее в виде набора высокоуровневых инструкций.

Постановка задачи

Перед авторами была поставлена задача реализовать бесшовную интеграцию функциональных блоков EL с помощью Aspect.NET. Это позволило бы объединить сильные стороны двух инструментов, а именно: большой на-

бор готовых аспектов, приемлемая скорость результирующего кода и независимость целевого проекта от АОП-инструмента. Объектом исследования были выбраны Autoscaling Application Block и Transient Fault Handling Block, как наиболее полезные для разработки облачных приложений на платформе MS Azure.

Реализация

Рассмотрим упражнение «Hands-on Lab 1: Using the Logging Application Block with Windows Azure Storage» [5], где путем добавления ссылок на сборки EL производится подключение функционального блока логгирования к исходному проекту, а затем вызов его метода для передачи сообщения в облачное хранилище диагностической информации WAD. Это дает возможность настраивать параметры сбора и хранения отладочных сообщений через графический интерфейс Logging Application Block, либо через его конфигурационные файлы.

Итак, наша задача заключается в том, чтобы перенести все зависимости от EL и вызовы методов протоколирования в отдельный проект с аспектом. Применив затем с помощью Aspect.NET данный аспект к исходному проекту, мы получим его бесшовную интеграцию с Logging Application Block.

По мнению авторов [7], перехват вызовов методов, которые реагируют на внешние события, может быть осуществлен через наследование классов. Если в аспектном проекте создать класс, который наследует от целевого класса, а затем подменить им свой базовый класс в сборке исходного проекта, то требуемый перехват можно осуществить в переопределенном виртуальном методе. Специальный пользовательский атрибут [ReplaceBaseClass] предписывает компоновщику Aspect.NET заменить целевой класс своим аспектным наследником:

1. Заменить в исходной сборке все вызовы методов базового целевого класса (в том числе и конструкторы) на вызовы методов его наследника в аспектной сборке.
2. Принудительно объявить виртуальными те методы целевого класса, которые переопределены в замещающем его наследнике. Если они закрыты (`private`), то сделать их защищенными (`protected`).
3. Если вызов этих методов в исходной сборке производится с помощью MSIL-инструкции `call` или `ldftn`, заменить их на `callvirt` и `ldvirtftn` соответственно.
4. Объединить с помощью инструмента ILRepack (из проекта Mono.Cecil [10]) сборки с аспектом и исходную.
5. Присвоить какое-нибудь служебное имя базовому целевому классу, а его первоначальное имя — замещающему наследнику из аспекта.

Преимуществами такого алгоритма является простота подмены классов для пользователя, а также использование только штатного синтаксиса

языка .NET. Теперь с помощью аспекта можно: уточнять поведение любого метода целевого класса, реализовывать в нем дополнительные интерфейсы, накладывать различные пользовательские атрибуты и т. п. Вычислительная сложность такого алгоритма $O(N)$, где N — количество MSIL-инструкций в исходной сборке. Однако, в компоновщике аспектов Aspect.NET эти операции объединены с операциями вставки аспектов (также сложностью $O(N)$), поэтому общая асимптотическая сложность компоновки аспектов не увеличилась.

Следующий пример бесшовной интеграции основан на примере «Hands-on Lab 11: Transient Fault Handling». Здесь задача заключается в том, чтобы добавить в целевой код работы с базой данных стратегию обработки исключительных ситуаций. Сама стратегия отделена от кода, работающего с базой данных, и конфигурируется средствами EL. Например, для любого запроса к базе данных можно составить стратегию вида: попытаться совершить 4 последовательных запроса, если каждый из предыдущих совершается неудачно. При этом между вторым и третьим запросом должна быть пауза в 5 сек.

Запрос к базе данных производится с помощью вызова метода `SqlCommand.ExecuteReader()`, который может выбросить исключение при сбое соединения с базой данных. Чтобы применить к нему стратегию обработки исключительных ситуаций, необходимо выполнить данный блок в рамках метода `ExecuteAction()` класса `Microsoft.Practices.TransientFaultHandling.RetryPolicy<T>`, где T — класс, реализующий стратегию. Получить объект этого класса можно через специальный менеджер `TransientFaultHandling.RetryManager`, который должен быть инициализирован библиотекой EL и передан нам в конструкторе целевого класса. Для этого требуется создать объект нашего замещающего наследника с помощью средств EL. Средства АОП-программирования, в том числе и Aspect.NET, предоставляют механизмы замены вызова целевого метода на действие аспекта. Для подмены создания класса замещающего наследника мы этим и воспользовались, однако, в вышеупомянутом случае ситуация осложняется тем, что “целевым” является весь блок `using`. Для таких ситуаций и предназначен наш механизм замены целевого класса замещающим наследником.

Заключение

Бесшовная интеграция аспектов в целевые приложения дает возможность эффективно решать проблемы сопровождения проектов, когда необходимо добавлять новую функциональность без изменения исходного кода. Представленная методика позволяет применить сторонние библиотеки для устранения сквозной функциональности, избегая при этом зависимости от них в исходном коде целевого проекта. В ходе дальнейшей разработки можно безболезненно отказаться от использования выбранной библиотеки, либо

заменить ее на другую. Таким образом, существенно снижаются риски того, что неудачно выбранное АОП-решение или библиотека повлекут за собой существенную переделку целевого проекта. Реализация данной методики на базе Aspect.NET позволяет программистам использовать привычную среду разработки MS Visual Studio 2012, а также простой процесс создания аспектов, предоставляющих доступ к сервисам сторонних библиотек. Все упомянутые аспекты доступны на сайте проекта Aspect.NET [9].

Литература

1. *Эспозито Д.* Аспектно-ориентированное программирование, перехват и Unity 2.0 // MSDN Magazine, 12.2010 // Режим доступа [проверено 22.04.2013]: <http://msdn.microsoft.com/ru-ru/magazine/gg490353.aspx>
2. *Григорьев Д.А.* Реализация и практическое применение аспектно-ориентированной среды программирования для Microsoft .NET // Научно-технические ведомости // СПб. Изд-во СПбГПУ. 2009. № 3., — 225 с.
3. *Сафонов В.О.* Аспектно-ориентированное программирование // Учебное пособие // СПб.: Изд-во СПбГУ. 2011, — 28 с.
4. Сайт проекта Enterprise Library 5.0 Integration Pack for Windows Azure // Режим доступа [проверено 22.04.2013]: <http://entlib.codeplex.com/wikipage?title=EntLib5Azure>
5. Сайт проекта Hands-On Labs for Enterprise Library 5.0 Integration Pack for Windows Azure // Режим доступа [проверено 22.04.2013]: <http://www.microsoft.com/en-us/download/details.aspx?id=28785>
6. *Григорьева А.В.* Аспектно-ориентированный рефакторинг облачных приложений MS Azure с помощью системы Aspect.NET // Компьютерные инструменты в образовании // СПб. Изд-во АНО «КИО». 2012. № 1., — 21 с.
7. *Григорьев Д.А., Григорьева А.В., Сафонов В.О.* Беспшовная интеграция аспектов в облачные приложения на примере библиотеки Enterprise Library Integration Pack for Windows Azure и Aspect.NET // Компьютерные инструменты в образовании // СПб. Изд-во АНО «КИО». 2012. № 4., — 3 с.
8. Сайт проекта Mono.Cecil // Режим доступа [проверено 22.04.2013]: <http://www.mono-project.com/Cecil>
9. Сайт проекта Aspect.NET // Режим доступа [проверено 22.04.2013]: <http://aspectdotnet.org/>
10. Сайт проекта Mono.Cecil // Режим доступа [проверено 22.04.2013]: <http://www.mono-project.com/Cecil>
11. Сайт проекта MS Phoenix // Режим доступа [проверено 22.04.2013]: <http://research.microsoft.com/en-us/collaboration/focus/cs/phoenix.aspx>

К ВОПРОСУ О РАЗРАБОТКЕ И РЕАЛИЗАЦИИ ИНСТРУМЕНТА ВЫДЕЛЕНИЯ АСПЕКТОВ ИЗ НЕ АСПЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММ

Сиунов А.Э.

аспирант, каф. информатики, СПбГУ

Сафонов В.О.

д. т. н., проф.

Аспектнo-ориентированное программирование (АОП) — относительно новое, но, несомненно, перспективное направление, в основе которого лежит разработка и модификация сквозной функциональности в больших программных системах. АОП предназначено для «модуляризации сквозной функциональности и ее автоматизированного, безопасного и надежного добавления в целевую программу, а также поиска (локализации) и модификации в целевой программе некоторой уже реализованной сквозной функциональности» [1].

Центральным понятием в АОП является аспект, понимаемый как «реализация какой-либо сквозной функциональности», как «модуль, применение которого осуществляется не путем вызова, как для процедуры или метода, а путем систематизированного внедрения фрагментов кода аспекта в рассредоточенные модули целевой программы» [2].

Основными задачами АОП являются не только реализация и внедрение новой сквозной функциональности, но и, во-первых, выделение аспектов из не аспектнo-ориентированных программ с целью их последующего многократного использования (аспектизация) и, во-вторых, модификация сквозной функциональности, представленной определением аспекта и всеми его внедрениями [1].

При достаточно динамичном развитии аспектнo-ориентированного программирования в целом, задача выделения аспектов является на настоящий момент наименее реализованной — пока не существует общепризнанного специалистами и пользователями инструмента аспектизации. В рамках нашего исследования предполагается разработка и реализация аспектизатора — инструмента выделения аспектов из не аспектнo-ориентированных программ.

Поскольку основной целью АОП является внедрение определенных аспектов в целевую программу таким образом, чтобы при сохранении ее основной структуры можно было использовать добавленные в различные заданные ее точки новые функции с сохранением безопасной и надежной работой программы, для решения задачи аспектизации необходимо сначала выявить, какие аспекты не аспектнo-ориентированных программ были востребованы и с помощью какого инструментария данные аспекты будут выделяться.

Выделение аспектов сталкивается с рядом проблем:

- 1) какие методы действительно могут помочь в автоматизированном выявлении аспектов;
- 2) как определить, какая функциональность должна быть реализована с помощью аспектов;
- 3) как оценить качество автоматической аспектизации [3].

Говоря об оценке качества аспектизации, можно разделить задачу на два вопроса: насколько хорошо обнаруживаются аспекты и как много выявляется «неправильных» аспектов, то есть таких, которые пользователь, выбирая из предложенных, считает неподходящими.

Делая обзор подходов к выделению аспектов, исследователи [4] приходят к выводу о том, что на данный момент, несмотря на непрекращающиеся попытки обнаружить наилучшие методы аспектизации, однозначного подхода не выявлено. Существующие методы недостаточно точны, у них много ложных обнаружений аспектов, все еще требуется большое количество вовлечений пользователя, и эти методы не могут быть интегрированы с инструментами рефакторинга.

В статье [3] описан следующий алгоритм аспектизации: вычислим количество методов, в которых встречается вызов рассматриваемого метода. Чем это число больше, тем вероятнее, что данный метод может быть частью аспекта [3]. При этом необходимо исключить разнообразные часто встречающиеся по природе своей методы типа `set` и `get`, `toString` и др.

На наш взгляд, аспектизатор может быть реализован двумя путями. Первый — это поиск типичных аспектов, похожих на уже известные стандартные аспекты, например, проверка прав доступа, протоколирование и обработка ошибок. Второй путь предполагает, что аспектизатор сам делает попытки анализировать код. В частности, необходимо выделить те места, где вероятнее всего есть аспект. Таких мест может быть достаточно много. Например:

- 1) одинаковые имена переменных и вызовов методов с идентичными именами в совершенно разных частях программы;
- 2) те методы, где в начале или в конце есть вызов методов суперкласса;
- 3) похожие конструкции `try-catch` в разных частях программы и др.

Сквозная функциональность предполагает, что в методе выполняются те действия, которые логически не являются действием класса, не относятся к его задаче. Так, они могут проверять некоторые входные параметры или вызывать внешние методы, которые не изменяют при этом самих параметров. Такие действия можно выделить как действия аспекта, выполняющегося до метода или вместо него. Сквозная функциональность, выполняющаяся после действий метода, может предполагать дообработку входных данных, что также в ряде случаев может быть выделено как аспект.

В некотором отношении на действие аспекта похож шаблон проектирования IoC (Inversion of Control). Поэтому управляющий класс этого ша-

блона можно представить как действие аспекта. Также на наличие аспектов следует анализировать статические методы и паттерн Singleton, поскольку в них часто может встречаться выполнение проверок и обработка выходных данных, что тоже можно представить как действие аспекта.

Таковы в общих чертах пути разработки и реализации инструмента выделения аспектов из не аспектно-ориентированных программ.

Литература

1. Сафонов В.О. ASPECT.NET — инструмент аспектно-ориентированного программирования для разработки надежных и безопасных программ // Компьютерные инструменты в образовании. 2007. №5. С. 3–13.
 2. Сафонов В.О. Практическое руководство по системе аспектно-ориентированного программирования ASPECT.NET. Часть 1. // Компьютерные инструменты в образовании. 2008. №3. С. 20–33.
 3. Marin M., van Deursen A. and Moonen L. Identifying aspects using fan-in analysis. In: Proceedings 11th Working Conference on Reverse Engineering (WCRE), pp. 132–141, IEEE
 4. Cojocar G.S. Aspect mining: past, present, future. STUDIA UNIV. BABEȘ BOLYAI, INFORMATICA, Volume LVII, Number 4, 2012.
-

Рандомизация, усреднение и мультиагентные технологии в data mining и управлении



**Граничин
Олег Николаевич**

д.ф.-м.н.

профессор кафедры системного программирования СПбГУ
заведующий лабораторией
стохастических вычислительных систем НИИИТ СПбГУ

РАСПОЗНАВАНИЕ ЯЗЫКА ЖЕСТОВ НА ВИДЕО ПОТОКЕ

С.А. Землянская

*студент кафедры системного программирования СПбГУ,
svetlana.zemlyanskaya@gmail.com*

Аннотация: Статья посвящена разработанному алгоритму распознавания языка жестов на видео потоке. Подробно описан этап отслеживания движения на видео, локализация ладони и определения временных отрезков смены жестов. Приведён метод классификации на основе опорных векторов и множество признаков на которых производилась классификация жеста.

Введение

С развитием технологий размеры компьютеров, используемых в повседневной жизни, стремительно уменьшается, это приводит к необходимости разработки нового удобного способа ввода информации. Одним из наиболее заметных направлений развития области является ввод данных через визуальные системы с помощью жестов. Интерес к этому подходу связан, в первую очередь, с природным удобством использования такого интерфейса и богатством новых возможностей, которые он привносит.

На текущий момент существует множество разработок в данной области, тем не менее, нельзя считать эту тему полностью исчерпаной. Хорошие результаты достигнуты в направлениях, использующих дополнительные атрибуты при записи видео, такие как маркерные перчатки [1], дополнительные сенсоры — несколько камер или кинект [2], позволяющие построить карту глубины и в дальнейшем работать с трехмерным изображением. Дополнительная информация повышает точность конечного результата, но увеличивает сложность ввода информации, чего хотелось избежать в текущем исследовании. Другой успешно применяемый подход — использование цепей маркова при классификации жестов [3], даёт хорошие результаты на контрольном наборе из динамических жестов, каждый из которых представляет собой последовательный переход из одного состояния в другое. В работе в качестве итогового множества жестов были выбраны статические жесты, к которым такой подход не применим.

В ходе исследования был разработан алгоритм, позволяющий распознавать жесты на видеопотоке, снятом с одной веб камеры. Важным требованием было минимизировать ограничения на входное видео и записывающие устройства для увеличения простоты использования конечного приложения. Алгоритм был реализован на языке octave с использованием встроенных библиотек компьютерного зрения.

Постановка задачи

Целью данной работы было разработать и реализовать алгоритм, который получает на вход видеопоток, снятый с одной веб-камеры, разбивает его на отдельные жесты, каждый из которых классифицирует в рамках выбранного подмножества из азбуки глухонемых. Видео может содержать разнородный фон, шумовое или повторяющееся движение на заднем плане. От требования, что бы в кадр попадала только рука было решено отказаться, в связи с тем, что оно значительно снижало удобство записи входных данных. Итоговым множеством классифицируемых жестов было выбрано подмножество латинской азбуки языка глухонемых состоящее из 6-ти букв.

Алгоритм

Входная информация — видео поток с одной веб-камеры. Видео может содержать разнородный фон и шумовые движения на заднем плане.

1. Предварительная обработка изображения

Каждый кадр переводится в бинарный и с помощью адаптивного алгоритма [4] вычитается фон. На полученной карте движения убирается шум и выделяются компоненты связности для каждой из которых считается вес.

2. Локализация ладони

Для каждой компоненты связности на кадре подсчитывается её вес до тех пор, пока значение не превысит пороговое. Считаем область наибольшей активности движения — ладонью.

На последующих кадрах, в зависимости от того, является ли жест зафиксированным или происходит его смена, область ладони корректируется.

3. Разделение видео на жесты

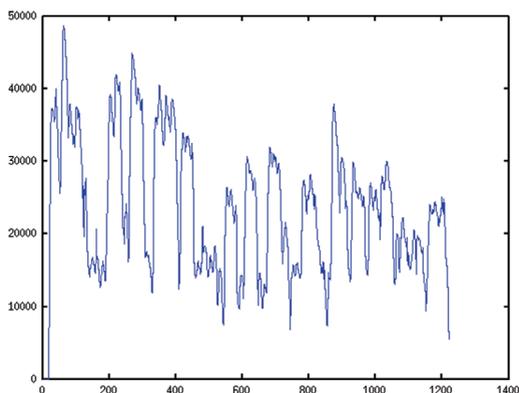


Рис. 1. Изменение веса наибольшей компоненты от времени

В работе используется адаптивный метод вычитания фона и, как результат, резкие пики на графике постепенно затухают. Возрастание активности движения является началом смены жеста, а его затухание признак того, что жест устоялся и его можно классифицировать.

4. Классификация жеста

В качестве признаков классификации была выбрана гистограмма направленных градиентов [5], размерность которой была уменьшена с помощью метода главных компонент [6].

Жест классифицируется с помощью метода опорных векторов [7].

Алгоритм вычитания фона

Принцип, лежащий в основе алгоритмов вычитания фона, состоит в том, что определённый кадр или группа кадров выбирается за эталонный фон. Все последующие кадры сравниваются с ним и, если модуль разницы яркостей для пикселя превосходит порог, то пиксель считается передним планом, иначе — фоном. В работе используется адаптивный метод основанный на смеси нормальных распределений [4].

Фоновое значение для каждого пикселя представлено в виде набора троек $[w, \mu, \delta^2]$: веса пикселя, математического ожидания и дисперсии, вес — представляет собой меру схожести между значением пикселя и фоном. В дальнейшем, будем называть такую тройку «процессом» пикселя. Такое представление позволяет алгоритму адаптироваться к зашумлённому фону. В текущей работе для каждого пикселя запоминается 5 его процессов, знания которых он принимает с течением времени.

На первом кадре видео происходит инициализация модели: $[1, c, \delta_{\text{фик}}^2]$, где c — яркость пикселя, $\delta_{\text{фик}}^2$ — выбранное начальное значение.

Для каждого последующего кадра, для каждого пикселя производится ряд шагов:

1. Измеряется различие между поступившим пикселем и его фоновыми значениями:

$$dif = \frac{\|c - \mu\|}{\delta}$$

- Если различие меньше порога, то процесс объявляется текущим.
- Если больше, то инициализируется новый процесс и, в зависимости от количества уже созданных процессов, либо добавляется в конец набора, либо заменяется процесс с наименьшим весом. Новый процесс помечается как текущий.

2. Значения текущего процесса обновляются:

$$\mu_t = (1 - \alpha_1) * \mu_{t-1} + \alpha_1 * c$$

$$\delta_t^2 = (1 - \alpha_2) * \delta_{t-1}^2 + \alpha_2 * (c - \mu_t)^2$$

$$w_t = (1 - \alpha_3) * w_{t-1} + \alpha_3$$

Для всех остальных процессов пикселя обновляется только вес:

$$w_t = (1 - \alpha_3) * w_{t-1},$$

где $\alpha_1, \alpha_2, \alpha_3$ — *выбранные параметры алгоритма, влияющие на скорость адаптации.*

3. Пиксель классифицируется как фоновый, если вес превышает выбранный порог и как принадлежащий переднему плану в противном случае.

Метод опорных векторов

Метод опорных векторов является алгоритмом обучения с учителем, т. е. предварительно определён набор признаков для которых мы знаем предполагаемый результат классификации.

Классификация по этому методу происходит с помощью разделения точек различных классов гиперплоскостью [8]. Таких гиперплоскостей может быть много, поэтому в качестве меры качества выбранной задается «зазор» между классами. Если существует гиперплоскость, разделяющая классы с максимальным зазором, то она называется оптимальной разделяющей гиперплоскостью, а соответствующий ей линейный классификатор называется оптимально разделяющим классификатором.

Строим разделяющую гиперплоскость, которая имеет вид:

$$w * x - b = 0,$$

где w — перпендикуляр к разделяющей гиперплоскости.

Отдельно введём понятие «ошибка классификации» для каждого элемента — e_i . Результат классификации, в таком случае, может принимать два значения: $y_i \in \{-1, 1\}$. Согласно такой классификации каждый вектор x должен удовлетворять такому условию:

$$y_i * (w * x_i - b) \geq 1 - e_i.$$

Для нахождения максимально возможного зазора между классами при таком линейном разделении необходимо минимизировать следующую функцию:

$$\frac{1}{2} * \|w\|^2 + \sum_{i=1}^n e_i \rightarrow \min.$$

Заключение

В результате проделанной работы, было написано приложение по распознаванию подмножество латинской азбуки жестов. На вход приложение получает видеопоток с одной вебкамеры, содержащий зашумлённое видео с неоднородным фоном. После обработки, видео разделяется на отдельные жесты, каждый и которых классифицируется в рамках выбранного множества жестов.

Для оценки классификации использовались три метрики: precision, recall и процент правильно классифицированных жестов:

Procent	Precision	Recall
89.44%	0.89142	0.96203

Работа выполнена при частичной финансовой поддержке РФФИ (грант № 13-07-00250-а).

Литература

1. *Aran, Oya* «Vision based sign language recognition: modeling and recognizing isolated signs with manual and non-manual components», 2008
2. *Simon Lang* «Sign Language Recognition with Kinect», 2011
3. *Jörg Zieren, Karl-friedrich Kraiss* «Robust person-independent visual sign language recognition», 2005
4. *Chris Stauffer, W. Eric L. Grimson* «Learning Patterns of Activity Using Real-Time Tracking», 2000.
5. *O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes*, 'Trainable «Classifier-Fusion Schemes: An Application To Pedestrian Detection», 2009
6. *Karl Pearson* «On Lines and Planes of Closest Fit to System of Points in Space», 1901
7. *Ванник В.Н., Червоненко А. Я* «Теория распознавания образов», 1974
8. *Hsu, Chih-Wei; Chang, Chih-Chung; and Lin, Chih-Jen* «A Prectical Guide to Support Vect Classification », 2003.
9. *Воскресенский А.Л., Ильин С.Н., Zelezny* «О распознавании жестов языка глухих», 2010.
10. *Абакумов В. Г., Ломакина Е.Ю.* «Автоматическое распознавание жестов в интеллектуальных системах », 2010.
11. *Шapiro Л., Стокман Дж. С.* «Компьютерное зрение », 2006.
12. *Форсайт Д., Понс Ж.* «Компьютерное зрение », 2004.
13. *Tom M. Mitchell* «Machine Learning », 1997.
14. *Lemeshow, David W. Hosmer* «Applied logistic regression », 2000.
15. *A. Barr, Edvard A. Feigenbaum* «The Handbook of Artifical Intelligence », 1990.
16. *Gary Bradski, Adrian Kaehler* «Learning OpenCV », 2008.
17. *Simon Lang* «Sign Language Recognition with Kinect», 2011
18. *Navneet Dalal and Bill Triggs* «Histograms of Oriented Gradients for Human Detection», 2004

РАССМОТРЕНИЕ ПОДХОДОВ К РЕАЛИЗАЦИИ СИМУЛЯТОРА КВАНТОВЫХ ВЫЧИСЛЕНИЙ НА КЛАССИЧЕСКОЙ АРХИТЕКТУРЕ

Коноплев Ю.М.

*студент кафедры системного программирования СПбГУ,
239konkos@gmail.com*

Сысоев С.С.

доцент кафедры системного программирования СПбГУ, sysoev@petroms.ru

Аннотация: Математическая модель квантовых вычислений позволила найти полиномиальные алгоритмы для некоторых сложных вычислительных задач. На данный момент авторам известно не более десятка квантовых алгоритмов, дающих различную степень ускорения по сравнению с классическими. Столь малое количество результатов за более чем 30 лет исследований, возможно, объясняется тем, что квантовые компьютеры до сих пор остаются чисто математической моделью.

В данной статье рассмотрены подходы к реализации симулятора квантового компьютера на классическом. Наиболее эффективный из рассмотренных подходов применен авторами для разработки симулятора универсального квантового компьютера. Прототип симулятора доступен в виде веб-приложения в сети Интернет и может быть использован в учебных и научных целях.

Введение

Развитие вычислительной техники — это постоянные эксперименты и открытия. Всем известен закон Мура, говорящий об удвоении вычислительных мощностей классических компьютеров каждые полтора года. Вычислительные машины прошли путь от механических устройств до ЭВМ, основанных на полупроводниках и микросхемах. Каждый новый тип устройства был эффективнее предыдущего — естественно было бы ожидать появления новых, еще более эффективных устройств.

Вместе с устройствами эволюционировали и алгоритмы. В 1936 году Алан Тьюринг предложил математическую модель вычислений, названную впоследствии машиной Тьюринга (Deterministic Turing Machine — DTM) [1]. Модель Тьюринга формализовала понятие алгоритма (способа вычисления некоторой функции на физическом устройстве), что позволило ввести понятие вычислимости и, в дальнейшем, сложности вычислений. Тогда же, в 1936 году, Тьюринг показал, что существуют невычислимые (undecidable) функции — такие функции, которые можно формально определить, но для вычисления которых невозможно предложить алгоритм, работающий конечное время.

Понятие сложности вычислений (количества ресурсов, необходимых для выполнения алгоритма) позволило выделить классы задач, названных неразрешимыми (intractable). В отличие от невычислимых функций, для них существуют алгоритмы, работающие за конечное время. Однако ресурсы, необходимые для выполнения этих алгоритмов, растут слишком быстро (например, экспоненциально) с увеличением размера входных данных. Открытым является вопрос, являются ли неразрешимыми задачи из класса NP-complete (знаменитая гипотеза $P \neq NP$).

Одной из неразрешимых в рамках классической теории вычислений задач является эмуляция эволюции квантовой системы произвольного размера. Основываясь на этом факте, в 1981 году нобелевский лауреат по физике Ричард Фейнман предложил построить на основе квантовой системы вычислительное устройство. Он предполагал, что система, которую сложно рассчитать на классической архитектуре, может сама оказаться эффективным вычислителем (у квантового компьютера не возникает проблем с эмуляцией самого себя).

Уже в 1985 году появилась работа Дэвида Дойча [2], в которой был предложен первый алгоритм, позволяющий использовать преимущества квантовых вычислений. В 1994 году вышла знаменитая статья Питера Шора [3], описывающая полиномиальный алгоритм факторизации чисел.

В 1996 году Гровером [4] был получен важный результат относительно возможности решения на квантовом вычислителе обобщенной модели задачи из класса NP, при полностью неопределенном виде функции (черном ящике).

Несмотря на высокий интерес к квантовым вычислениям, вызванный алгоритмами Шора и Гровера, на текущий момент существует всего несколько квантовых алгоритмов. Одной из возможных причин подобного положения дел авторы считают практическую сложность разработки алгоритмов для квантового компьютера. Мы надеемся упростить исследователям этот процесс, предложив эффективный общедоступный эмулятор квантовых вычислений.

Целью данной работы была реализация эмулятора квантового вычислителя на классическом компьютере в виде веб-приложения. Подобный эмулятор позволяет «потрогать руками» алгоритмы, что должно способствовать лучшему освоению теории квантовых вычислений студентами, а также разработке новых алгоритмов.

Основные понятия

Кубит, система из нескольких кубитов

Простейшей квантовой системой является вектор в двумерном Гилбертовом пространстве, называемый кубитом. Его можно представить в виде суперпозиции двух базисных векторов:

$$A*|0\rangle + B*|1\rangle \quad (1)$$

Система из нескольких кубитов описывается, как тензорное произведение систем её образующих:

$$(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\alpha\beta_0|0\rangle + \beta_1|1\rangle) \otimes \dots$$

Из этого следует, что размерность пространства системы из n кубит 2^n . Отсюда вытекает сложность моделирования квантового компьютера на классическом.

Измерение состояния квантовой системы

Классический бит можно легко измерить. С кубитом всё не так просто. Он находится не в одном из двух состояний (0 или 1), а в их суперпозиции. На данный момент не существует способов определить коэффициенты в (1). При измерении кубита мы можем получить как 0, так и 1, с некоторой вероятностью. Вероятность каждого исхода определяется, как квадрат модуля коэффициента при базисном векторе:

$$\begin{aligned} |A|^2 & \text{— вероятность при измерении получить } 0 \\ |B|^2 & \text{— вероятность при измерении получить } 1 \end{aligned}$$

Эволюция квантовой системы, квантовые вентили

Эволюция квантовой системы описывается, как действие на нее некоторого унитарного оператора U . Этот оператор можно разложить в композицию операторов меньшей размерности.

Далее приведены примеры простейших однокубитных операторов:

- Тожественное преобразование:

$$\delta_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Отрицание:

$$\delta_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Фазовый сдвиг:

$$\delta_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Преобразование Адамара:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Также возможны вентили, имеющие более одного входа. Например, матрица контролируемого отрицания (C-NOT) имеет вид:

$$C-NOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Набор квантовых вентилях называют универсальным, если любое унитарное преобразование можно аппроксимировать с любой заданной точностью конечной последовательностью вентилях из этого набора.

Например набор, состоящий из отрицания, фазового сдвига, преобразования Адамара, поворота на $\pi/2_n$ и C-NOT, является универсальным.

Постановка задачи

Целью данной работы была разработка эмулятора квантового вычислителя в виде веб-приложения.

Вся работа включала в себя две основные задачи:

1. Эмуляция действия квантовых вентилях
2. Разработка пользовательского интерфейса

Темой данной статьи является только первая задача.

Решение

Простая схема вычислителя

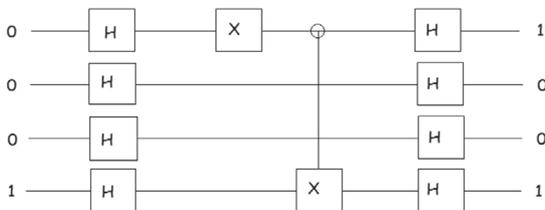


Рис. 1.

На рис.1 представлена схема квантового вычислителя с четырьмя рядами.

Для квантового компьютера применение однокубитного вентиля — это просто действие на один кубит системы. При симуляции на классическом компьютере — это действие матрицы размерностью $2^n \times 2^n$ на вектор, имеющий размерность 2^n , где n — разрядность вычислителя. В данном случае при $n=4$ размерность равна 16. Но, например, при $n=20$ размер матрицы становится больше 10^{12} .

Первый подход (наивный)

В предыдущей части говорилось, что состояние квантовой системы из n кубит — тензорное произведение систем (кубит) её образующих. Исходя из этого можно показать, что матрица оператора, действующая на систему — это тензорное произведение единичных матриц 2×2 и матрицы самого оператора. Если оператор должен быть применён к кубиту с номером j , то в произведении матрица оператора должна находиться на j -м месте:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \dots \otimes H_j \otimes \dots$$

Чтобы найти состояние системы после применения оператора, нужно просто умножить вектор состояния системы на полученную матрицу оператора.

Минусы подхода:

1. Затраты памяти на хранение матриц размерностью $2^n \times 2^n$.
2. При подсчёте матрицы оператора происходит огромное количество лишних элементарных умножений на 0 и 1.
3. Контролируемые операторы (например, CNOT) не раскладываются в тензорное произведение элементарных вентилях, и для формирования их матрицы нужен другой алгоритм.

Данный подход был реализован. Эмуляция производилась на обычном офисном ПК с объемом оперативной памяти 2 Гб. Было выяснено, что система работает при разрядности вычислителя не более 12 кубит. При 13 кубитах выдавалась ошибка памяти.

Мы решили более пристально изучить процесс действия матрицы оператора на систему.

Второй подход

Рассмотрим состояние системы не как вектор размерности 2^n , а как суперпозицию базисных векторов пространства

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n-1\rangle$$

Далее будем рассматривать действие оператора не на всю систему, а только на базисные вектора с соответствующими коэффициентами.

Таким образом, мы будем на каждом шаге вычисляться не вся матрица, а только один столбец, соответствующий данному базисному вектору.

Так же стоит заметить, что часто большая часть коэффициентов при базисных векторах равна нулю. Для таких векторов мы вообще не будем производить никаких вычислений, так как на результат они не влияют.

В ходе работы было выяснено, что каждый столбец матрицы оператора имеет довольно простой вид. В нём не нулевыми будут один или максимум

два элемента, который высчитываются исходя из матрицы оператора для одного кубита (2×2). Соответственно и операцию умножения нужно совершать только с этими ненулевыми элементами.

Результат применения оператора к системе рассчитывается следующим образом:

$$\alpha_0 \cdot h_0 + \alpha_1 \cdot h_1 + \alpha_2 \cdot h_2 + \dots$$

Здесь h_j — вычисленные нами столбцы.

Таким образом, мы избежали лишних умножений, ускорив процесс примерно в 2^n раз.

На каждом шаге мы храним всего три вектора размерностью 2^n : исходное состояние системы, столбец матрицы оператора, результирующий вектор состояния системы.

Реализация описанного подхода на том же макете позволила эффективно эмулировать вычисления для систем до 24-х кубит включительно.

Результаты

В процессе работы было рассмотрено два способа представления квантовых вентелей для квантовой системы заданной размерности. Первый способ использует представление этих вентелей, как тензорное произведение матриц из универсального набора.

Второй подход потребовал более тщательного понимания процесса и позволил получить значительную экономию, как по времени, так и по ресурсам памяти.

В результате получен прототип эмулятора квантового вычислителя, доступный по адресу <http://qc-sim.appspot.com/>.

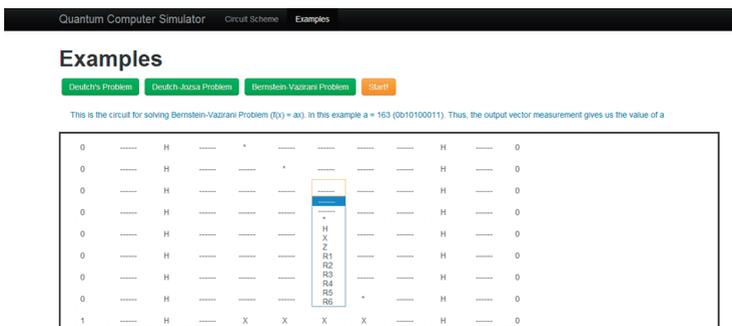


Рис. 2. Вид симулятора

Так же на сайте в качестве примеров представлены схемы вычислителей для алгоритмов Дойча, Дойча-Джозы и Бернштейна-Вазирани.

Литература

1. *Turing A.M.* On Computable Numbers, with an Application to the Entscheidungs Problem // Proceedings of the London Mathematical Society. 1936. Ser. 2, 42, p. 230–265.
 2. *David Deutsch* (1985). «The Church-Turing principle and the universal quantum computer». Proceedings of the Royal Society of London A 400: 97.
 3. *Peter W. Shor* (1994). «Algorithms for quantum computation: discrete logarithms and factoring». Proceedings of the 35th IEEE Symposium on Foundations of Computer Science. pp. 124 – 134.
 4. *Lov K. Grover* (1996). «A fast quantum mechanical algorithm for database search». Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212 – 219.
-

ПРИМЕНЕНИЕ ПРОЦЕССОВ СИНХРОНИЗАЦИИ В ОСЦИЛЛЯТОРНЫХ СЕТЯХ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАСТЕРИЗАЦИИ

А.В. Новиков

*аспирант кафедры компьютерных систем и программных технологий,
spb.andr@yandex.ru*

Е.Н. Бендерская

*к. т. н., доц. кафедры компьютерных систем и программных технологий,
helen.bend@gmail.com*

Санкт-Петербургский государственный политехнический университет

Аннотация: В статье исследуются процессы синхронизации, протекающие в осцилляторных сетях различной топологии, а также способы оценки локальной и глобальной синхронизации осцилляторов в сети. Модели осцилляторов в рассматриваемых сетях основаны на уравнении Курамото. Изложены способы практического использования осцилляторных сетей, при решении задач кластеризации N -мерных данных, графов, а также при решении задач раскрашенного графа.

Введение

Нейронные сети находят широкое применение в различных областях, позволяя решать различные задачи, такие как кластеризация, сегментация и распознавание образов, обеспечивая параллельный процесс вычислений, реализуя в своей структуре формальные математические алгоритмы. Таким образом, нейронные сети обеспечивают максимальное быстроедействие при использовании соответствующих аппаратных средств.

Процессы синхронизации, протекающие в головном мозгу, предположительно используются при реализации когнитивных функций, таких как зрение, двигательные функции, память [1,2]. Таким образом, возникает актуальность исследования процессов и режимов синхронизации в осцилляторных нейронных сетях, представляющие собой биологически правдоподобные мультиагентные системы, реализующие когнитивные функции головного мозга.

Модели фазовых осцилляторов Курамото

Модель фазового осциллятора Курамото позволяет исследовать процессы и режимы синхронизации между осцилляторами, которые зависят

от структуры и параметров нейронной сети. Модель Курамото описывается следующим уравнением [3]:

$$\frac{\partial \theta_i}{\partial t} = w_i + \frac{K}{N} \cdot \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i=1, \dots, N \quad (1)$$

Фаза осциллятора θ_i является основной переменной состояния и располагается в диапазоне от 0 до 2π . Внутренняя частота осциллятора w_i в данной модели представляет собой смещение по фазе, проявляющееся при процессе синхронизации. Сила связи K между осцилляторами в нейронной сети является основным параметром, оказывающим влияние на режимы синхронизации. Параметр N в уравнении (1) определяет количество осцилляторов в нейронной сети.

Режимы синхронизации и десинхронизации в уравнении (1) при использовании полносвязной модели сети обеспечивается выбором силы связи K . Высокое значение силы $K \geq N$ связи между осцилляторами обеспечивает быстрый выход в режим глобальной синхронизации, причем, чем выше значение силы, тем выше скорость выхода в режим синхронизации, как это представлено на рисунке 1. При выборе более низких значений K могут быть обеспечены режимы локальной синхронизации и глобальной десинхронизации на конечном промежутке времени. Важно отметить, что при рассмотрении динамики работы сети на бесконечном промежутке даже при бесконечно малых величинах силы связи режим глобальной синхронизации будет обеспечен в бесконечности.

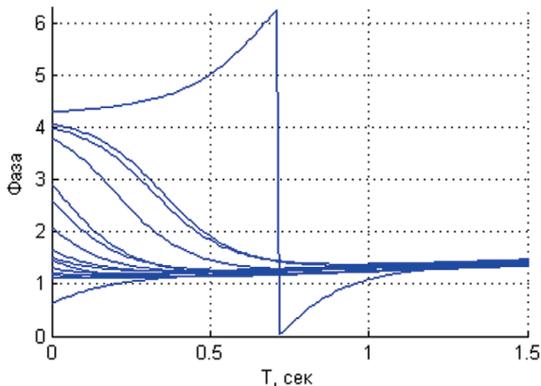


Рис. 1. Режим глобальной синхронизации ($r=0,998$) в осцилляторной сети ($K=100, N=15$)

Для оценки уровня синхронизации осцилляторов используется коэффициент r [3], используя значения которого можно определить текущий режим синхронизации осцилляторов в сети:

$$r = \left| \frac{1}{N \cdot e^{i \cdot \varphi}} \cdot \sum_{j=1}^N e^{i \cdot \theta_j} \right|, \quad \varphi = \frac{1}{N} \cdot \sum_{j=1}^N \theta_j \quad (2)$$

Ниже представлены значения уровня синхронизации соответствующие каждому из режимов синхронизации:

- Глобальная синхронизация наблюдается при $r \rightarrow 1$.
- Локальная синхронизация наблюдается при $r \rightarrow \sqrt{1 - K_c \cdot K^{-1}}$, где K — это сила связи между осцилляторами в нейронной сети, а K_c — это параметр роста, равный удвоенной ширине распределения частоты осциллятора:

$$K_c = 2 \cdot \gamma, \quad g(\omega) = \frac{\gamma}{\pi \cdot (\gamma^2 + (\omega - \omega_0)^2)} \quad (3)$$

- Полная десинхронизация наблюдается при $r \rightarrow 0$.

Изменение функции синуса для вычисления разности фаз между осцилляторами на степенную или экспоненциальную функцию позволяет обеспечить циклическое изменение режимов синхронизации. Свойство постоянного выхода из синхронного состояния представляет интерес при решении задач, в которых малое изменение в начальных условиях (состояние десинхронизации) приводит к большим изменениям в конце (например, количество локально синхронных ансамблей осцилляторов).

Важным достоинством использования модели осциллятора Курамото заключается в том, что режимы синхронизации обеспечиваются в сетях с различной структурой связей (отличных от полносвязной). В ходе экспери-

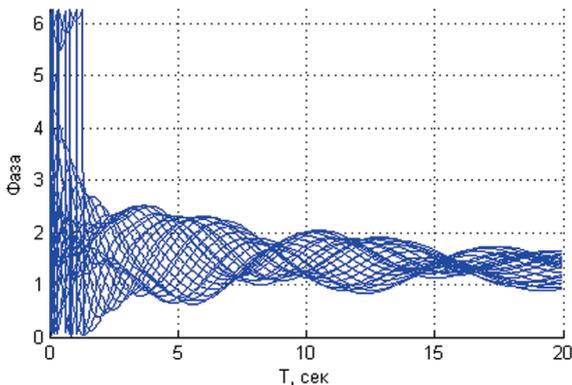


Рис. 2. Динамика фазовых осцилляторов ($K=100$) в сети со структурой однонаправленного кольцевого списка ($r=0,97$)

ментального исследования было установлено, что режим глобальной синхронизации успешно устанавливается в осцилляторных сетях со структурами типа сетки и двунаправленного списка, предложенных в работе Кумина и Ансворда [2]. Динамика сети со структурой однонаправленного кольцевого списка (также обеспечивающего режим глобальной синхронизации), предложенного в данной работе, представлена в качестве примера на рисунке 2.

Модели осцилляторных сетей, использующих уравнение Курамото, являются робастными. Для экспериментов, которые подтверждают данное утверждение, использовалась модель фазового осциллятора с шумом и двумя параметрами беспорядка: внутренним смещением фаз осцилляторов и силами связи между осцилляторами, выбранными по Гауссову закону распределения [8]:

$$\frac{\partial \theta_i}{\partial t} = w_i + \alpha \cdot \zeta_i(t) + \frac{K}{N} \cdot \sum_{j=1}^N \sin(\theta_j - \theta_i) \quad (4)$$

$$\frac{\partial \theta_i}{\partial t} = w_i + \alpha \cdot \zeta_i(t) + \frac{1}{N} \cdot \sum_{j=1}^N K_{ij} \cdot \sin(\theta_j - \theta_i + A_{ij}) \quad (5)$$

Наиболее интересный случай синхронизации возникает при использовании модели сети (5), который представляет собой режим синхронного пропорционального движения, который обуславливается в большей степени начальным состоянием фаз. Фазы осцилляторов при выходе в этот режим постоянно находится в динамическом движении так, что разность между всеми фазами остается постоянной, в том числе и при наличии шума, как это представлено на рисунке 3.

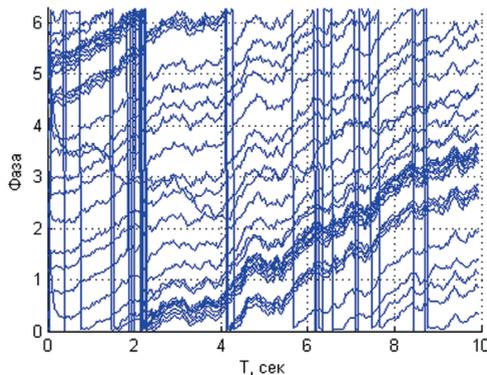


Рис. 3. Режим синхронизации в осцилляторной сети с параметрами беспорядка ($r=0,4$)

Таким образом, можно резюмировать, что используя различные вариации структур и параметров моделей осцилляторных сетей, основанных на уравнении Курамото, обеспечиваются режимы глобальной и частичной (локальной) синхронизации и десинхронизации. Один из основных параметров, влияющих на установление режима, является сила связи K между осцилляторами. Чем выше силы связи, тем выше уровень и скорость выхода в режим синхронизации. Соответственно при очень малых значениях сил связи или их отсутствии сеть будет работать в режиме полной десинхронизации. Ввиду того что модель является робастной, существуют широкие возможности по ее адаптации для решения практических задач, например, задач различных кластеризации, для которых требуется выводить сеть в режим локальной синхронизации.

Решение задачи кластеризации N -мерных данных

Режимы синхронизации в осцилляторных сетях могут использоваться, как было упомянуто ранее, для реализации различных задач. Предметом рассмотрения в данной статье является решение задач кластеризации с использованием процессов синхронизации.

Под задачей кластеризации понимается задача разбиения заданной выборки объектов на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из максимально схожих по параметрам объектов. Таким образом, решение задачи кластеризации, использующее нейросетевые принципы синхронизации, сводится к обеспечению в сети режимов локальной синхронизации, где каждый ансамбль синхронных между собой осцилляторов соответствует одному кластеру данных.

Рассмотрим задачу кластеризации N -мерных данных, используя осцилляторную сеть на базе модифицированной модели Курамото [4]:

$$\frac{\partial \theta_i}{\partial t} = \frac{K}{Nb_\epsilon(\theta_i)} \cdot \sum_{j \in Nb_\epsilon(\theta_i)} \sin(\theta_j - \theta_i), \quad i = 1, \dots, N \quad (6)$$

В отличие от базовой модели осциллятора Курамото (1) в представленной модели отсутствует смещение, задаваемое частотой осциллятора w_i , а вместо общего количества осцилляторов выбирается множество соседних осцилляторов $Nb_\epsilon(\theta_i)$, расположенные в радиусе ϵ относительно i -го осциллятора. При этом предлагается использовать постоянную силу связи K между связанными осцилляторами равную единице [4].

Структура осцилляторной сети для решения задачи кластеризации определяется набором входных данных. Каждый осциллятор ставится в соответствие одному объекту из множества входных данных, то есть количество осцилляторов в сети равняется количеству входных объектов и располагается

в N -мерном пространстве координат, соответствующих объекту, к которому привязан осциллятор. Связи устанавливаются между соседними осцилляторами, которые в досягаемости радиуса ϵ . На рисунке 4 представлен пример структуры осцилляторной сети соответствующей объектам в двухмерном пространстве.

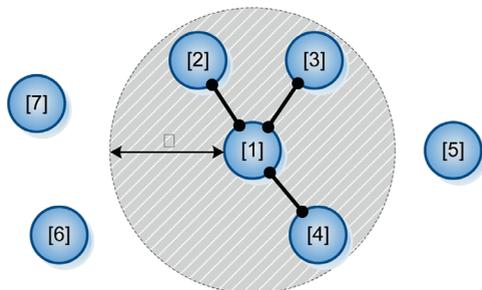


Рис. 4. Пример структуры осцилляторной сети в двухмерном пространстве для решения задачи кластеризации данных

Основным параметром, влияющим на результат решения задачи кластеризации, является радиус ϵ , из которого выбираются соседние осцилляторы. Таким образом, требуется экспертная оценка, которая позволяет выбрать радиус, иными словами входные данные должны быть представлены матрицей расстояний между объектами. В общем случае, можно использовать евклидово расстояние между осцилляторами.

Оценкой результатов кластеризации r_c является оценка локальной синхронизации фаз осцилляторов в сети [4], которая была модифицирована в рамках проведенного исследования:

$$r_c = \left| \frac{1}{Nb(I)} \cdot \sum_{i=1}^N \sum_{j \in Nb\epsilon(\theta_i)} e^{\theta_j - \theta_i} \right| \quad (6)$$

В представленной формуле $N_b(I)$ – общее количество соседней относительно каждого осциллятора. Локальная синхронизация или другими словами конечный результат синхронизации достигается при $r_c \rightarrow 1$.

В качестве выборок данных, на которых исследовалась работа осцилляторной сети по кластеризации данных, был общепризнанный набор Fundamental Clustering Problems Suite (FCPS) [5]. Эксперименты показали, что кластеры успешно выделяются на всех выборках (один из результатов работы при кластеризации выборки данных под названием Target представлен на рисунке 5), за исключением EnguTime, на которой два кластера одинаковой плотности в двухмерном пространстве пересекаются между

собой. Объясняется это тем, что кластеры образуются локальными осцилляторными сетями, которые строятся на базе входных данных, исходя из радиуса ϵ . Иными словами, чем выше связность между осцилляторами, тем выше вероятность их синхронизации между собой. Таким образом, кластерам соответствуют сильно связанные между собой локальные сети осцилляторов, которые могут иметь также связи между собой, но ввиду их малого количества — локальные ансамбли осцилляторов не оказывают друг на друга влияния.

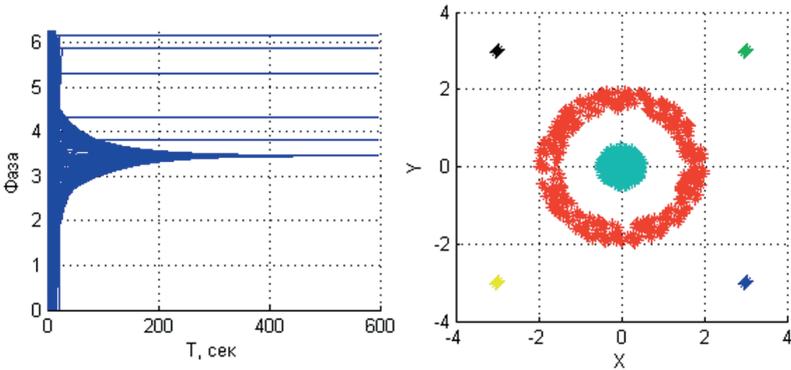


Рис. 5. Динамика работы осцилляторной сети — $r_c=0,99$ (слева) и соответствующие кластеры, которые были выделены (справа)

Из вышесказанного, очевидно, что при пересечении двух объектов с одинаковой плотностью, как в выборке EnguTime, будет выделен только один кластер, поскольку будет одна сильно связанная сеть. Стоит отметить, плохая связанность, обусловленная неверно выбранным радиусом, в осцилляторной сети может привести к появлению множества кластеров, что делает представленный алгоритм кластеризации чувствительным к радиусу выбора соседей ϵ .

Решение задачи раскрашенного графа

Особенности протекания процессов синхронизации в осцилляторных сетях, основанных на уравнении Курамото, позволяют адаптировать модель к решению задачи раскрашенного графа. Требуется раскрасить вершины графа минимально возможным количеством цветом таким образом, чтобы смежные вершины, имеющие связь, имели различные цвета:

$$\chi(G) = \min \{k : P(G, k) > 0\} \tag{8}$$

Модифицированная модель фазового осциллятора сети Курамото для решения задачи раскрашенного графа [6]:

$$\frac{\partial \theta_i}{\partial t} = \omega_i + \frac{1}{K_{MAX}} \cdot \sum_{j=1}^N K_{ij} \cdot \sin(\theta_j - \theta_i), \quad i = 1, \dots, N \quad (9)$$

$$K_{ij} = \begin{cases} K_p > 0, & e_{ij} \in \overline{E} \\ K_n < 0, & e_{ij} \in E \end{cases} \quad (10)$$

Принципиальным отличием в представленной модели (9) является наличие отрицательных K_n (ингибиторных) связей между осцилляторами. Значение K_{MAX} определяется как максимальная степень инверсного графа, на базе которого строится сеть.

Количество осцилляторов соответствует количеству узлов в графе, где каждому узлу сопоставляется свой осциллятор. Осцилляторная сеть является полностью связной, где отрицательная связь устанавливается между осцилляторами, вершины которых на графе имеют соединения. Положительная связь устанавливается между осцилляторами, вершины которых на графе не имеют соединения. Подобная архитектура позволяет десинхронизироваться между осцилляторами, которые соответствуют соединенными вершинам графа, и соответственно синхронизироваться с осцилляторами, чьи вершины на графе не имеют соединения.

Положительная K_p и отрицательная K_n связи являются ключевыми параметрами при решении задачи раскрашенного графа. Выбор этих параметров в значительной мере обеспечивает качество решения (режима локальной синхронизации) задачи раскрашенного графа. В данном случае ансамбль синхронных между собой осцилляторов соответствует одному цвету. В качестве рекомендации по настройке сети, предлагается для выделения вершин графа в отдельный кластер необходимо повышать значение положительной связи или (и) уменьшать значение отрицательной связи и в тоже время для разделения вершин по кластерам необходимо увеличивать значение отрицательной связи или (и) уменьшать значение положительной связи [6].

В ходе исследования динамики работы сетей на различной сложности графов было установлено, что решение задачи раскрашенного графа не всегда является оптимальным, как это представлено на рисунке 6, где минимальное количество цветов, которое достаточно для раскраски соответствует трем, в то время как сеть «раскрасила» граф четырьмя цветами. Отклонение от оптимального результата объясняется тем, что сформировавшиеся кластеры пытаются «перетянуть» друг у друга общие осцилляторы — тем самым выделяя их в отдельный кластер. Анализируя динамику на рисунке 6, очевидно, что вершина с номером 7 может войти в состав кластеров

{1, 4, 10} и {3, 6, 8}, в результате чего возникает конфликт между двумя этими ансамблями и образуется совершенно новый кластер на равноудаленном от них расстоянии, состоящий из одной вершины 7.

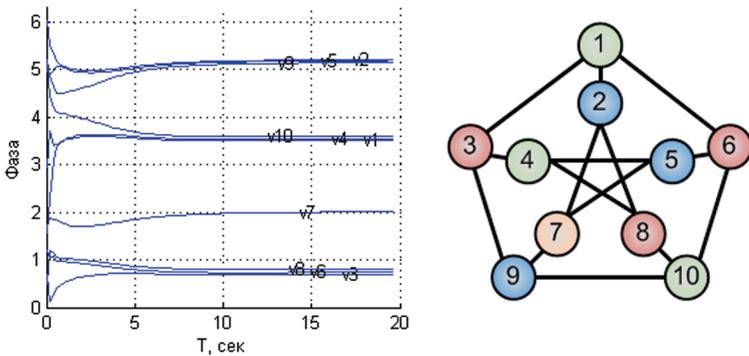


Рис. 6. Динамика работы осцилляторной сети при $K_p = 7$, $K_n = -14$, $K_{max} = 6$ (слева) и соответствующая раскраска вершин, соответствующие синхронным ансамблям осцилляторов (справа)

Кроме того подобные принципы решения, использующие модели Курамото, позволяют также осуществлять кластеризацию графов или сетей, выделяя подмножества сетей и подсетей во времени путем построения дендрограмм на основе динамики работы фазовых осцилляторов [7].

Заключение

В данном исследовании были проанализированы осцилляторные сети с различной структурой и различными параметрами, функционирующих на принципах синхронизации, в результате чего было выявлено, что осцилляторные сети, базирующиеся на уравнении Курамото, являются робастными и способны обеспечить различные режимы синхронизации в зависимости от выбранных параметров сети. Была предложена модель сети, использующая степенную функцию, для циклического перехода в режимы синхронизации и десинхронизации. Было показано, что ввиду робастности уравнения Курамото открывается широкий спектр по применению осцилляторных сетей, использующих принцип синхронизации, на практике.

Исследованы возможности осцилляторной сети по кластеризации N -мерных данных, в ходе которых было установлено, что результат кластеризации напрямую зависит от связности осцилляторов, которая обуславливается плотностью и распределением объектов в пространстве. Что в свою очередь является ограничением при решении задач, в которых имеется пересечение двух множеств данных разной формы, но одинаковой плотности.

В статье была рассмотрена адаптация модели Курамото путем введения ингибиторных связей для решения задач раскрашенного графа и указана основная причина отклонения от оптимального решения, которая заключается в «перетягивании» между двумя сформированными кластерами общих вершин в отдельный, равноудаленный кластер.

Тем не менее, по-прежнему открытым вопросом остается возможность по автоматизации методов кластеризации, использующих осцилляторные сети, с целью исключения экспертных оценок для настройки параметров сетей в зависимости от входных данных.

Литература

1. *Казанович Я. Б., Шматченко В.В.* Осцилляторные нейросетевые модели сегментации изображений и зрительного внимания / Научная сессия МИФИ-2004, часть 1, Нейроинформатика-2004. Лекции по нейроинформатике, стр. 15–68.
 2. *Cumin D., Unsworth C.P.* Generalizing the Kuramoto Model for the Study of Neuronal Synchronisation in the Brain / Report University of Auckland School of Engineering, 638, 2006.
 3. *Kuramoto Y.* Chemical Oscillations Waves, and Turbulence / Springer-Verlag Berlin Neidelberg New York Tokyo, 1984 — 157p.
 4. *Bohm C., Plant C., Shao J., Yang Q.* Clustering by synchronization / KDD '10 Proceeding of the 16th ACM SIGKDD international conference of Knowledge discovery and data mining, pp. 583–592, 2010.
 5. *Ultsch A.*, Clustering with SOM: U*C / in Proc. Workshop on Self Organizing Feature Maps, pp. 31–37, Paris 2005.
 6. *Wu J., Jiao L., Chen W.* Clustering dynamics of nonlinear oscillator network: Application to graph coloring problem / Physica D, 240, pp. 1972–1978, 2011
 7. *Wang X., Jiao L., Wu J.* Extracting hierarchical organization of complex networks by dynamics towards synchronization / Physica A, 388, pp. 2975–2986, 2009.
 8. *Daido H.* Quasientrainment and slow relaxation in a population of oscillators with random and frustrated interactions / Phys. Rev. Lett. 73, pp. 1073–1076, 1992.
-

РАСПОЗНАВАНИЕ НЕДЕФОРМИРУЕМЫХ ТРЕХМЕРНЫХ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ ПО КОНТУРАМ

Петров Н.С.

*студент 545 гр. кафедры системного программирования СПбГУ,
nikolai.s.petrov@gmail.com*

Аннотация: В данной работе будет дано сравнение известных методов распознавания трехмерных объектов по контурам и предложен новый метод, успешно применяющийся в задаче распознавания автомобилей. Данный метод устойчив к небольшим изменениям изображения объекта, таким как небольшие повороты и небольшие изменения в масштабе. В этом методе трехмерные объекты представляются конечным набором образцов, с которыми считается схожесть входного изображения. Метод основан на ориентациях градиентов изображения, поэтому слабо зависит от освещенности объекта. Преимуществом этого подхода также является возможность использования внутренних и частичных контуров.

Введение

Задача распознавания объектов возникает во многих сферах, например, в медицинских приложениях для распознавания типа кости на рентгене, или в криминалистической сфере для сличения объекта на изображении с объектом из базы данных, например распознавание лиц или машин.

Существующие методы рассматривают на два типа изменений модели: недеформирующие изменения и деформирующие. Первый вид изменений предполагает вращение, сдвиг и масштабирование модели, а второй – еще и деформации самой модели, таких как, например, изменение позы человеческого тела или мимики.

Предлагаемый ниже метод предполагает нахождение объекта при недеформирующих изменениях. Для этого из входного изображения изучаемого объекта извлекается его контур, затем находят особые точки контура, используя которые находится контур из базы контуров, построенных по трехмерным моделям.

Существующие решения

Задача распознавания объекта по его контуру в общем виде решается либо параметризацией контура и дальнейшим подсчетом функции схожести по двум параметризациям, либо подсчетом некоторого дескриптора контура (например, набор гистограмм) и сравнивая уже их.

В подходе Belongie и др. вводятся контексты формы контура и сравниваются уже они [1]. Выбирается равномерно n точек контура, из каждой точки пускаются отрезки во все остальные точки выборки и строится гистограмма по направлениям и длинам этих отрезков. Набор всех таких гистограмм и является контекстом формы контура. Сравнение двух контуров проходит накладыванием одного набора на другой со всевозможными смещениями, находя наилучший поворот одного контура к другому. Сложность этого подхода $O(n^3)$.

Подход Sebastian и др. основан на редакционном расстоянии, введенном ранее для строк [2]. Для каждой точки контура известно расстояние заранее определенной точки (начала контура) и кривизна в данной точке. При сравнении двух контуров функция схожести считается путем накладывания одного контура на другой и подсчетом разницы кривизны с возможностью выбрасывания кусков контуров с начислением штрафов.

Для сравнения контуров также можно использовать понятие граф шоков, как в подходе Mascini и др. [3]. По контуру строится его скелет в виде дерева как набор точек равноудаленных от пар точек контура. Узлами этого дерева и являются шоки, они бывают разных типов и силы. Сравнивая уже эти графы, можно сравнить сами контуры.

Вышеописанные методы могут помочь определить является ли изучаемый контур контуром самолета или молотка, но ими сложно различить схожие контуры недеформируемых объектов. Поэтому необходимо, что-то более зависящее от необычных точек контура и сильнее характеризующее конкретный объект. Плюс есть необходимость использовать внутренние контуры.

Предлагаемый подход

На входном изображении объекта, полученном с помощью камеры с известными параметрами без дисторсии, необходимо сначала выделить контур объекта, используя градиент изображения. Затем из точек контура оставить только особые и запомнить лишь направление градиента в этих точках. Затем отфильтровать некоторым способом полученные точки и уже их использовать при сравнении контуров.

Выделение контуров

Для выделения контуров использовался градиент изображения, посчитанный с помощью оператора Собеля. Обычно для этого используют яркость изображения в градациях серого, но для получения более явных и полных контуров используется такой метод: считаем градиент в каждой точке для трёх каналов отдельно и записываем как итог градиент с максимальной нормой.

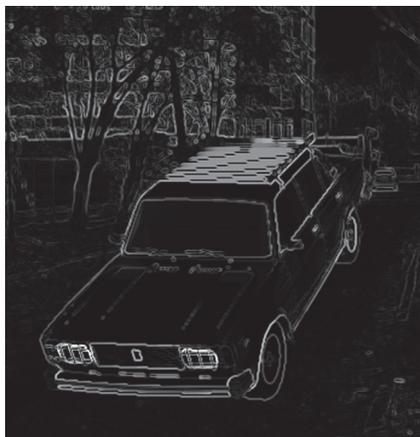


Рис. 1. Норма градиента изображения в градациях серого

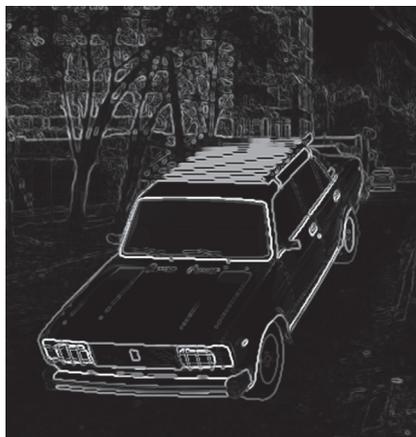


Рис. 2. Максимальная норма градиента изображения по трем каналам

Выделение особых точек

Среди точек полученного градиента изображения оставляем только те, норма градиента в которых больше определенного порога, таким образом, получая точки контура. Для этих точек градиента оставим только угол между направлением градиента и осью Ox . Если угол α больше π , то запишем вместо него $\alpha - \pi$, потому что градиент может быть направлен в противоположные стороны при различных фонах. Отказ от использования информации о норме градиента позволяет использовать точки контура, находящиеся в тени, тем же образом, каким точки контура в освещенной части объекта. Далее бинаризуем посчитанные углы в n бинов, то есть если угол α , $0 < \alpha < \pi/n$, тогда он попадает в первый бин, а если α , $\pi/n < \alpha < \pi * 2/n$, тогда во второй и так далее. Затем из точек удаляем те, для которых неверно, что направление в этой точке является самым частым направлением в некоторой небольшой окрестности точки. Это обеспечивает локальную сонаправленность точек контура и чистит контур от шума. Оставшиеся точки (направление и координаты) и будут использоваться для сравнения двух контуров.

Создание базы по трехмерным моделям

База контуров объектов используется для нахождения на входном изображении объекта и его ракурса. По некоторому набору трехмерных моделей генерируются их изображения в различных ракурсах и в разных масштабах. Затем на этих изображениях детектируются контуры и выделяются особые точки, описанным выше способом. Эти результаты можно сохра-

нить для дальнейшего переиспользования. Так же можно поделить эту базу на группы в зависимости от размеров контуров в пикселях.

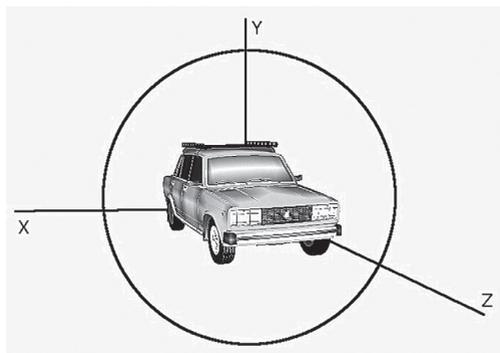


Рис. 3. Пример трехмерной модели в ракурсе повернутом по оси Oy на 20 градусов и по оси Ox на 10 градусов от фронтального

Функция схожести и ее подсчет

Назовем каждый контур из базы образцом модели в некотором ракурсе и масштабе. Сдвинем координаты точек в образцах так, чтобы абсцисса самой левой точки была нулем, а ордината самой верхней точки была тоже нулём. Тогда для входного изображения I в точке c и образца T можно ввести функцию схожести

$$E(I, T, c) = \sum_{r \in P} |\cos(\mathbf{ori}(O, r) - \mathbf{ori}(I, c+r))|,$$

где P — это множество точек образца, O — изображение, соответствующее образцу, $\mathbf{ori}(O, r)$ — бинаризованое направление градиента изображения O в точке r . Функция основана на работе Steger [5] и подобная использовалась в работе Farhan [4]. Проблема состоит в том, что такая функция совершенно не устойчива к изменениям, поэтому для каждой точки образца необходимо рассматривать некоторую окрестность R (например, квадрат 7×7 пикселей) вокруг прикладываемой точки:

$$E(I, T, c) = \sum_{r \in P} \left(\max_{t \in R(c+r)} |\cos(\mathbf{ori}(O, r) - \mathbf{ori}(I, t))| \right).$$

Но проведенные тесты показали, что эта функция явно зависит от количества точек контура в образце и если оно не постоянно, то возникает множество ошибок при распознавании. Поэтому конечная функция схожести имеет вид

$$E(I, T, c) = \sum_{r \in P} \frac{\left(\max_{t \in R(c+r)} |\cos(\text{ori}(O, r) - \text{ori}(I, t))| \right)}{n},$$

где n — количество точек в образце T .

Заключение

Для предложенного метода проведен ряд экспериментов: от ракурсов, присутствующих в базе сделан поворот по осям Ox , Oy , Oz на малый угол (2–3 градуса) и запущен поиск по базе ближайшего объекта и его ракурса. Результат считается положительным, если модель распозналась правильно и ракурс найден с точностью до 5 градусов. Результат по 15 моделям:

Поворот на малый угол	Распознано правильно 98,5%
Небольшие изменения масштаба	Распознано правильно 97,6%
Тест на размытие	Распознано правильно 87–95%
Преграждение четверти объекта	Распознано правильно 75–88%

Результат теста на размытие зависит от размера ядра размытия и его силы. Чем больше — тем ниже процент. При преграждении процент распознавания зависит от того какая часть объекта перекрыта (много ли на ней точек контура).

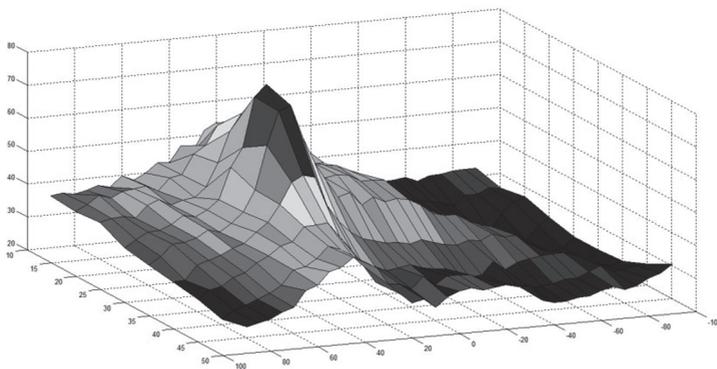


Рис. 4. График схожести тестового примера по базе (по осям — углы вращения по x и по y). Он показывает, что в области правильного ракурса явный пик функции, который можно найти быстрее, чем перебором всех ракурсов

Литература

1. *Belongie, S.; Malik, J.; Puzicha, J.*, «Shape matching and object recognition using shape contexts,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.24, no.4, pp.509,522, Apr 2002
 2. *Sebastian, T.B.; Klein, P.N.; Kimia, B.B.*, «On aligning curves,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.25, no.1, pp.116,125, Jan. 2003
 3. *Macrini, D.; Shokoufandeh, A.; Dickinson, S.; Siddiqi, K.; Zucker, S.*, «View-based 3-D object recognition using shock graphs,» *Pattern Recognition, 2002. Proceedings. 16th International Conference on* , vol.3, no., pp.24,28 vol.3, 2002
 4. *Farhan U.; Shun'ichi K.; Satoru I.*, «Object Search Using Orientation Code Matching», *IAPR Workshop on Machine Vision Applications*, Nov. 28–30, 2000
 5. *C. Steger*, «Occlusion Clutter, and Illumination Invariant Object Recognition,» in *International Archives of Photogrammetry and Remote Sensing*, 2002.
-

ВОССТАНОВЛЕНИЕ ГЛУБИНЫ ТОЧКИ ПРИ НАБЛЮДЕНИИ С ОДНОЙ КАМЕРЫ С ПОМОЩЬЮ РАНДОМИЗАЦИИ ЕЕ ПОЛОЖЕНИЯ

Кривоконь Д.С.

dmitry00@gmail.com

Вахитов А.Т.

alexander.vakhitov@gmail.com

Аннотация: В статье представлен рандомизированный алгоритм для оценки глубины точки на основе наблюдений с одной камеры. Рандомизация положения камеры позволяет решить задачу в случае движения точки без наложения существенных ограничений на характер ее движения, подавить помехи в наблюдениях оптического потока и ошибки дискретизации. Статья содержит результаты работы алгоритма, как на модельных данных, так и в реальных условиях.

Введение

В задачах наблюдения при помощи видеокамер часто возникает необходимость оценки геометрии окружающей среды. К примеру, для успешного управления роботом требуется оценивать положения находящихся в области видимости объектов и препятствий, аналогичные задачи возникают в приложениях автоиндустрии, где необходимо вычислять положения пешеходов, автомобилей и т.п. Для решения задач в случае, когда окружающая среда статична (например, при наблюдении роботом), существуют методы построения карты, основанные на детектировании особых точек на кадрах видео последовательности [1] и оценки оптического потока в них [2,3]. Положения камер и точек в таких алгоритмах вычисляются при помощи минимизации суммарной ошибки перепроектирования [4].

Интересным подходом к задачам реконструкции являются динамические системы. Так авторы [5] предложили несколько асимптотических наблюдателей, которые оценивают глубину точки за счет известного оптического потока между кадрами. Они доказали сходимости их алгоритма и показали, что при помощи него можно получить адекватные оценки глубины на реальных данных в контролируемых условиях.

В нашей статье мы предлагаем метод типа PASA [6] для оценки глубины одной точки, основанный на подходе [5]. Используя случайные возмущения положения центра проектирования камеры, мы оцениваем глубину точки при помощи дискретной процедуры. Использование рандомизации позволяет избежать от неоднозначностей, появляющихся в задаче, подавить помехи, ошибку дискретизации, а также оценивать положение точки в случае ее движения.

Последнее свойство алгоритма является крайне важным. Предыдущие подходы к решению задачи реконструкции в случае движения так или иначе использовали некие ограничения на возможное движение объекта. Так в [7] авторы рассматривают линейно движущийся объект, а в [8] представлено решение для точки, движущейся в некоторой плоскости. Для сходимости нашего подхода необходимо ограничить движение объекта с течением времени. Такое предположение можно считать разумным для большинства реальных приложений.

Методы типа RASA

RASA (рандомизированные алгоритмы стохастической аппроксимации) были предложены в работах Граничина О.Н. [6,9] и J.C. Spall [10]. Это методы стохастической аппроксимации типа алгоритма Робинса-Монро разработанные специально для задач оптимизации. RASA представляет собой итеративную процедуру, которая на каждой итерации корректирует текущую оценку при помощи аппроксимации градиента по случайному направлению. Для данной статьи важны результаты о сходимости методов RASA с вероятностью 1 в случае неизвестной, но ограниченной помехи наблюдения [6,9], а также результаты о оптимальной скорости сходимости в классе алгоритмов стохастической оптимизации [11]. Также представляет интерес алгоритм с постоянным размером шага, представленный в [12] и примененный для задачи отслеживания минимума нестационарного функционала.

Постановка задачи

Для простоты вывода мы будем рассматривать сферическую модель камеры, все последующие уравнения и сам алгоритм могут быть переформулированы для проективного случая при помощи простой замены координат. В каждый момент времени движущаяся система координат, привязанная к камере, задается положением центра проектирования камеры C_n в некоторой фиксированной системе координат и кватернионом q_n , который задает ее ориентацию. Пусть $v_n = C_n - C_{\{n-1\}}$ — скорость движения центра камеры, а ω_n — ее скорость поворота. Далее рассмотрим движущуюся точку наблюдаемую камерой. Ее положение определено лучом η_n (пиксель) из центра камеры к точке и глубиной D_n относительно этого луча. Дискретизируя уравнение динамики глубины из [5], можно получить:

$$D_n = D_{\{n-1\}} - \langle v_n, \eta_n \rangle + \epsilon_n,$$

где ϵ_n представляет собой ошибку дискретизации и возможный дрейф точки. При введенных обозначениях задачу отслеживания положения точки

можно сформулировать как задачу получения таких оценок \hat{D}_n , что $\hat{D}_n \rightarrow D_n$. Наш алгоритм основан на использовании данных об оптическом потоке в каждый момент времени. Глубина точки и оптический поток связаны следующим уравнением:

$$w_n = \Gamma_n (\langle v_n, \eta_n \rangle \eta_n - v_n) - \omega_n \times \eta_n + \epsilon_{w_n},$$

где ϵ_{w_n} — это помеха оптического потока.

Алгоритм

Мы предлагаем следующую процедуру для получения оценок \hat{D}_n истинного значения глубины D_n :

1. Сгенерировать случайное возмущение Δ_n по равномерному распределению на сфере радиуса 1 в трехмерном пространстве.
2. Сдвинуть камеру согласно пробному возмущению.
3. Сделать наблюдение оптического потока w_n .
4. Обновить \hat{D}_{n-1} используя:

$$\hat{D}_n = \hat{D}_{n-1} - \langle v_n, \eta_n \rangle + \alpha (1 - \hat{D}_{n-1} \hat{\Gamma}_n), \tag{1}$$

$$\text{где } \hat{\Gamma}_n = w_n \frac{\langle \Delta_n, \eta_n \rangle \eta_n - \Delta_n}{\| \langle \Delta_n, \eta_n \rangle \eta_n - \Delta_n \|}.$$

Уравнение (1) является дискретной аппроксимацией асимптотического наблюдателя представленного в [5]. Идея алгоритма заключается в том, что $\hat{\Gamma}_n$ будет аппроксимировать истинное значение обратной глубины. Рассмотрим уравнение (1) В нем есть члены, которые каким-то образом включают в себя Δ_n . Если мы скалярно домножим все выражение на $\langle \Delta_n, \eta_n \rangle \eta_n - \Delta_n$, то некоторые члены будут включать в себя Δ_n линейно, а некоторые квадратично. Линейные члены будут равны нулю после усреднения, а единственный квадратичный член будет содержать как множитель $\hat{\Gamma}_n$, что и позволяет получить формулу для оценок.

Моделирование

Условия

Мы рассматриваем несколько сценариев в нашем тестировании. В каждом из них предполагается, что положение и ориентация камеры точно известны в каждый момент времени. В качестве начального положения камеры мы используем начало координат, а ее скорость движения либо равна

нулю, либо меняется между $(1, 0, 0)$ и $(-1, 0, 0)$ с периодом в 10 итераций. Для простоты мы не рассматриваем изменения ориентации камеры, т.к. оно несущественно влияет на алгоритм оценивания. Начальное положение точки устанавливается в $(0, 0, 10)$, а ее динамика разбивается на два случая: стационарная точка и линейно движущаяся точка с постоянной скоростью $(0.1, 0.1, 0.1)$. Начальное приближение глубины точки равно 100, а начальная ошибка, соответственно 90. Наконец, мы рассматриваем три случая помехи в оптическом потоке:

1. Нулевая помеха
2. Равномерная с мат. ожиданием 0
3. Смещенная равномерная

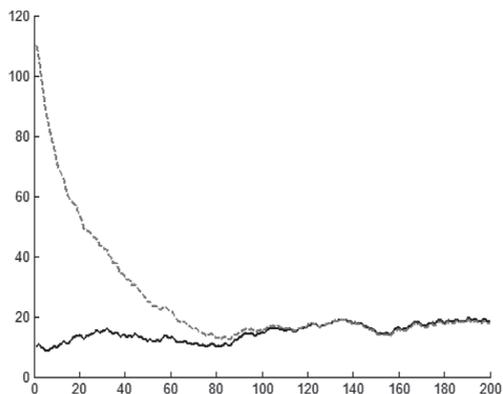


Рис. 1. Стационарный случай без помех

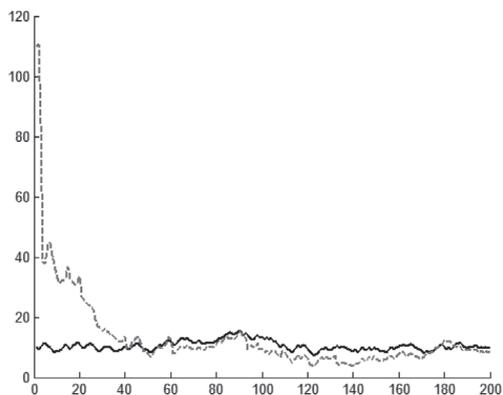


Рис. 2. Стационарный случай с равномерной смещенной помехой

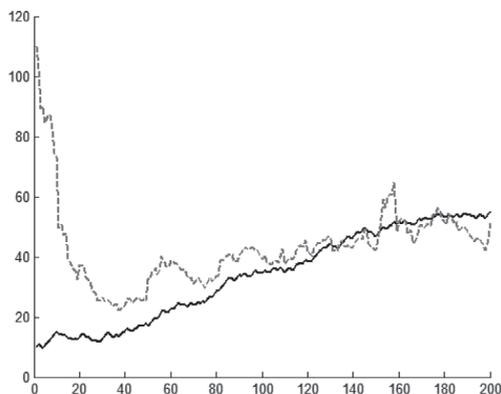


Рис. 3. Случай линейно движущейся точки и смещенной помехи

Результаты

Алгоритм тестировался на последовательностях данных в 200 итераций. Из наших тестов мы получили, что алгоритм подавляет ошибку начального приближения в большинстве случаев к 80-й итерации. На основе этого для оценки качества работы алгоритма мы использовали среднюю ошибку оценок, начиная с 100 итерации (для более стабильного определения). Таблица 1 содержит результаты тестирования. Каждое значение в ней подсчитано усреднением по 1000 запускам алгоритма. На Рис. 1, 2, 3 показаны примеры траекторий.

Условия теста	Помеха № 1	Помеха № 2	Помеха № 3
Стац. камера, стац. точка	0.1844	2.0609	3.8970
Стац. камера, движущаяся точка	2.5218	7.8300	9.8289
Движущаяся камера, стац. точка	2.7666	3.4075	4.5668
Движущаяся камера, стац. точка	7.4645	8.8753	11.3551

Таблица 1. Результаты моделирования

Реальный эксперимент

Условия

В реальном эксперименте первостепенной задачей является определение положения объекта и камеры, т.к. эти данные необходимы для последующей оценки качества работы алгоритма. Для решения этой задачи мы использовали подход на использовании специальных маркеров (см. Рис. 4).



Рис. 4. Пример изображения с маркерами

Каждый маркер представляется собой плоский объект известной геометрии, соответственно, зная положение его проекции на изображении можно определить положение камеры относительно него при помощи стандартных методов реконструкции. Основываясь на этом, мы построили нашу систему тестирования следующим образом.

Для определения движения камеры используется набор из 8 маркеров, для которых известны их положения относительно друг друга (большее количество маркеров необходимо для большей точности). Движущийся объект представлен одним маркером. Определяя сначала положение камеры относительно базы маркеров, а потом положение камеры относительно объекта, мы получаем всю необходимую нам информацию.

Случайность в данном эксперименте симулируется вручную при помощи периодического движения камеры. Такой подход в целом удовлетворяет условиям работы алгоритма, т.к. движения камеры в среднем будет равно нулю и не будет коррелировано с движением объекта.

Результаты

Алгоритм тестировался для двух случаев: случай со статическим объектом и случай с движущимся объектом. В каждом из них он показал результаты, согласующиеся с моделированием. Помеха в начальном приближения с течением времени подавлялась, после чего алгоритм успешно отслеживал истинную глубину точки. Примеры траекторий и соответствующих им графиков ошибки показаны на Рис. 5, 6, 7, 8.

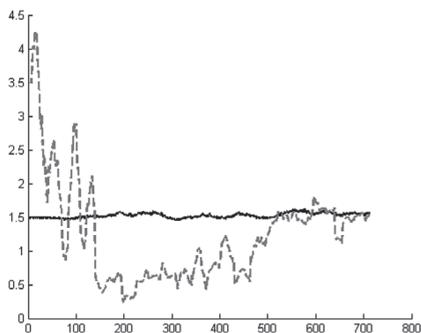


Рис. 5. Траектория глубины для статического случая

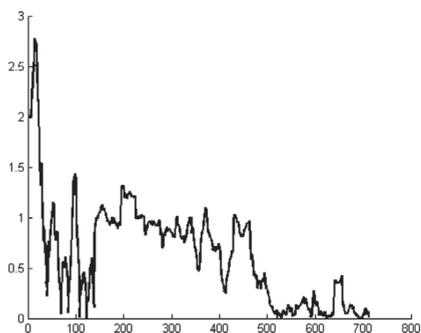


Рис. 6. График ошибки для статического случая

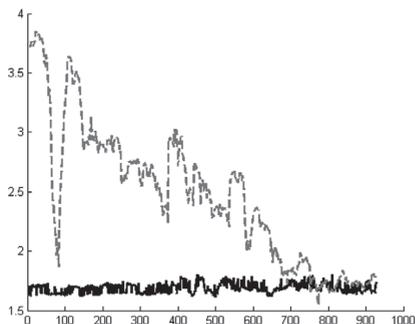


Рис. 7. Траектория глубины для случая с движением

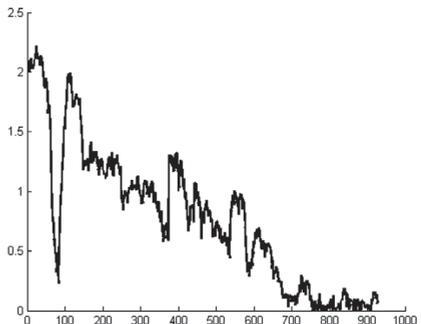


Рис. 8. График ошибки для случая с движением

Заключение

В статье был представлен алгоритм для оценки глубины при помощи одной камеры на основе наблюдения оптического потока. Были показаны результаты моделирования и тестирования на реальных данных. В обоих случаях алгоритм успешно справлялся с задачей отслеживания глубины, даже в тех вариантах, где присутствовали смещенные помехи и движение объекта. В будущем планируется добавить в алгоритм шаг предсказания движения объекта, а также изучить возможность применения алгоритма к задаче оценки геометрии статической сцены в целом.

Работа выполнена при частичной финансовой поддержке РФФИ (грант № 13-07-00250-а).

Литература

1. *Klein G., Murray D.* Parallel tracking and mapping for small AR workspaces // In: Proc. of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007. P.225–234.
2. *Lucas B.D., Kanade T.* An iterative image registration technique with an application to stereo vision // In: Proc. of the 7th International Joint Conference on Artificial intelligence. 1981.
3. *Horn B.K.P., Schunck B.G.* Determining optical flow // Artificial intelligence. 1983. 17(1). P. 185–203.
4. *Lourakis M., Argyros A.* The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-marquardt Algorithm. Technical Report 340. Institute of Computer Science-FORTH. Heraklion. Crete. Greece. 2004.
5. *Zarrouati N., Aldea E., Rouchon P.* SO(3)-invariant Asymptotic Observers for Dense Depth Field Estimation Based on Visual Data and Known Camera Motion. arXiv preprint arXiv:1103.2539. 2011.
6. *Граничин О.Н.* Процедура стохастической аппроксимации с возмущением на входе // Автоматика и телемеханика. 1992. # 2. С. 97–104.
7. *Han M., Kanade T.* Reconstruction of a scene with multiple linearly moving objects // In: Proc. of IEEE Conference on the Computer Vision and Pattern Recognition. 2000. P. 542–549.
8. *Sturm P.* Structure and motion for dynamic scenes the case of points moving in planes // In: Proc. of the European Conference on Computer Vision. 2002. P. 507–509.
9. *Граничин О.Н.* Об одной стохастической рекуррентной процедуре при зависимых помехах в наблюдении, использующей на входе пробные возмущения // Вестник Ленингр. ун-та. Сер. 1. 1989. Вып.1(4). С. 19–21.
10. *Spall J.C.* Multivariate stochastic approximation using a simultaneous perturbation gradient approximation // IEEE Trans. Automat. Contr. 1992. 37(3). P. 332–341.
11. *Поляк Б.Т., Цыбаков А.Б.* Оптимальные порядки точности поисковых алгоритмов стохастической аппроксимации // Проблемы передачи информации. 1990. #2. С. 45–53.
12. *Granichin O., Gurevich L., Vakhitov A.* Discrete-time minimum tracking based on stochastic approximation algorithm with randomized differences // In: Proc. of the Combined 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf. Shanghai. P.R. China. 2009. P. 5763–5767.

СИСТЕМА ОПРЕДЕЛЕНИЯ ЗАДАННЫХ ЭЛЕМЕНТОВ НА АЭРОФОТОСНИМКАХ

Белова Е.В.

*Бакалавр информационных технологий, ПМ-ПУ, СПбГУ,
ebelovav@gmail.com*

Аннотация: В этой статье рассматривается задача поиска изменений на местности на изображениях, сделанных беспилотным летательным аппаратом (БПЛА). Для нахождения новых объектов на местности применяется метод анализа динамики изменения контуров объектов. Для нахождения заданных объектов производится фильтрация по шаблону.

Помимо вышеобозначенного метода, планируется опробовать методы SIFT, SURF и их различные модификации. Так же, будет введение в объектно-независимый подход к структурному анализу изображений, на котором строится метод анализа динамики изменения контуров объектов.

Ключевые слова: беспилотные летательные аппараты (БПЛА), контуры объектов, объектно-независимый подход к структурному анализу изображений.

I. Введение

В настоящем активно развивается отрасль беспилотных летательных аппаратов (БПЛА), роботов, что позволяет производить исследование той местности и при тех условиях, при которых человеку сделать это трудно. Важными характеристиками БПЛА являются: дальность полёта, высота полёта над землёй, время полёта, полёт при разных погодных условиях и при разных условиях освещения, передача данных мониторинга на базовую станцию (БС) в режиме реального времени, стоимость аппарата.

Активно ведутся разработки ПО для БПЛА. Важное место в разработке ПО для управления БПЛА занимает вопрос синхронизации, обход препятствий, зависание в точке над землёй. Построение мультиагентной системы, которая обеспечивает «общение» между БПЛА необходима для лучшего мониторинга и передачи данных.

В данной статье речь идёт о БПЛА «РАПРИКА» — самолет типа планер с размахом крыла немногим более двух метров. Съёмное крыло можно разделить на две части, благодаря чему довольно большая модель не занимает много места при хранении и транспортировке. Взлет модели осуществляется с руки или с катапульты. Посадка, для которой не потребуются большая площадка, осуществляется пилотом, который перехватывает управление в зоне его видимости. Для увеличения маневренности и уменьшения скорости полета (около 40 км/ч в стигль) закрылки могут работать совместно с элеронами, что позволит изменить профиль всей рабочей плоскости крыла.

В области анализа изображений активно используются такие методы, как SIFT, SURF и их модификации. Минус этих методов в том, что при сильных изменениях микроструктур (изменение освещённости, погоды) они не будут хорошо работать.

Статья состоит из следующих глав. В Главе 2 будет описана общая постановка задачи и текущая подзадача. В главе 3 будет описан метод анализа динамики изменения контуров объектов и фильтрация объектов по шаблону. В Главе 4 будет вкратце описан объектно-независимый подход к структурному анализу изображений. Наконец, в заключительной Главе 5 будет рассказано о планах на будущее.

II. Постановка задачи

Во время мониторинга местности БПЛА, снимки передаются на клиентский компьютер (базовая станция, может быть планшет, ноутбук). Когда мониторинг закончен и снимки все переданы, полученные файлы обрабатываются приложением.

Приложение должно сопоставлять изображения разных выборок изображений (разные мониторинги) и выдавать, как результат работы, те изображения, которые не прошли сопоставление.

Ниже перечислены требования к реализации приложения:

1. Иметь интерфейс, где была бы возможность задавать параметры поиска нужного объекта на изображении или изображения в целом. То есть некий шаблон — паттерн.
2. Распознавать этот объект или изображение.
3. Отсекать дубли, смазанные снимки.
4. Распознавать как изображение одного и того же объекта, в случае, когда объект снят под разным углом.
5. Выдавать изображение пользователю, выделив искомый объект или всё изображение.

A. На данный момент решается подзадача:

- сопоставление серий снимков, снятых в разное время (летом, зимой), то есть набор изображений разных мониторингов.

B. Программно-аппаратный комплекс

Основные параметры БПЛА:

- длина: 1,2 м
- размах крыла: 2 м
- площадь крыла: 35,8 дм²
- максимальный полетный вес: 2 000–2 100 г
- полезная нагрузка: 600 г
- время полета: до 3 ч
- высота полета: 1 000 м

Язык программирования и операционная система:

- Операционная система — Linux Gentoo (ядро 3.7)
- Язык программирования — Python 2.7

Под ОС Linux Gentoo я работаю давно, а язык программирования был выбран такой, поскольку он удобен для прототипирования. В дальнейшем, при необходимости, готовую версию приложения планируется переписать на C++, чтобы увеличить быстродействие.

III. Метод анализа динамики изменения контуров объектов и фильтрация отличий изображений по заданному шаблону

C. Метод анализа динамики изменения контуров объектов

Для нахождения новых объектов на местности применяется метод анализа динамики изменения контуров объектов. Алгоритм в себя включает следующее:

- Выделение контуров на каждом кадре серии
- Сопоставление контуров на соседних кадрах
- Вычисление характеристик изменения контура в серии (для каждого контура)
- Сопоставление контуров с одинаковыми характеристиками на разных сериях

D. Фильтрация отличий изображений по заданному шаблону

Для нахождения заданных объектов производится фильтрация по шаблону. Фильтрация строится по следующему принципам:

- Разбиение шаблона (набор структурных элементов) на структурные элементы
- Разбиение отличающихся частей изображения на структурные элементы
- Поиск совпадений с шаблоном

Алгоритм метода анализа динамики изменения контуров объектов и фильтрации отличий изображений по заданному шаблону базируется на объектно-независимом подходе к структурному анализу изображений.

IV. Объектно-независимый подход к структурному анализу изображений

В основе анализа изображения лежит объектно-независимый подход, а в рамках этого подхода разработан объектно-независимый ортогональный базис обобщённых структурированных элементов, пригодный для структурной декомпозиции изображения на разных иерархических уровнях: от

макрообъектов трёхмерных сцен до текстур. На основе этого подхода разработан объектно-независимый иерархический способ структурного описания и сопоставления изображений, базирующийся на предложенных обобщённых структурных элементах.

Объектно-независимые структурные описания (СО) изображений, устойчивые к естественным изменениям условий наблюдения, целесообразно строить на основе фрагментов границ поверхностей объектов (ПО), локально аппроксимированных прямолинейными отрезками контуров прямых линий (углами, разветвлениями различной формы и перекрестиями линий). Тогда нижний иерархический уровень СО изображений будет строиться на основе неприводимых структурных элементов (НСЭ), пример формы которых для одной из возможных их пространственных ориентаций представлен на рис. 1а и описывается формулой:

$$f(x,y) = 2kx, \quad (1)$$

где $f(x,y)$ — функция декартовых координат x и y , а k — константа.

Это обобщённая формула участка изображения, характеризующегося большим градиентом какого-либо признака. Для обнаружения НСЭ такого типа в системах компьютерного зрения (СКЗ) используются обобщённые эталонные функции (ОЭФ), имеющие аналогичную форму (фильтры Робертса, Собеля, Превитт и т. п.).

Например, рис. 1а описывает форму горизонтального фильтра Превитт, рис. 1б — описывает фильтр Собеля.

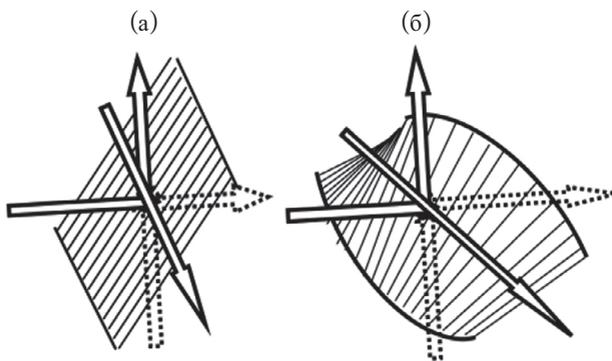


Рис. 1. Графические иллюстрации формы фильтров и соответствующих им неприводимых структурных элементов первого типа, использующихся для выделения границ объектов: горизонтальные фильтры Превитт — (а) и Собеля — (б)

Пространственный спектр такого обобщённого НСЭ и соответствующих ему детекторов имеет нулевую ширину в направлении, параллельном выде-

ляемой границе (или близкую к нулю ширине, например, для горизонтального фильтра Собеля). Это позволяет игнорировать несущественные вариации формы анализируемой границы.

Назовём такой обобщённый НСЭ — обобщённым НСЭ первого типа. Предполагается, что НСЭ первого типа должны служить составными частями пространственно более протяжённых объектов, так как ориентированы на описание фрагментов их границ.

$$f(x,y) = k(x^2 + y^2) + C, \tag{2}$$

где k и C — константы. Это параболоид вращения (рис. 2в)

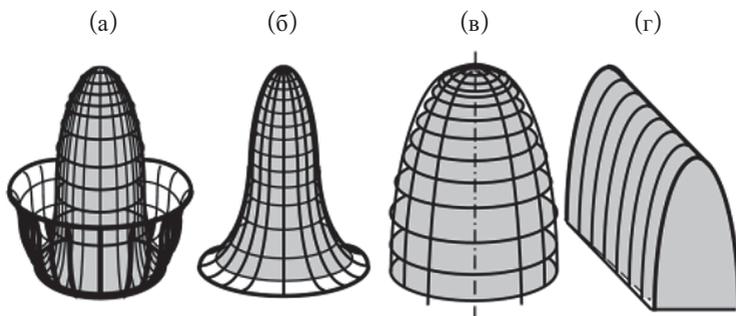


Рис. 2. Узкополосные обобщённые эталонные функции: а — рецептивное поле в форме «мексиканской шляпы»; б — функция информативности, управляющая саккадическими движениями глаз; в — непронизвольный структурный элемент второго типа, описываемый параболоидом вращения; г — непронизвольный структурный элемент второго типа, имеющий вид параболического цилиндра и используемый для определения положения центра тяжести по одной координате

Построенная таким способом ОЭФ, имеющая ширину полосы пространственного спектра, близкую к нулю в любом направлении, может в отличии от НСЭ первого типа использоваться не как детектор границ протяжённых объектов, а в качестве весовой функции детектора пространственно более компактных объектов произвольной формы. Описываемые такой ОЭФ объектно-независимые обобщённые НСЭ назовём элементами второго типа.

Анализ изображений с проективными преобразованиями (ПП) с помощью ОЭФ второго типа позволяет оценивать изменение наклона поверхности по ПП положения и формы элементов её текстуры, аналогично тому, как это делается в человеческой зрительной системе (ЗС), что ещё раз подтверждает общность применяемого подхода и правильность сделанного выбора формы НСЭ.

ОЭФ, представленные на (1) и (2), и соответствующие им НСЭ имеют одно важное свойство. ОЭФ (2), предназначенная для определения двух де-

картовых координат точки центра тяжести (ЦТ) объекта, была синтезирована на основе функции

$$f(x,y)=kx^2+C, \quad (3)$$

обеспечивающей получение аналитического выражения для положения ЦТ вдоль единственной декартовой координаты. Это — функция параболического цилиндра, показанная на рис. 2г.

Таким образом, функции, представленные рис. 1а и 2г (горизонтальный фильтр Превитт и непроизводный структурный элемент второго типа, имеющий вид параболического цилиндра и используемый для определения положения центра тяжести по одной координате соответственно) и функции, представленные на рис. 1б (горизонтальный фильтр Собеля) и рис. 2в (непроизводный структурный элемент второго типа, описываемый параболоидом вращения), образуют ортогональный базис обобщённых НСЭ для объектно-независимой структурной декомпозиции изображений.

Известны и другие ортогональные базисы НСЭ, например, базис Робертса, но для них не гарантируется устойчивость к аффинным преобразованиям (АП) и ПП, а состав таких базисов избыточен с точки зрения предлагаемой стратегии структурной декомпозиции текстур и форм макрообъектов, заимствованной в живых ЗС.

V. Заключение

Итак, в статье был представлен, разработанный мною и находящейся в процессе реализации *метод анализа динамики изменения контуров объектов*. Так же, было рассказано об объектно-независимом подходе к структурному анализу изображений и приведены примеры разных методов анализа изображений с кратким описанием их сути.

На данном этапе исследования реализуется подзадача общей задачи поиска заданных объектов — сопоставление серий снимков, снятых в разное время (летом, зимой), то есть набор изображений разных мониторингов. Подзадача решает вопрос с анализом динамики изменений изображений и поиска отличий, путём сопоставления снимков.

В заключение, важно отметить, что область применения приложения имеет практическое применение. Так, летом планируется его опробовать на мониторинге местности заповедника для поиска новых объектов на территории.

В будущем, планируется доработать алгоритм — решить другие обозначенные задачи. Помимо вышеобозначенного метода, планируется опро-

бовать методы SIFT, SURF и их различные модификации. Так же, будет введение в объектно-независимый подход к структурному анализу изображений, на котором строится метод анализа динамики изменения контуров объектов.

Литература

1. *В.Р. Луцив* Объектно-независимый подход к структурному анализу изображений. 2008. С. 28–33.
 2. *К.С. Амелин*, «Беспилотный летательный аппарат для работы в группе (БПЛА-агент)»
-

МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ В ВЕРОЯТНОСТНОЙ МАШИНЕ ТЬЮРИНГА

А.И. Кукушкин

*аспирант кафедры компьютерных систем и программных технологий,
polladin@rambler.ru*

Е.Н. Бендерская

*к. т. н., доц. кафедры компьютерных систем и программных технологий, helen.
bend@gmail.com*

Санкт-Петербургский государственный политехнический университет

Аннотация: В данной работе исследуется один из подходов включения в классические вычисления базовой идеи квантовых вычислений посредством моделирования вероятностей переходов и взаимосвязей между разными вариантами развития стохастических процессов.

Введение

Основная идея, положенная в основу данной работы, заключается в том, что комбинирование хаотических процессов и конечной динамики системы позволяет генерировать цепи Маркова с конечным числом состояний [1]. При этом соответствующий стохастический процесс описывается двумя типами уравнений: первое принадлежит к моделированию случайных процессов, второе описывает развитие вероятностей. Объединение этих двух типов уравнений дает возможность смоделировать варианты развития вероятностей похожие на развитие вероятностей в квантовой механике.

Рассматриваемый подход представляет интерес в свете разработки новых интеллектуальных систем, так как большинство процессов ассоциирующихся с интеллектуальной деятельностью, таких как, поиск оптимального поведения, процесс принятия решения, представляют собой определённую вероятностную структуру, которая может быть описана с помощью цепей Маркова и представлена с помощью вероятностной машины Тьюринга.

Математическое описание вероятностной машины Тьюринга

Для описания вероятностной машины Тьюринга используется система уравнений [1]:

$$\dot{v} = \nu v^{1/2} \sin \omega t + \varepsilon_0 (\tilde{y} - x), \quad \dot{x} = v \quad (1)$$

$$\dot{v}_n = \nu_n v_n^{1/2} \sin \omega t + \varepsilon_0 [\tilde{y} - \varphi(n, x)], \quad \dot{n}_* = v_n, \quad n = N - n_* > 0 \quad (2)$$

$$\tilde{y} = y(t) - 0,51 \tag{3}$$

$$y(t + \tau) = 4y(t)[1 - y(t)], \quad t = \frac{\pi}{2\omega}n \tag{4}$$

$$y(t^*) = 0,2, \quad -\frac{\pi}{4\omega} < t^* < \frac{\pi}{4\omega}n \tag{5}$$

Временной ряд описываемый формулой (4) сгенерирован с помощью полностью детерминированной (но хаотической) динамической системой. Простейшая из таких систем — это система на основе логистической функции, которая играет ключевую роль в популяционной динамике, химических процессах и других областях.

Уравнения (1, 2) моделируют динамику вероятностной машины Тьюринга. В этих уравнениях присутствует $y(t)$, который представляет собой источник стохастических временных рядов, и не вносит изменение в решение уравнения, т. к. $\xi_0 \ll 1$, а знак при $t=0, \pi/2\omega, \dots$ и т. д. выбирает одно из двух возможных решений.

$$v = \pm \left(\frac{4v}{3\omega} \sin^2 \frac{\omega}{2} t \right)^{3/2} \tag{6}$$

Таким образом, исходная динамическая система преобразуется в стохастический процесс, который характеризуется следующими распределениями плотности вероятности [1]:

$$f_n(n, x, t + \tau) = p_n f_n(n - 1, x, t) + (1 - p_n) f_n(n + 1, x, t) \tag{7}$$

$$\sum_{n=1}^N f_n(n, x, t) = 1 \tag{8}$$

$$f(x, t + \tau) = \sum_{n=1}^N f_n [p f(x - nh, t) + (1 - p) f(x + nh, t)] \tag{9}$$

$$\int_{-\infty}^{\infty} f(x, t) dx = 0 \tag{10}$$

$$p = \begin{cases} 0 & \text{if } x > y_{\max} \\ \frac{y_{\max} - x}{y_{\max} - y_{\min}} & \text{if } y_{\min} \leq x \leq y_{\max} \\ 1 & \text{if } x < y_{\min} \end{cases} \tag{11}$$

$$p = \begin{cases} 0 & \text{if } \varphi(x, n) > y_{\max} \\ \frac{y_{\max} - \varphi(x, n)}{y_{\max} - y_{\min}} & \text{if } y_{\min} \leq \varphi(x, n) \leq y_{\max} \\ 1 & \text{if } \varphi(x, n) < y_{\min} \end{cases} \quad (12)$$

Следовательно, можно вычислить матрицу вероятностей перехода к следующему состоянию:

$$P = \begin{pmatrix} p_{11} & \dots & p_{1N} \\ \dots & \dots & \dots \\ p_{N1} & \dots & p_{NN} \end{pmatrix} \quad (13)$$

$$p_{ij} = pf_n(x_i, n_j)$$

$$P^* = \begin{pmatrix} p_{11}^* & \dots & p_{1N}^* \\ \dots & \dots & \dots \\ p_{N1}^* & \dots & p_{NN}^* \end{pmatrix} \quad (14)$$

$$p_{ij}^* = (1-p)f_n(x_i, n_j)$$

Исходя из этих матриц можно построить цепь Маркова:

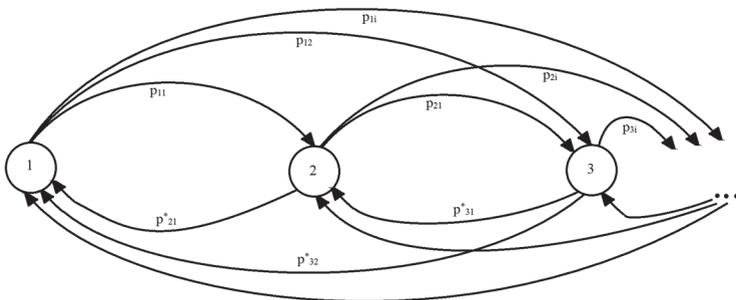


Рис. 1. Цепь Маркова построенная по матрицам (13, 14)

Тогда можем сказать, что вероятностная машина Тьюринга, описываемая с помощью уравнений (1–5) моделирует стохастический процесс, который описан с помощью цепи Маркова изображённой на рис. 1. и стохастическая матрица которой описана с помощью матриц (13, 14).

Моделирование вероятностной машины Тьюринга

На рис. 2 показана динамика стохастического процесса, описанного цепью Маркова (13-14), моделируемого на вероятностной машине Тьюринга, описанной уравнениями (1–5).

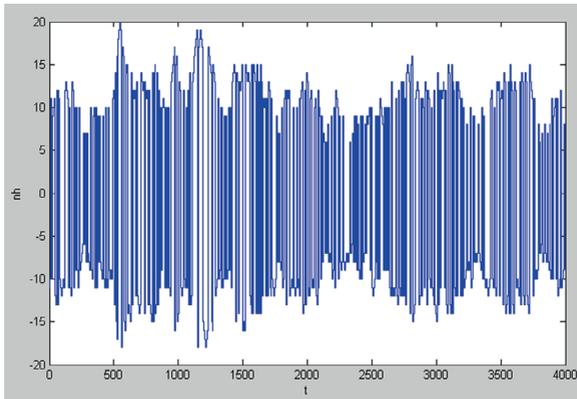


Рис. 2. Динамика стохастического процесса моделируемого на вероятностной машине Тьюринга

Моделирование интеллектуальных систем

Модифицируя уравнения (1–5) вероятностной машины Тьюринга [1] можно получить следующие соотношения:

$$\dot{v} = v v^{1/2} \sin \omega t + \varepsilon_0 (\tilde{y} - \varphi(x, H, \sigma, \dots)), \dot{x} = v \quad (15)$$

$$\dot{v}_n = v_n v_n^{1/2} \sin \omega t + \varepsilon_0 [\tilde{y} - \varphi_i], \dot{x}_i = v_i, \quad i = 1, 2, \dots, n \quad (16)$$

$$\varphi_i = \tanh \sum_{j=1}^n T_{ij} x_j \quad (17)$$

Рассматривая эти уравнения в рамках теории нейронных сетей, уравнение (15) будет уравнением одного нейрона, из которого можно составить сеть описываемую уравнениями (16), а связи между нейронами будут описываться уравнениями (17).

В уравнении (17) T_{ij} не зависит от x_j . Такая динамическая система, в терминах развития вероятностей, уникально определяется выбором констант T_{ij} .

Основная идея использования такой системы в том, что информация может отсутствовать на нижнем уровне системы, но существовать на уровне

коллективной активности. При этом происходит самоорганизация и взаимодействие между основными компонентами без глобального контроля.

Моделирование сети из двух нейронов

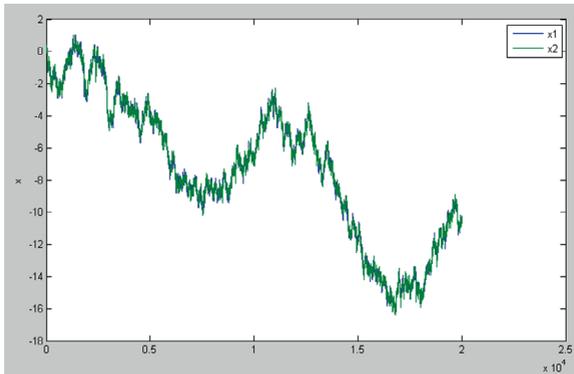


Рис. 3. Выход X сети из двух основных элементов при $T = [1 \ -1; \ -1 \ 1]$

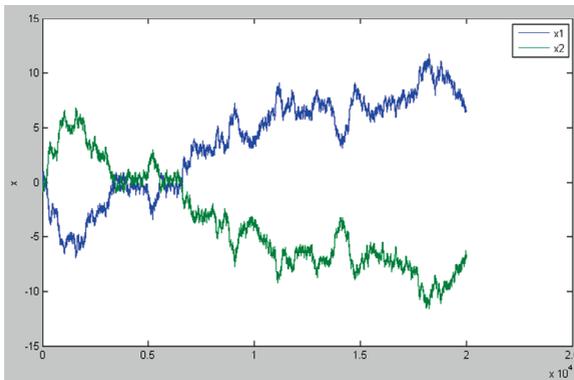


Рис. 4. Выход X сети из двух основных элементов при $T = [1 \ 1; \ 1 \ 1]$

Заключение

Целью данной работы было исследование возможности разработки вычислителя, который моделирует стохастические процессы, происходящие в интеллектуальных системах. В качестве такого вычислителя была рассмотрена вероятностная машина Тьюринга, которая может моделировать вероятностную структуру, описанную с помощью цепи Марков, которая в свою

очередь может описывать большинство процессов связанных с интеллектуальной деятельностью.

При модифицировании вероятностной машины Тьюринга, может быть получена интеллектуальная система, которая работает по принципу самоорганизации. Такая система может быть описана в терминах нейронных сетей. Таким образом, предполагается создание нейронной сети, где активность нейрона может быть смоделирована с помощью вероятностной машины Тьюринга.

Литература

1. *Zak M.* From instability to intelligence : complexity and predictability in nonlinear dynamics. Springer, Tokyo, 1997. 552 p.
 2. *Richard P. Feynman.* Simulating Physics with Computers. Department of Physics, California Institute of Technology, Pasadena, California 91107. Received May 7, 1981. 22 p.
 3. *Zak M.* Introduction to Quantum Intelligence. Center for Space Microelectronics Technology Jet Propulsion Laboratory California Institute of Technology Pasadena, CA 91109. 1996. 60 p.
 4. *Daniel Pierre Bovet, Pierluigi Crescenzi.* Introduction to the theory of complexity. June 2006. 282 p.
-

СРАВНЕНИЕ ПОДХОДОВ ОБУЧЕНИЯ НА БАЗЕ СЛОВАРЯ И МАР К ПРОБЛЕМЕ ПОВЫШЕНИЯ РАЗРЕШЕНИЯ НА ПРИМЕРЕ ИЗОБРАЖЕНИЙ АВТОМОБИЛЬНЫХ НОМЕРОВ

Улитин. А.А.

*студент кафедры системного программирования СПбГУ,
alexander.a.ulitin@gmail.com*

Вахитов. А.Т.

*доцент кафедры системного программирования СПбГУ,
alexander.vakhitov@gmail.com*

Аннотация: В докладе дано сравнение двух известных подходов к проблеме повышения разрешения: на базе обучения словаря и maximum a posteriori оценивания с использованием некой заранее заданной функции плотности вероятности. Сравнение будет делаться на примере разработанной собственными силами базы данных автомобильных номеров.

Введение

Цифровое изображение имеет конечное количество пикселей. Количество этих пикселей деленное на единицу площади называется разрешением цифрового изображения.

Для анализа и обработки изображений в большинстве случаев используются изображения высокого разрешения, так как они позволяют увидеть детали, которые не различимы, или плохо различимы на изображениях с низким разрешением. Многие задачи компьютерного зрения изначально полагаются на то, что изображение на входе в хорошем разрешении. С первого взгляда, задача увеличения разрешения чисто аппаратная – чтобы увеличить разрешение необходимо просто взять фотоаппарат с большей разрешающей способностью. Но часто уже имеется некоторая фотография или набор фотографий, и нет возможности повторно произвести съемку. Примером может послужить низкокачественные картинки с изображений камер наблюдения. В этом случае уместно использовать программные алгоритмы повышения разрешения. Эти алгоритмы используют некоторое дополнительное знание о изображении, для того, чтобы качественно повысить разрешение.

В этой работе рассмотрены два алгоритма для повышения разрешения на примере автомобильных номеров.

Постановка задачи

Для рассуждений необходимо некоторым образом формально записать то, с чем имеем дело.

Метрика PSNR

Для того, чтобы сравнивать несколько алгоритмов повышения разрешения необходимо ввести какую-нибудь количественную метрику. Чтобы иметь возможность посчитать метрику необходимо знать эталонное изображение, то есть такое изображение, которое должен выдать идеальный алгоритм повышения разрешения. Очевидно, что создать эталонный алгоритм невозможно — это отображение из $N^{n \cdot m} \rightarrow N^{nk \cdot mk}$, $k \geq 2$. Знание эталонного изображения несколько выходит за рамки поставленной во введении задачи — в реальных условиях истинное изображение будет неизвестно. Для тестирования алгоритмов были использована стандартная модель получения изображений низкого разрешения из высокого.

Пусть x — истинное изображение, \tilde{x} — найденное изображение из изображений низкого разрешения. Посчитаем среднеквадратичную ошибку между пикселями этих изображений.

$$\text{MSE}(\tilde{x}, x) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [\tilde{x}(i, j) - x(i, j)]^2$$

И выразим отношение MAX_I — максимальное значение яркости изображения к среднеквадратичной ошибке в децибелах $\text{PSNR}(\tilde{x}, x)$.

$$\text{PSNR}(\tilde{x}, x) = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}(\tilde{x}, x)} \right)$$

Это и будет нашей метрикой.

Интерполяция

Одним из наиболее известных способов повышения разрешения является интерполяция. Для любого изображения можно бесконечно повышать разрешения, просто добавляя дополнительные значения между пикселями исходного изображения. Однако, такое повышение разрешения не всегда дает хорошие результаты. Поскольку значение PSNR лишь показывает разницу между двумя изображениями, то нет возможности численно оценить насколько алгоритм Super-resolution хорошо справился с задачей. Но если сравнивать этот же алгоритм со значением полученные каким-либо методом интерполяции то наглядно видно, где алгоритм справляется хорошо, а где плохо. Более подробно про интерполяцию [1] [2]

Изображения как векторы

Для возможности записывать все операции над изображением как матричное умножение, будем считать, что изображение высокого разрешения,

которое мы хотим восстановить x размера $L_1N_1 \times L_2N_2$ записано в виде вектора $x = [x_1, x_2, \dots, x_N]^T$, где $N = L_1N_1 \times L_2N_2$.

Задача

$$y_r = DH_r W_r x + \sigma_k \epsilon, \quad 1 \leq r \leq m,$$

где:

- x — оригинальное изображение
- y_r — наблюдение r
- D — матрица понижение разрешения
- W_r — матрица геометрического искажения
- H_r — матрица размытия наблюдения r
- $\sigma_k \epsilon$ — шум наблюдения r
- m — количество наблюдений

Задача найти:

$$\tilde{x} = \operatorname{argmax} PSNR(\hat{x}, x)$$

Используемые алгоритмы

Для того, чтобы повысить разрешение автомобильного номера использовались два подхода: обучаемый [3] и интерполяционный с использованием регуляризации [4]. Этот выбор основывался исходя из природы изображения автомобильного номера.

Обучаемый алгоритм на словарях

В статье [3] авторы предлагают использовать для задачи повышения разрешения одного изображения использовать пару связанных словарей D_x и D_y — словари содержащие патчи из пространства изображений с высоким и низким разрешением.

Алгоритм требует предварительного обучения нейронной сети. Для этого была создана база с изображениями автомобильных номеров. Используя авторскую реализацию алгоритма была обучена пара словарей для восстановления изображений.

Обоснованием, для применения этого метода были следующие:

- Автомобильные номера содержат конечный набор патчей. Автомобильные номера очень похожи между собой. И явно видно, что между двумя изображениями автомобильных номеров есть общие кусочки изображения.

- Автомобильные номера имеют простую структуру — у них нет сложных переходов, большинство смены цветов это просто переход от белого к черному.

Алгоритм с использованием регуляризации

В статье [4] авторы предлагают улучшение алгоритма MAP для задачи повышение изображения. Изображение моделируется как Марковская сеть и используется алгоритм шаговой оптимизации в сочетании с адаптивным регуляризатором для сохранения границ.

Общий вид алгоритма следующий:

1. Посчитать среднее от выровненных и билинейно интерполированных изображений $X^{(0)}$. Будем называть полученные на этом шаге изображение «начальным приближением»
2. Положить $\gamma = 2v$, где v — максимальное значение градиента по осям x и y в начальном приближении $X^{(0)}$
3. До:
 1. $X^{(n+1)} = X^n - \alpha \cdot \text{grad}(X^{(n)}, \gamma)$
 2. $n = n + 1$
 3. If $(\text{norm}(X^{(n)} - X^{(n-1)}) < \epsilon)$
 then $\gamma^n = \max\{\gamma_{\text{target}}, k\gamma^{(n-1)}\}$
4. $\hat{x} = X^n$

Until $(\text{norm}(X^{(n)} - X^{(n-1)}) < \epsilon)$ and $\gamma^n = \gamma_{\text{target}}$

Где $\alpha, \epsilon, k, \gamma_{\text{target}}$ параметры алгоритма. В нашей реализации мы использовали $\alpha = 0.07, \epsilon = 0.005, k = 0.95, \gamma_{\text{target}} = 10$

$$\text{grad}(x, \gamma) = \frac{1}{\sigma^2} \sum_{r=1}^m W_R^T H_r^T D^T (DH_r W_r x - y_r) + \lambda \cdot G(x, \gamma)$$

Где λ экспериментально подобранные регуляризационный параметр и Градиент для точки (i, j) задан следующей формулой:

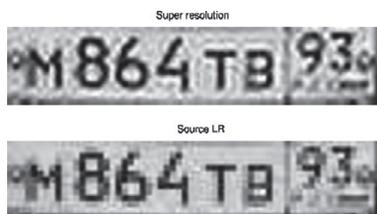
$$\begin{aligned} G(i, j) = & 2[x(i, j) - x(i, j - 1)] \exp(-[x(i, j) - x(i, j - 1)]^2 / \gamma) \\ & + 2[x(i, j) - x(i, j + 1)] \exp(-[x(i, j) - x(i, j + 1)]^2 / \gamma) \\ & + 2[x(i, j) - x(i + 1, j)] \exp(-[x(i, j) - x(i + 1, j)]^2 / \gamma) \\ & + 2[x(i, j) - x(i - 1, j)] \exp(-[x(i, j) - x(i - 1, j)]^2 / \gamma) \end{aligned}$$

Результаты

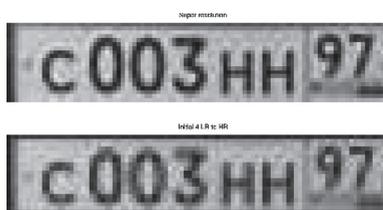
Для сравнения качества алгоритмов были выбраны несколько изображений из обучающей базы для тренировки словаря у первого алгоритма. Стоит заметить, что эти изображения были убраны из тренировочного набора.



Рис. 1. Исходные изображения



Пример результата работы алгоритма с тренированными словарями



Пример результата работы алгоритма с использованием регуляризации

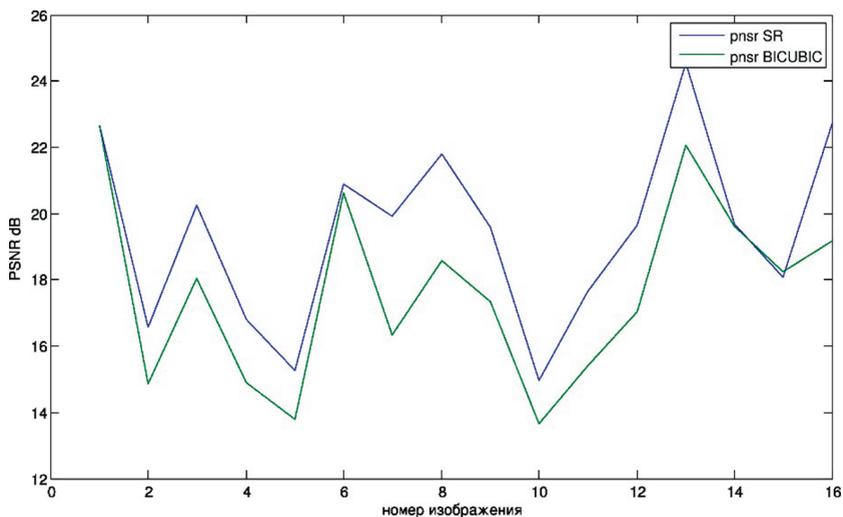


Рис. 2. Результаты алгоритма с тренированными словарями. На оси абсцисс отмечен номер изображения, из тестового набора

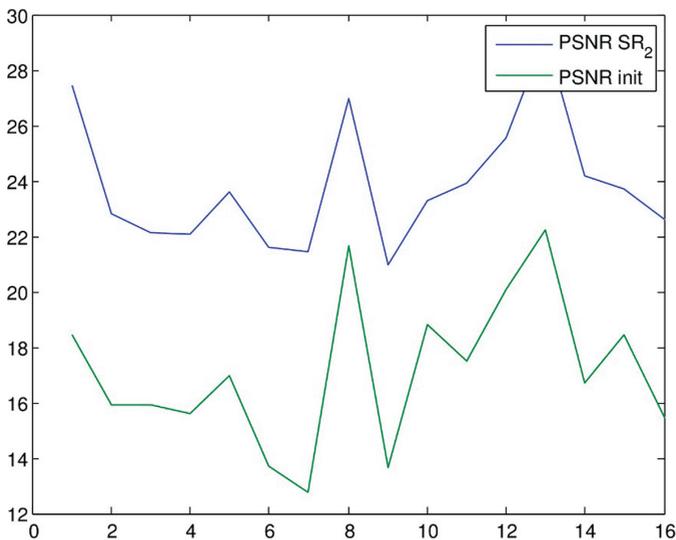


Рис. 3. Результаты алгоритма с использованием регуляризации

Вывод

Как результат работы, можно заключить, что оба метода справляются с задачей повышения разрешения лучше, чем наивные интерполяционные подходы. Однако, результаты метода с использованием MAP намного лучше, чем результаты первого. Это можно объяснить тем, что второй алгоритм использует регуляризатор для выделения границ, что для большинства изображений будет давать не самые правильные результаты, но в случае автомобильных номеров это работает.

Полные результаты сравнения, реализации методов, словари для обучения и полные исходные коды на MatLab-е находятся в Git репозитории по адресу <https://github.com/ulex/sr-license-plates>

Литература

1. *R.C. Gonzalez и E. Richard*, «Woods, digital image processing,» ed: Prentice Hall Press, ISBN 0-201-18075-8, 2002.
2. *R. Keys*, «Cubic convolution interpolation for digital image processing,» Acoustics, Speech and Signal Processing, IEEE Transactions on, т. 29, №6, pp. 1153–1160, 1981.
3. *J. Yang, Z. Wang, Z. Lin, S. Cohen и T. Huang*, «Coupled dictionary training for image super-resolution,» Image Processing, IEEE Transactions on, т. 21, №8, pp. 3467-3478, 2012.

4. *K. Suresh, G.M. Kumar и A. Rajagopalan*, «Superresolution of license plates in real traffic videos,» *Intelligent Transportation Systems, IEEE Transactions on*, т. 8, №2, pp. 321-331, 2007.
 5. *P.M.K.K.M.G. Park S.C.*, «Super-resolution image reconstruction: a technical overview,» *IEEE Signal Processing Magazine*, т. 20, pp. 21–36, 2003.
 6. *J. Tian и К.-К. Ма*, «A survey on super-resolution imaging,» *Signal, Image and Video Processing*, т. 5, № 3, pp. 329–342, 2011.
-

Параллельные алгоритмы и вэйвлетная обработка числовых потоков



**Демьянович
Юрий Казимирович**

д.ф.-м.н., профессор
заведующий кафедрой параллельных алгоритмов СПбГУ

О КУРСЕ ЛЕКЦИЙ «АРХИТЕКТУРА ПАРАЛЛЕЛЬНЫХ СИСТЕМ»

Евдокимова Т.О.

доцент кафедры параллельных алгоритмов СПбГУ, m9beto@math.spbu.ru

Аннотация: Данная статья посвящена обзору готовящегося к печати учебного пособия «Архитектура параллельных систем», основанного на материале курса лекций, читающегося на математико-механическом факультете СПбГУ.

В работе анализируются содержание и структура пособия, их связь с целями обучения, а также принципы отбора материала.

Введение

Специальный курс «Архитектура параллельных систем» читается на 3-м курсе подготовки специалистов по специальности «Математическое обеспечение и администрирование информационных систем». Такое расположение курса в учебном плане накладывает на содержание, структуру и методы изложения материала в учебном пособии определенные требования.

Помимо имеющихся у студентов к 3-му курсу знаний и умений с одной стороны и целей изучения дисциплины с другой, при подборе и представлении материала необходимо понимать и учитывать следующие моменты:

- постоянное изменение (расширение и усложнение) предметной области, связанное с непрерывным техническим совершенствованием вычислительных систем;
- уровень изложения (степень абстракции), ограниченный сверху уровнем абстракции в развитии самой науки;
- структуру учебной деятельности студентов.

Таким образом, сформулированные в целях и результатах курса известные показатели — уровень усвоения материала, автоматизация усвоения, степень осознанности усвоения, — дополняются указанными тремя менее формализуемыми. И все вместе они определяют объём курса.

Структура и содержание пособия

Курс лекций «Архитектура параллельных систем» состоит из введения и 24 разделов, дополненных предметным указателем и списком основной и дополнительной литературы.

Материал может быть формально разделён на три модуля, освещающих следующие темы:

1. архитектура памяти вычислительных систем;

2. коммуникационные среды как элемент архитектуры параллельной системы (рассмотрены среды SCI, MYRINET, Raceway, Memory Channel, среды на транспьютероподобных микропроцессорах);
3. оборудование и структура параллельных систем (сюда можно отнести разделы, посвященные коммутаторам, параллельной системе МВС-100/1000, нейросетевым вычислительным системам, классификации параллельных систем, их надёжности и проводительности, параллельным и распределённым базам данных, реализации ассоциативных алгоритмов, векторно-конвейерным компьютерам, обзору концепций создания параллельных систем, систем с распределённой памятью и концепции GRID, а также раздел о распараллеливании метода Гаусса решения СЛАУ).

Цели и результаты курса

Сформулируем указанные в учебном плане цели и результаты курса в терминах показателей, использующихся для характеристики учебных пособий.

1. Начальная подготовка к изучению предмета и навыки самостоятельной учебной работы требуют от слушателей владения основами численных методов и основами программирования, достаточными для составления простейших программ (в объеме двух первых курсов обучения). Следовательно, мы имеем дело с в некотором роде направленным информационным процессом, то есть процессом, учитывающим познавательные особенности учащихся. Учет индивидуального темпа учебной деятельности учащегося затруднён при исключительно лекционном построении курса, однако частично возможен при наличии соответствующих семинарских или самостоятельных занятий.
2. Поскольку материал курса является для слушателей новым, то степени абстракции изложения предмета варьируются от первой (феноменологической, то есть осуществляющейся на естественном, житейском языке) через вторую (качественную теорию, использующую сложившуюся терминологию и символику науки, предполагающую возможность формулировки свойств и закономерностей функционирования объектов) до третьей (количественной теории, имеющей достаточные знания свойств объектов для предсказания поведения объектов на основе модели). Надо отметить, что такой подход «снизу вверх» характерен для каждого из разделов курса.
3. Результирующий уровень усвоения содержания (уровень мастерства владения информацией) после окончания курса устанавливается равным исполнительскому, то есть слушатель должен уметь воспроизводить известную информацию и применять алгоритмы для решения типовых задач. Однако, по некоторым разделам курса результирующим может быть установлен как ученический уровень, достаточный для

узнавания и классификации известного материала, так и экспертный, позволяющий ориентироваться в новых ситуациях.

4. Коэффициент автоматизации усвоения материала, то есть насколько свободно и без задержек слушатель должен ориентироваться в материале и задачах курса по сравнению с профессионалом/преподавателем, не предполагается равным 1. По-видимому, он может быть принят за 0,5.
5. Степень осознанности усвоения и деятельности отлична от механистической по тем разделам курса, по которым уровень усвоения материала предполагается выше ученического (например, по разделу, описывающему распараллеливание метода Гаусса) и является либо внутрипредметной, либо межпредметной.

Анализ содержания курса

Значения диагностических показателей

Приведем значения (средние) показателей степени абстракции изложения предмета (Са), уровня усвоения содержания (Уу) и степени осознанности усвоения (Со) в соответствии с распределением материала по указанным выше модулям:

Описание модуля	Значение показателей: Са/Уу/Со
Архитектура памяти вычислительных систем	Качественная / Исполнитель / Внутрипредметная
Коммуникационные среды как элемент архитектуры параллельной системы	Качественная / Исполнитель / Внутрипредметная
Оборудование и структура параллельных систем	Количественная / Экспертная / Межпредметная
Движущаяся камера, стац. точка	11.3551

Таблица 1. Качественные показатели разделов курса

Отбор содержания курса

Вопрос первоначального отбора материала для подобного курса лекций достаточно сложен. Во-первых, необходимо учитывать базовость данного предмета, из которой следует неизбежность первой ступени абстракции в начале изложения основных разделов. С другой стороны, установленные требования к уровню усвоения требуют не только широкого подбора материала, но и глубокого его изложения, то есть качественной и количественной ступени абстракции. Также всё это должно сопровождаться достаточным количеством упражнений для самопроверки и тестов для внешней проверки уровня усвоения и степени осознанности усвоения, предлагающихся слушателям/читателям регулярно, как минимум после каждого раздела курса.

Во-вторых, лекционный курс для студентов не может иметь бесконечный объем и обычно ограничен четырьмя часами теоретических занятий в неделю. Разумеется, при составлении учебника исходный материал дополняется разнообразными моментами: историческими справками и формулировками задач, требующих для решения соответствующих знаний и навыков — для увеличения заинтересованности учащихся, то есть для создания произвольной мотивации учения; дополнительным материалом, выходящим за рамки основного по сложности и/или широте — для поддержания интереса у уже мотивированных и владеющих языком этой ступени абстракции изложения читателей и т. п.

Таким образом, в курсе необходимо излагать базовый, в некотором смысле стандартный материал (например, архитектуры памяти вычислительных систем), освещать распространённые технологические решения (различные коммуникационные среды), описывать как примеры имеющихся параллельных систем (например, различные кластеры), так и задач с помощью них решающихся (задачи линейной алгебры или сводящиеся к ним).

Пути дополнения курса

Представляется очевидным, что полезным и для читателей, и для преподавателей будет дополнение курса лекций сборником упражнений. Будучи составленными в достаточном количестве по каждому разделу курса для определения разных уровней усвоения они позволят читателям, обладающим внутренней мотивацией, производить самоконтроль усвоения материала и уровня автоматизации, а преподавателям — обеспечивать не только направленный информационный процесс, но и, возможно, замкнутое управление учением, особенно если в самом курсе лекций или в сборнике упражнений будут методически выделены учебные элементы курса — содержательно значимые структуры минимального объёма, знание и понимание которых является необходимым.

Заключение

В работе был представлен курс лекций «Архитектура параллельных систем». Проведена классификация частей курса в соответствии со значениями диагностических параметров, придающих целям курса общепонятность и измеримость. Рассмотрены причины трудностей, возникающих при подборе материала курса, и возможные пути их преодоления.

Литература

1. Демьянович Ю.К., Евдокимова Т.О., Чистяков П.П. Архитектура параллельных систем. Учебное пособие. (готовится к печати)
2. Беспалько В.П. Учебник. Теория создания и применения. М.: НИИ школьных технологий, 2006. 192 с.

О ВЕЙВЛЕТНОМ ГРЕБНЕ НА НЕРЕГУЛЯРНОЙ СЕТКЕ

В.О. Дронь

*аспирант, ассистент кафедры параллельных алгоритмов
математико-механического факультета СПбГУ, vera.dron@gmail.com*

Аннотация: Рассматривается сплайн-вэйвлетное разложение в случае двух-интервальной гребенчатой структуры на неравномерной сетке. Даны представления матрицы вложения и матрицы продолжения, а также установлена связь между этими матрицами. Представлены алгоритмы декомпозиции и реконструкции.

Введение

Важную роль в исследовании вэйвлетных разложений числовых информационных потоков играет изучение сплайновых аппроксимаций различных порядков, разложений со свойствами локализации в отдельных частях рассматриваемой области, а также использование неравномерных сеток. Цель данной работы состоит в том, чтобы рассмотреть структуру сплайн-вэйвлетного разложения в случае тесного расположения узлов на неравномерной сетке.

Предварительные сведения

Для любого натурального числа m введем обозначения

$$J_m \stackrel{\text{def}}{=} \{0, 1, \dots, m\}, J'_m \stackrel{\text{def}}{=} \{-1, 0, 1, \dots, m\}.$$

Пусть N — натуральное число. На отрезке $[a, b]$ рассмотрим сетку

$$X : a = x_0 < \dots < x_{N-2} < x_{N-1} < x_N = b, \quad (1)$$

и обозначим

$$S_{-1} \stackrel{\text{def}}{=} (x_0, x_1), S_j \stackrel{\text{def}}{=} (x_j, x_{j+1}) \cup (x_{j+1}, x_{j+2}), j \in J_{N-2},$$

$$S_{N-1} \stackrel{\text{def}}{=} (x_{N-1}, x_N), G \stackrel{\text{def}}{=} \bigcup_{i \in J_{N-1}} (x_i, x_{i+1}).$$

Систему векторов $\{\mathbf{a}_i\}_{i \in J'_m}$ пространства \mathbb{R}^2 будем называть полной цепочкой векторов, если $\det(\mathbf{a}_{j-1}, \mathbf{a}_j) \neq 0$ при $j \in J_m$.

Пусть $A \stackrel{\text{def}}{=} \{\mathbf{a}_i\}_{i \in J'_{N-1}}$ — полная цепочка двумерных векторов.

По двухкомпонентной вектор-функции $\varphi(t)$, $t \in G$, с линейно независимыми компонентами на любом интервале $(a', b') \subset G$, определим функции $\omega_j(t)$, $t \in G$, $j \in J'_{N-1}$ с помощью аппроксимационных соотношений

$$\sum_{j \in J'_{N-1}} \mathbf{a}_j \omega_j(t) = \varphi(t) \quad \forall t \in G, \quad \omega_j(t) \equiv 0 \quad \forall t \in G \setminus S_j, \quad j \in J'_{N-1}.$$

Тогда функции $\omega_j(t)$, $j \in J'_{N-1}$, определяются на множестве G следующим образом:

$$\omega_{-1}(t) = \begin{cases} \frac{\det(\varphi(t), \mathbf{a}_0)}{\det(\mathbf{a}_{-1}, \mathbf{a}_0)} & \text{при } t \in (x_0, x_1), \\ 0 & \text{при } t \in G \setminus S_{-1}, \end{cases}$$

для $j \in J_{N-2}$

$$\omega_j(t) = \begin{cases} \frac{\det(\mathbf{a}_{j-1}, \varphi(t))}{\det(\mathbf{a}_{j-1}, \mathbf{a}_j)} & \text{при } t \in (x_j, x_{j+1}), \\ \frac{\det(\varphi(t), \mathbf{a}_{j+1})}{\det(\mathbf{a}_j, \mathbf{a}_{j+1})} & \text{при } t \in (x_{j+1}, x_{j+2}), \\ 0 & \text{при } t \in G \setminus S_j, \end{cases}$$

$$\omega_{-1}(t) = \begin{cases} \frac{\det(\mathbf{a}_{N-2}, \varphi(t))}{\det(\mathbf{a}_{N-2}, \mathbf{a}_{N-1})} & \text{при } t \in (x_{N-1}, x_N), \\ 0 & \text{при } t \in G \setminus S_{N-1}, \end{cases}$$

Пусть s и r — натуральные числа, и $s < r$. Из сетки (1) удалим группу узлов с нечетными номерами в количестве $r - s$, а именно, удалим узлы

$$x_{2s+1}, x_{2s+3}, x_{2s+5}, \dots, x_{2r-1},$$

так что укрупненная сетка имеет вид

$$\tilde{X} : a = \tilde{x}_0 < \tilde{x}_1 < \tilde{x}_2 < \dots < \tilde{x}_{\tilde{N}-1} < \tilde{x}_{\tilde{N}} = b,$$

где $\tilde{N} = N - r + s$, а

$$\begin{aligned} \tilde{x}_i &= x_i & \text{при} & \quad 0 \leq i \leq 2s, \\ \tilde{x}_{2s+i} &= x_{2s+2i} & \text{при} & \quad 1 \leq i \leq r-s, \\ \tilde{x}_{s+r+i} &= x_{2r+i} & \text{при} & \quad 1 \leq i \leq N-2r. \end{aligned}$$

Положим

$$\tilde{S}_j \stackrel{\text{def}}{=} (\tilde{x}_j, \tilde{x}_{j+1}) \cup (\tilde{x}_{j+1}, \tilde{x}_{j+2}), \quad j \in J'_{\tilde{N}-2}, \quad x \tilde{G} \stackrel{\text{def}}{=} \bigcup_{i \in J'_{\tilde{N}-1}} (\tilde{x}_i, \tilde{x}_{i+1}),$$

$$\tilde{S}_{-1} \stackrel{\text{def}}{=} (\tilde{x}_0, \tilde{x}_1), \quad \tilde{S}'_{\tilde{N}-1} \stackrel{\text{def}}{=} (\tilde{x}_{\tilde{N}-1}, \tilde{x}_{\tilde{N}}).$$

Кроме того, введем обозначения

$$I_*^h \stackrel{\text{def}}{=} \{-1, 0, \dots, 2s-2\}, \quad I_*^m \stackrel{\text{def}}{=} \{2s-1, \dots, s+r-1\},$$

$$I_*^t \stackrel{\text{def}}{=} \{s+r, s+r+1, \dots, \tilde{N}-1\}.$$

Очевидно, что $I_*^h \cup I_*^m \cup I_*^t = J'_{\tilde{N}-1}$.

Рассмотрим цепочку векторов $\tilde{A} \stackrel{\text{def}}{=} \{\mathbf{a}_{-1}, \mathbf{a}_0, \dots, \mathbf{a}_{\tilde{N}-1}\}$, далее будем предполагать, что выполнено условие

(A) Цепочка векторов $\tilde{A} \stackrel{\text{def}}{=} \{\tilde{\mathbf{a}}_j\}$ полная и справедливы соотношения

$$\tilde{\mathbf{a}}_j = \mathbf{a}_j \quad \text{при} \quad j \in I_*^h,$$

$$\tilde{\mathbf{a}}_j = \mathbf{a}_{2j-2s+1} \quad \text{при} \quad j \in I_*^m,$$

$$\tilde{\mathbf{a}}_j = \mathbf{a}_{j+r-s} \quad \text{при} \quad j \in I_*^t.$$

Совокупность $\{X, A, \tilde{X}, \tilde{A}\}$ будем называть *двухинтервальной гребенчатой структурой*.

Построим систему функций $\{\tilde{\omega}_j\}_{j \in J'_{\tilde{N}-1}}$, отыскивая ее из соотношений

$$\sum_{j \in J'_{\tilde{N}-1}} \tilde{\mathbf{a}}_j \tilde{\omega}_j(t) = \varphi(t) \quad \forall t \in \tilde{G}, \quad \tilde{\omega}_j(t) \equiv 0 \quad \forall t \in \tilde{G} \setminus \tilde{S}_j, \quad j \in J'_{\tilde{N}-1}.$$

Рассмотрим пространства $\mathbb{S} \stackrel{\text{def}}{=} tCl_p \mathcal{L}\{\omega_j\}_{j \in J'_{\tilde{N}-1}}$ и $\tilde{\mathbb{S}} \stackrel{\text{def}}{=} tCl_p \mathcal{L}\{\tilde{\omega}_j\}_{j \in J'_{\tilde{N}-1}}$, где $\mathcal{L}\{\dots\}$ — линейная оболочка функций, указанных в фигурных скобках, а Cl_p — замыкание в топологии поточечной сходимости.

Замечание 1. Ввиду предположений относительно вектор-функции $\varphi(t)$ верно, что $\dim \mathbb{S} = N+1$ и $\dim \tilde{\mathbb{S}} = \tilde{N}+1$.

Матрица вложения

Пусть (c, d) — некоторый интервал вещественной оси. Обозначим через $C\langle c, d \rangle$ линейное пространство функций, непрерывных на (c, d) и имеющих

конечные пределы на концах этого интервала. Введем линейное пространство функций, определяемое прямым произведением пространств $C\langle x_i, x_{i+1} \rangle$, $i=0, 1, \dots, N-1$, а именно

$$C_X \stackrel{\text{def}}{=} \bigotimes_{i=0}^{N-1} C\langle x_i, x_{i+1} \rangle.$$

Замечание 2. Верно, что $\omega_j \in C_X$, $j \in J'_{N-1}$ и справедливо включение

$$\mathfrak{S} \subset \mathbb{S} \subset C_X.$$

Пусть $\{g_i\}_{i \in J'_{N-1}}$ — система функционалов над C_X , биортогональная системе функций $\{\omega_j\}_{j \in J'_{N-1}}$, $\langle g_i, \omega_j \rangle = \delta_{i,j}$, со свойством $\text{supp } g_i \subset [x_i, x_{i+1} + \varepsilon]$, $j \in J'_{N-1}$, $\text{supp } \omega_j \subset [x_0, x_0 + \varepsilon]$; здесь $\varepsilon > 0$ — произвольное положительное число.

Рассмотрим матрицу \mathfrak{P} с элементами

$$p_{i,j} \stackrel{\text{def}}{=} \langle g_i, \tilde{\omega}_j \rangle, \quad i \in J'_{N-1}, j \in J'_{N-1}.$$

Рассмотрим три группы значений индекса j , а именно

$$I_H \stackrel{\text{def}}{=} \{-1, 0, 1, \dots, 2s-2\}, \quad I_T \stackrel{\text{def}}{=} \{2r, 2r+1, \dots, N-1\},$$

$$I_M \stackrel{\text{def}}{=} \{2s-1, 2s, \dots, 2r-1\}.$$

Очевидно, что $I_H \cup I_T \cup I_M = J'_{N-1}$, $I_*^h = I_H$.

Теорема 1. Элементы матрицы \mathfrak{P} вычисляются по формулам

$$p_{i,j} = \delta_{i,j} \quad \text{при} \quad \forall i \in J'_{N-1}, j \in I_H,$$

$$p_{i,j} = \delta_{j, i-s+r} \quad \text{при} \quad \forall i \in J'_{N-1}, j \in I_T;$$

остальные элементы вычисляются при $q \in \{s-1, s, \dots, r-1\}$ по формулам

$$p_{i,2q} = p_{i,2q+1} = 0 \quad \forall i \in J'_{N-1} \setminus \{s+q-1, s+q\},$$

$$p_{s+q-1,2q+1} = 0, \quad p_{s+q,2q+1} = 1;$$

при $q \in \{s, s+1, \dots, r-1\}$ справедливы соотношения

$$p_{s+q-1,2q} = \frac{\det(\mathbf{a}_{2q}, \mathbf{a}_{2q+1})}{\det(\mathbf{a}_{2q-1}, \mathbf{a}_{2q+1})}, \quad p_{s+q,2q} = \frac{\det(\mathbf{a}_{2q-1}, \mathbf{a}_{2q})}{\det(\mathbf{a}_{2q-1}, \mathbf{a}_{2q+1})}.$$

Матрица продолжения

Рассмотрим систему функционалов $\{\tilde{g}_i\}_{i \in J'_{\tilde{N}-1}}$ биортогональную к системе $\{\tilde{\omega}_j\}_{j \in J'_{\tilde{N}-1}}$, $\langle \tilde{g}_i, \tilde{\omega}_j \rangle = \delta_{i,j}$ со свойством

$$\text{supp } \tilde{g}_i \subset [\tilde{x}_i, \tilde{x}_i + \varepsilon) \quad \forall \varepsilon > 0, \quad J'_{\tilde{N}-1}, \quad \text{supp } \tilde{g}_{-1} \subset [\tilde{x}_0, \tilde{x}_0 + \varepsilon).$$

Используя соотношение $\langle \tilde{g}_i, \varphi \rangle = \tilde{a}_i$, вычислим матрицу \mathfrak{Q} размеров $\tilde{N}+1 \times N+1$ с элементами $q_{i,j} \stackrel{\text{def}}{=} \langle \tilde{g}_i, \omega_j \rangle$; здесь $i \in J'_{\tilde{N}-1}$, $j \in J'_{N-1}$.

Теорема 2. Элементы матрицы \mathfrak{Q} вычисляются по следующим формулам.

1. Для всех $i \in J'_{\tilde{N}-1}$ справедливы соотношения

$$q_{i,j} = \delta_{i,j} \quad \text{при } j \in I_H,$$

$$q_{i,j} = \delta_{i,i+s-r} \quad \text{при } j \in I_T.$$

2. Кроме того,

$$q_{i,2q} = 0 \quad \forall i \in J'_{\tilde{N}-1} \setminus \{s+q\}, \quad s \leq q \leq r-1,$$

$$q_{i,2s-1} = 0, \quad \forall i \in J'_{\tilde{N}-1} \setminus \{2s-1, 2s\}.$$

$$q_{i,2q+1} = 0 \quad \forall i \in J'_{\tilde{N}-1} \setminus \{s+q+1\} \quad \text{при } s \leq q \leq r-2,$$

$$q_{2s-1,2s-1} = 1, \quad q_{i,2r-1} = 0, \quad \forall i \in J'_{\tilde{N}-1}.$$

3. Наконец, справедливы соотношения

$$q_{s+q,2q} = \frac{\det(\mathbf{a}_{2q-1}, \mathbf{a}_{2q+1})}{\det(\mathbf{a}_{2q-1}, \mathbf{a}_{2q})} \quad \text{при } s \leq q \leq r-1, \quad (2)$$

$$q_{s+q+1,2q+1} = \frac{\det(\mathbf{a}_{2q+3}, \mathbf{a}_{2q+2})}{\det(\mathbf{a}_{2q+1}, \mathbf{a}_{2q+2})} \quad \text{при } s-1 \leq q \leq r-2, \quad (3)$$

Теорема 3. Верно равенство

$$\mathfrak{Q} \mathfrak{P}^T = I,$$

где I — единичная квадратная матрица порядка $\tilde{N}+1$.

Вэйвлетное разложение

Рассмотрим оператор P проектирования пространства \mathbb{S} на подпространство $\tilde{\mathbb{S}}$, задаваемый формулой

$$P_u \stackrel{\text{def}}{=} \sum_{j \in J'_{\tilde{N}-1}} \langle \tilde{g}_i, u \rangle \tilde{\omega}_j \quad \forall u \in \mathbb{S},$$

и введем оператор $PQ = \mathcal{I} - P$, где \mathcal{I} — тождественный в \mathbb{S} оператор.

Пространством вейвлетов (всплесков) называется пространство $\mathbb{W} \stackrel{\text{def}}{=} QS$.

Таким образом, получаем прямое разложение $\mathbb{S} = \tilde{\mathbb{S}} + \mathbb{W}$ — *сплайн-вейвлетное разложение* пространства \mathbb{S} . Пусть $u \in \mathbb{S}$, тогда можно получить два представления элемента u

$$u = \sum_{j \in J'_{N-1}} c_j \omega_j, \quad \sum_{i \in J'_{N-1}} a_i \tilde{\omega}_i, \quad \sum_{j \in J'_{N-1}} b_j \omega_j,$$

где

$$a_i \stackrel{\text{def}}{=} \langle \tilde{g}_i, u \rangle, \quad b_j, c_j \in \mathbb{R}^1.$$

Тогда *формулы реконструкции* имеют вид:

$$c_j = \sum_{i \in J'_{N-1}} a_i \varphi_{i,j} + b_j, \quad j \in J'_{N-1}.$$

Формулы декомпозиции представим в матричном виде:

$$\mathbf{b} = \mathbf{c} - \mathfrak{F}^T \mathfrak{Q} \mathbf{c}, \quad \mathbf{a} = \mathfrak{Q} \mathbf{c},$$

где

$$\mathbf{a} \stackrel{\text{def}}{=} (a_{-1}, \dots, a_{\tilde{N}-1})^T, \quad \mathbf{b} \stackrel{\text{def}}{=} (b_{-1}, \dots, b_{N-1})^T,$$

Вектор \mathbf{a} называется *основным потоком*, а вектор \mathbf{b} — *вейвлетным потоком* при сплайн-вейвлетном разложении *исходного потока* \mathbf{c} .

Заключение

В работе представлены алгоритмы декомпозиции и реконструкции сплайн-вейвлетного разложения для двухинтервальной гребенчатой структуры на неравномерной сетке. Данные алгоритмы просты в реализации и позволяют экономить вычислительные ресурсы с учетом возможностей параллельных вычислительных систем.

Литература

1. Yu. K. Dem'yanovich, V.O. Dron'. Wavelet decomposition on a comb structure // J. of Math. Sci., New York 189, No. 3, 2013, pp. 402–421.
2. Ю.К. Демьянович, И.Д. Мирошниченко. Гнездовые сплайн-вейвлетные разложения // Проблемы математического анализа. Вып. 64, 2012, с. 51–61.
3. Yu. K. Dem'yanovich. Interference in spline-wavelet decompositions // J. of Math. Sci., New York 186, No. 2, 2012, pp. 234–246.

ПРОСТРАНСТВА МИНИМАЛЬНЫХ СПЛАЙНОВ И КАЛИБРОВОЧНЫЕ СООТНОШЕНИЯ*

Ю.К. Демьянович

Yuri.Demjanovich@gmail.com

Аннотация: В работе даны необходимые и достаточные условия существования и гладкости пространств $\mathbb{S}_m(X, A, \varphi)$ (вообще говоря, неполиномиальных) сплайнов, и доказано, что во множестве таких пространств существует единственное (при фиксированной сетке) пространство класса C^{m-1} — пространство B_φ -сплайнов порядка m . Установлена вложенность пространств B_φ -сплайнов порядка m , построенных на вложенных сетках, и получены калибровочные соотношения для их координатных функций.

Введение

Вложенность пространств сплайнов важна для построения сплайн-вэйвлетных (всплесковых) разложений (см. [1–8]). Условия вложенности пространств неполиномиальных сплайнов на неравномерных сетках рассматривались в работах [3–8], однако, они касались либо пространств сплайнов небольших порядков [3–7], либо пространств негладких сплайнов (см. [8]).

Цель данной работы представлены достаточные условия вложенности для пространств гладких сплайнов любого порядка и найдены калибровочные соотношения для координатных функций этих пространств. Здесь получены необходимые и достаточные условия существования и гладкости пространств сплайнов лагранжева типа, и установлено, что во множестве таких пространств порядка m существует единственное (при фиксированной сетке) пространство класса C^{m-1} .

Пространства (X, A, φ) -сплайнов порядка m

На конечном или бесконечном интервале (α, β) вещественной оси \mathbb{R}^1 введем сетку $X \stackrel{\text{def}}{=} \{x_j\}_{j \in \mathbb{Z}}$,

$$X : \dots < x_{-1} < x_0 < x_1 < \dots; \text{ пусть } \alpha \stackrel{\text{def}}{=} \lim_{j \rightarrow -\infty} x_j, \beta \stackrel{\text{def}}{=} \lim_{j \rightarrow +\infty} x_j.$$

Пусть $\varphi(t)$ — $m+1$ -компонентная вектор-функция (вектор-столбец) с компонентами из $C^m[\alpha, \beta]$, такая, что вронскиан ее компонент отделен от нуля:

* Работа частично поддержана грантами РФФИ 10-01-00245 и 10-01-297

$$|\det(\varphi, \varphi', \varphi'', \dots, \varphi^{(m)})(t)| \geq c > 0 \quad \forall t \in [\alpha, \beta]. \quad (1)$$

Рассмотрим последовательность \mathbf{A} вектор-столбцов \mathbf{a}_i пространства \mathbb{R}^{i+1} , $\mathbf{A} \stackrel{\text{def}}{=} \{\dots, \mathbf{a}_{-1}, \mathbf{a}_0, \mathbf{a}_1, \dots\}$, и матрицы $A_j \stackrel{\text{def}}{=} (\mathbf{a}_{j-m}, \mathbf{a}_{j-m+1}, \dots, \mathbf{a}_j)$, $j \in \mathbb{Z}$.

Введем обозначение $\mathcal{A}_j \stackrel{\text{def}}{=} \{A \mid \det A_j \neq 0 \quad \forall j \in \mathbb{Z}\}$.

Пусть $\varphi_j^{(S)} \stackrel{\text{def}}{=} \varphi^{(S)}(x_j)$, $h_X \stackrel{\text{def}}{=} \sup_{j \in \mathbb{Z}} (x_{j+1} - x_j)$, $M \stackrel{\text{def}}{=} \bigcup_{j \in \mathbb{Z}} (x_j, x_{j+1})$,

$$S_j \stackrel{\text{def}}{=} [x_j, x_{j+m+1}], \quad J_k \stackrel{\text{def}}{=} \{k-m, \dots, k\}$$

При $\mathbf{A} \in \mathcal{A}$ аппроксимационные соотношения

$$\sum_{j'=i-m}^i \mathbf{a}_{j'} \omega_{j'}(t) \equiv \varphi(t) \quad \forall t \in (x_i, x_{i+1}) \quad i \in \mathbb{Z}, \quad \text{supp } \omega_j \subset S_j \quad \forall j \in \mathbb{Z} \quad (2)$$

однозначно определяют функции $\omega_j(t)$, $t \in M$, $j \in \mathbb{Z}$.

Благодаря предположению (1) функции ω_j , $j \in \mathbb{Z}$, образуют линейно независимую систему. Рассмотрим линейное пространство $\mathbb{S}_m(X, \mathbf{A}, \varphi)$ минимальных (X, \mathbf{A}, φ) -сплайнов порядка m , определяемое соотношением

$$\mathbb{S}_m(X, \mathbf{A}, \varphi) \stackrel{\text{def}}{=} Cl_p \mathcal{L} \{\omega_j\}_{j \in \mathbb{Z}};$$

здесь через \mathcal{L} обозначена линейная оболочка, а Cl_p означает замыкание в топологии поточечной сходимости.

Непрерывность и непрерывная дифференцируемость минимальных сплайнов

Теорема 1. Пусть $\varphi^{(S)} \in C(\alpha, \beta)$, где S — натуральное число. Для того, чтобы функции $\omega_j^{(S)}(t)$ ($\forall j \in \mathbb{Z}, t \in M$) могли быть продолжены до функций, непрерывных на интервале (α, β) , необходимо и достаточно выполнение соотношений

$$\det(\mathbf{a}_{j-m}, \mathbf{a}_{j-m+1}, \dots, \mathbf{a}_{j-1}, \varphi_j^{(S)}) = 0 \quad \forall j \in \mathbb{Z}.$$

B_φ -сплайны порядка m

Рассмотрим частный случай соотношений (2), выбирая в них цепочку векторов \mathbf{a}_j специальным образом. Положим $\mathbf{d}_j^T \mathbf{x} \equiv \det(\mathbf{a}_{j-m}, \mathbf{a}_{j-m+1}, \dots, \mathbf{a}_{j-1}, \mathbf{x})$, $\forall \mathbf{x} \in \mathbb{R}^{m+1}$.

Введем векторы \mathbf{a}_j^* с помощью символического определителя

$$\mathbf{a}_j^* \stackrel{\text{def}}{=} \det \begin{pmatrix} \varphi_{j+1} & \varphi'_{j+1} & \varphi''_{j+1} & \dots & \varphi^{m-1}_{j+1} \\ \mathbf{d}_{j+2}^T \varphi_{j+1} & \mathbf{d}_{j+2}^T \varphi'_{j+1} & \mathbf{d}_{j+2}^T \varphi''_{j+1} & \dots & \mathbf{d}_{j+2}^T \varphi^{m-1}_{j+1} \\ \mathbf{d}_{j+3}^T \varphi_{j+1} & \mathbf{d}_{j+3}^T \varphi'_{j+1} & \mathbf{d}_{j+3}^T \varphi''_{j+1} & \dots & \mathbf{d}_{j+3}^T \varphi^{m-1}_{j+1} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{d}_{j+m}^T \varphi_{j+1} & \mathbf{d}_{j+m}^T \varphi'_{j+1} & \mathbf{d}_{j+m}^T \varphi''_{j+1} & \dots & \mathbf{d}_{j+m}^T \varphi^{m-1}_{j+1} \end{pmatrix}$$

и обозначим $\mathbf{A}^* \stackrel{\text{def}}{=} \{\mathbf{a}_j^*\}_{j \in \mathbb{Z}}$.

Теорема 2. Если выполнено условие (1), то существует такое $\delta = \delta_\varphi > 0$, что при $h_X < \delta$ справедлива импликация $\mathbf{A}^* \in \mathcal{A}$.

Считая, что $\mathbf{A}^* \in \mathcal{A}$, в аппроксимационных соотношениях (2) положим $\mathbf{a}_j = \mathbf{a}_j^*$; сплайны, полученные из этих соотношений обозначим ω_j^* .

Теорема 3. Справедлива формула $\omega_j^* \in C^{m-1}(\alpha, \beta) \forall j \in \mathbb{Z}$.

Рассматривая линейное пространство $S_m^*(X, \varphi) \stackrel{\text{def}}{=} Cl_b \mathcal{L}\{\omega_j^*\}_{j \in \mathbb{Z}}$, получаем $S_m^*(X, \varphi) \subset C^{m-1}(\alpha, \beta)$. Пространство $S_m^*(X, \varphi)$ называется пространством B_φ -сплайнов.

Обозначим $\mathfrak{S}_m(X, \varphi)$ множество пространств сплайнов порядка m при фиксированной сетке X и фиксированной вектор-функции $\varphi(t)$:

$$\mathfrak{S}_m(X, \varphi) \stackrel{\text{def}}{=} \{S_m(X, \mathbf{A}, \varphi) \mid \forall \mathbf{A} \in \mathcal{A}\}.$$

Теорема 4. Во множестве \mathfrak{S}_m существует единственное пространство класса $C^{m-1}(\alpha, \beta)$; таким пространством является пространство $S_m^*(X, \varphi)$.

Теорема 5. Каждая функция $\omega_j^*(t)$ определяется значениями вектор-функции $\varphi(t)$ на множестве $\text{supp } \omega_j^*$.

Калибровочные соотношения

Укрупним исходную сетку X выбрасыванием узла x_{k+1} , а именно, положим

$$\tilde{x}_j = x_j \text{ при } j \leq k, \quad \tilde{x}_j = x_{j+1} \text{ при } j \geq k+1$$

и рассмотрим новую сетку $\tilde{X} \stackrel{\text{def}}{=} \{\tilde{x}_j\}_{j \in \mathbb{Z}}$,

$$\tilde{X} : \dots < \tilde{x}_{-1} < \tilde{x}_0 < \tilde{x}_1 < \dots$$

Используя новую сетку и полагая $\tilde{\varphi}_j \stackrel{\text{def}}{=} \varphi(\tilde{x}_j)$, $\tilde{\varphi}_j^{(S)} \stackrel{\text{def}}{=} \varphi^{(S)}(\tilde{x}_j)$, введем векторы $\tilde{\mathbf{a}}_j^*$ с помощью символического определителя

$$\tilde{\mathbf{a}}_j \stackrel{\text{def}}{=} \det \begin{pmatrix} \tilde{\varphi}_{j+1} & \tilde{\varphi}'_{j+1} & \tilde{\varphi}''_{j+1} & \dots & \tilde{\varphi}_{j+1}^{m-1} \\ \tilde{\mathbf{d}}_{j+2}^T \tilde{\varphi}_{j+1} & \tilde{\mathbf{d}}_{j+2}^T \tilde{\varphi}'_{j+1} & \tilde{\mathbf{d}}_{j+2}^T \tilde{\varphi}''_{j+1} & \dots & \tilde{\mathbf{d}}_{j+2}^T \tilde{\varphi}_{j+1}^{m-1} \\ \tilde{\mathbf{d}}_{j+3}^T \tilde{\varphi}_{j+1} & \tilde{\mathbf{d}}_{j+3}^T \tilde{\varphi}'_{j+1} & \tilde{\mathbf{d}}_{j+3}^T \tilde{\varphi}''_{j+1} & \dots & \tilde{\mathbf{d}}_{j+3}^T \tilde{\varphi}_{j+1}^{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ \tilde{\mathbf{d}}_{j+m}^T \tilde{\varphi}_{j+1} & \tilde{\mathbf{d}}_{j+m}^T \tilde{\varphi}'_{j+1} & \tilde{\mathbf{d}}_{j+m}^T \tilde{\varphi}''_{j+1} & \dots & \tilde{\mathbf{d}}_{j+m}^T \tilde{\varphi}_{j+1}^{m-1} \end{pmatrix}$$

Рассмотрим аппроксимационные соотношения

$$\sum_{j'=i-m}^i \tilde{\mathbf{a}}_{j'} \tilde{\omega}_{j'}(t) \equiv \varphi(t) \quad \forall t \in (\tilde{x}_i, \tilde{x}_{i+1}) \quad i \in \mathbb{Z}, \quad \text{supp } \tilde{\omega}_j \subset [\tilde{x}_j, \tilde{x}_{j+m+1}] \quad \forall j \in \mathbb{Z}$$

Пространство B_φ -сплайнов на новой сетке обозначим

$$\mathbb{S}_m^*(\tilde{X}, \varphi) \stackrel{\text{def}}{=} Cl_p \mathcal{L} \{ \tilde{\omega}_j^* \}_{j \in \mathbb{Z}}.$$

Теорема 6. Если сетка \tilde{X} столь мелкая, что цепочка $\tilde{\mathbf{A}}^* \stackrel{\text{def}}{=} \{ \tilde{\mathbf{a}}_j^* \}_{j \in \mathbb{Z}}$ полная, то пространство B_φ -сплайнов на сетке X содержит пространство B_φ -сплайнов на сетке \tilde{X} :

$$\mathbb{S}_m^*(\tilde{X}, \varphi) \subset \mathbb{S}_m^*(X, \varphi).$$

Калибровочные соотношения имеют вид

$$\begin{aligned} \tilde{\mathbf{a}}_{k-m}^* \tilde{\omega}_{k-m}^*(t) + \tilde{\mathbf{a}}_{k-m+1}^* \tilde{\omega}_{k-m+1}^*(t) + \dots + \tilde{\mathbf{a}}_{k-1}^* \tilde{\omega}_{k-1}^*(t) + \tilde{\mathbf{a}}_k^* \tilde{\omega}_k^*(t) &\equiv \\ &\equiv \mathbf{a}_{k-m}^* \omega_{k-m}^*(t) + \mathbf{a}_{k-m+1}^* \omega_{k-m+1}^*(t) + \dots + \\ &+ \mathbf{a}_{k-1}^* \omega_{k-1}^*(t) + \mathbf{a}_k^* \omega_k^*(t) + \mathbf{a}_{k+1}^* \omega_{k+1}^*(t) \quad \forall t \in (\alpha, \beta). \end{aligned}$$

Следствие. При выполнении условий теоремы 2 для любой сетки X' со свойствами $X' \subset X$, $h_{X'} \leq \delta_\varphi$, справедливо включение

$$\mathbb{S}_m^*(X', \varphi) \subset \mathbb{S}_m^*(X, \varphi).$$

Заключение

В заключение отметим, что установленные утверждения относятся к сплайнам лагранжева типа, т. е. к сплайнам ω_j , носители которых имеют вид $[x_j, x_{j+m+1}]$.

Однако, аппроксимационные соотношения могут порождать минимальные сплайны и в других случаях расположения носителей. Изучение неко-

торых из этих случаев проводилось ранее в иных направлениях, однако, нет более или менее общих результатов о гладкости и вложенности получаемых на этом пути пространств сплайнов; эти результаты представляются весьма загадочными, хотя, по-видимому, и могут быть получены аналогичными методами.

Литература

1. *Стечкин С.Б., Субботин Ю.Н.* Сплайны в вычислительной математике. М., 1976. 248 с.
 2. *Новиков И.Я., Протасов В.Ю., Скопина М.А.* Теория всплесков. М. 2005.
 3. *Демьянович Ю.К.* Всплесковые разложения на неравномерной сетке// Труды СПбМО, 2007. Т.13. С.27–51.
 4. *Демьянович Ю.К.* Вложенные пространства тригонометрических сплайнов и их всплесковое разложение// Ж. Математические Заметки, 2005. Т.78, вып.5. С.22–26.
 5. *Демьянович Ю.К. Косоголов О.М.* Калибровочные соотношения для неполиномиальных сплайнов// Проблемы математического анализа. 2009. №43. С.3–19.
 6. *Dem'yanovich Yu. K., Miroshnichenko I.D.* Calibration relations to of the fourth order//Journal of Math.Sciences, Vol.178, No.6. 2011. P.576–588.
 7. *Dem'yanovich Yu. K.* Embedding of non-polynomial spline spaces//Mathematical Sciences 2012, 6:8 (Springer Open, 2012).
 8. *Демьянович Ю.К.* Минимальные сплайны и всплески//Вестник СПбГУ, №2. 2008. С.8–22.
-

ПЕРВИЧНЫЙ КРИПТОАНАЛИЗ И РАСПАРАЛЛЕЛИВАНИЕ НОВОГО АЛГОРИТМА ШИФРОВАНИЯ

Рассмотрен новый алгоритм симметричного шифрования, проведено изучение его криптостойкости по отношению к некоторым видам атак, предложен способ эффективного распараллеливания проводимых преобразований.

Чистяков П.П.

Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра параллельных алгоритмов

Аннотация: Предложен новый алгоритм симметричного шифрования, получивший название SMN89. Проведён его анализ с точки зрения его криптостойкости и возможности распараллеливания производимых в ходе его применения вычислений. Установлено, что алгоритм имеет высокий запас криптостойкости против атак таких видов, как сдвиговые атаки, атаки методами «грубой силы», «встреча посередине» и интерполяции. Выявлено ускорение при использовании крупномодульного распараллеливания вычислений.

1. Введение

Существует много алгоритмов шифрования высокой стойкости (таких, как ГОСТ 28147-89, семейство RC и др.). Однако вследствие постоянного развития криптоанализа всё чаще стали проводиться успешные атаки на существующие алгоритмы, что приводит к необходимости разрабатывать и изучать новые алгоритмы шифрования. Целью данной работы является описание предлагаемого автором алгоритма шифрования, определение возможности для оптимизации вычислений в ходе его применения, в том числе за счёт распараллеливания, а также предварительная оценка криптостойкости алгоритма по отношению к некоторым видам атак.

По результатам проведённых исследований можно сделать следующие выводы: алгоритм SMN89 имеет высокий запас криптостойкости против атак методом «грубой силы» вследствие очень большой длины ключа; из-за наличия преобразований, которые вносят отличия между любыми двумя подряд идущими раундами, сдвиговая атака на алгоритм не может быть проведена; проведение атак вида «встреча посередине», целью которых обычно является разбиение ключа на две половины, поиск которых можно вести изначально независимо друг от друга, не оправдано, поскольку из-за использования ключа целиком на любом раунде алгоритма желаемого сокращения области перебора не происходит.

Экспериментальным путём установлено, что при крупномодульном распараллеливании вычислений в ходе применения алгоритма на двухъядерном

процессоре максимальное достигаемое ускорение равно 2. По всем рассмотренным диапазонам размера открытого текста ускорение составило в среднем 32%; что касается зависимости ускорения от количества выполняемых раундов алгоритма, то здесь наблюдалась стабилизация значения ускорения около отметки в 1.5 с увеличением числа раундов.

2. Актуальность проблемы

В последние десятилетия в криптографии как в науке о защите информации наблюдается огромный прогресс, связанный с непрерывным развитием вычислительных систем и постоянной необходимостью в защите компьютерной информации. В настоящее время криптографические методы доступны не только государственным структурам, но и обычным людям и организациям; сама разработка криптографических алгоритмов также осуществляется зачастую частными лицами.

Криптографические методы в современном мире используются не только для защиты информации, но и в качестве основы многих новейших информационных технологий, таких, как, например, электронный документооборот, электронные деньги и многих других.

3. Краткий обзор некоторых теоретических аспектов

3.1 Базовая модель криптографии

Сообщения передаются от отправителя к получателю по открытому каналу, к которому имеет доступ «противник». Он либо только прослушивает канал (пассивный перехват сообщений), либо модифицирует передаваемые сообщения и посылает свои (активный перехват сообщений). Перед отправкой сообщения в открытый канал отправителю необходимо подвергнуть исходную информацию специальному преобразованию (зашифровыванию), позволяющему скрыть её истинное содержание; соответственно после приёма на другом конце нужно выполнить обратное преобразование (расшифровывание). Открытый текст — информация, подающаяся на вход шифратора, шифртекст — информация на входе дешифратора.

3.2 Три категории криптоалгоритмов.

Существуют бесключевые, одноключевые (общий секретный ключ для отправителя и получателя) и двухключевые (на разных этапах вычислений применяются два вида ключей: для зашифровывания информации, предназначенной конкретному получателю, отправитель использует уникальный открытый ключ адресата-получателя; для расшифровывания получатель использует парный секретный ключ) алгоритмы.

3.3 Общая схема шифрования.

- 1). Зашифровывание: $C = Ek_1(M)$.
- 2). Расшифровывание: $M = Dk_2(C)$.

От ключа зависит результат работы алгоритма.

3.4 Основные структура алгоритмов симметричного шифрования.

- 1). Сеть Фейстеля.
- 2). Подстановочно-перестановочная сеть.
- 3). «Квадрат».

3.5 Режимы работы алгоритмов.

- 1). ECB (электронная кодовая книга).
- 2). CBC (сцепление блоков шифра).
- 3). CFB (обратная связь по шифртексту).
- 4). OFB (обратная связь по выходу).

3.6 Эффективность криптоалгоритма – отношение временных затрат криптоаналитика на вскрытие шифровки к временным затратам криптографа на создание шифровки.

4. Структура алгоритма CMN89.

Алгоритм CMN89 имеет смешанную структуру; она представляет собой модификацию сети Фейстеля, на каждом раунде которой производится преобразования (в том числе подстановки и перестановки) с блоками данных, представленными в виде квадратного массива байтов.

Алгоритм шифрует данные блоками длиной 64 байта, подающимися на вход шифратора по 4 одновременно в виде единой 256-байтной строки открытого текста. Часть параметров являются переменными: количество раундов может варьироваться от 0 до 255, секретный ключ может иметь длину от 0 до 255 байтов. Далее в статье для обозначения этих величин используются переменные R и $Lange$, соответственно.

Перед шифрованием выполняются следующие преобразования: производится постепенное удвоение длины ключа за счёт приписывания в конец его копии до того момента, пока длина не превысит 256 байтов, затем оставляются только первые 256 байтов, они разбиваются на 16 16-байтных подключей ($K_0 \dots K_{15}$) раундов.

64-байтный блок открытого текста разделяется на 4 субблока (N_1, N_2, N_3, N_4) по 16 байтов. После с ними производятся $16 * R$ подраундов преобразований (R раундов, в каждом по 16 подраундов для полного перешифования входных данных). Каждый из них состоит из следующих операций:

1. Перестановка байтов местами внутри субблока. В данной версии алгоритма она имеет вид (5, 2, 11, 6, 8, 15, 12, 1, 14, 3, 0, 7, 9, 4, 13, 10) и применяется к субблокам N1 и N4.

2. Наложение подключа на результат перестановки с использованием операции XOR: номер подключа определяется остатком от деления на 16 глобального номера текущего подраунда. Таким образом, при зашифровывании используется прямой циклический порядок наложения подключей: от K0 до K15.

3. Далее субблоки N1 и N4 представляются в виде матриц размера 4*4. Для каждой из них производится циклический сдвиг байтов вправо в каждой из строк на количество позиций, определяемое глобальным номером kkk текущего подраунда и собственным номером строки sn (от 1 до 4): $kkk\%4+sn-1$.

4. Побайтная табличная замена в субблоках N1 и N4. Текущий вариант таблицы заимствован из алгоритма Camellia.

5. Наложение субблоков друг на друга с помощью операции XOR, перемена их местами. Итог наложения результата преобразований 1–4 субблока N1 на субблок N3 становится первым субблоком нового формируемого 64-байтного блока, вторым субблоком становится начальный субблок N1, третьим — начальный субблок N4, четвертым — итог наложения результата преобразований 1–4 субблока N4 на субблок N2.

6. Разложение блока на две составляющие, при этом номер байта, с помощью которого формируется вторая составляющая, определяется как $kkk\%64$, где kkk — глобальный номер текущего подраунда.

7. Составление нового 64-байтного блока из двух полученных составляющих и его разделение на 16-байтные субблоки либо для осуществления преобразований 1–6 на следующем подраунде, либо составления из них очередной 64-байтной части выходной 256-байтной строки.

В конце зашифровывания производится увеличение всех байтов на $Lange\%5$.

При расшифровывании операции производятся также с 256-байтными строками. Сначала выполняется уменьшение всех байтов на $Lange\%5$. Далее эти строки разбиваются на 4 блока по 64 байта, которые обрабатываются по отдельности. Порядок операций на каждом из $16*R$ подраундов следующий:

1. Восстановление блока: номер восстанавливаемого байта определяется как $kkk\%64$, где kkk — глобальный номер текущего подраунда. Важно отметить, что при расшифровывании глобальная нумерация подраундов идет в обратном порядке.

2. Разбиение 64-байтного блока на 4 субблока (N1, N2, N3, N4) по 16 байтов. Перестановка байтов местами внутри субблоков N2 и N3 (такая же, как и при зашифровывании).

3. Наложение подключа на результат перестановки с помощью операции XOR: номер подключа определяется остатком от деления на 16 глобального

номера текущего подраунда. При расшифровывании нумерация подраундов идёт в обратном порядке и используется обратный циклический порядок наложения подключей: от K15 до K0.

4. Субблоки N2 и N3 представляются в виде матриц размера 4×4 . Для каждой из них производится циклический сдвиг байтов вправо в каждой из строк на количество позиций, определяемое глобальным номером текущего подраунда и собственным номером строки (так же, как и при зашифровывании, но при обратной нумерации подраундов).

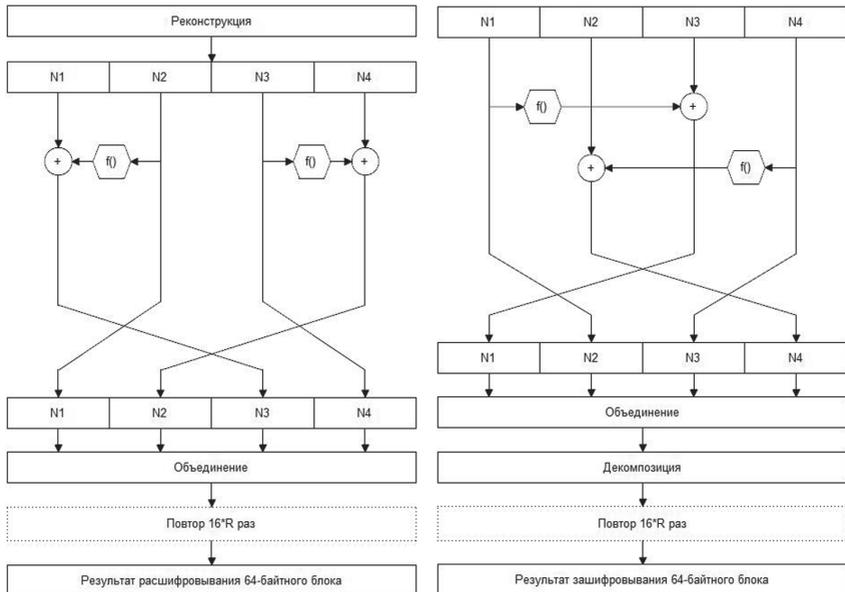
5. Побайтная табличная замена в тех же субблоках с помощью той же таблицы, что и при зашифровывании.

6. Наложение субблоков друг на друга с помощью операции XOR и их перестановка местами. Первым субблоком нового формируемого 64-байтного блока становится начальный субблок N2, вторым — итог наложения результата преобразований 2–5 субблока N3 на субблок N4, третьим – итог наложения результата преобразований 2–5 субблока N2 на субблок N1, четвёртым — первоначальный субблок N3.

7. Составление единого преобразованного 64-байтного блока из 16-байтных субблоков либо для осуществления преобразований 1–6 на следующем подраунде, либо записи всего этого блока как очередной 64-байтной части выходной 256-байтной строки.

Схемы зашифровывания и расшифровывания представлены ниже.

Функция $f()$ при зашифровывании соответствует преобразованиям 1–4, а при расшифровывании — преобразованиям 2–5.



5. Полученные результаты.

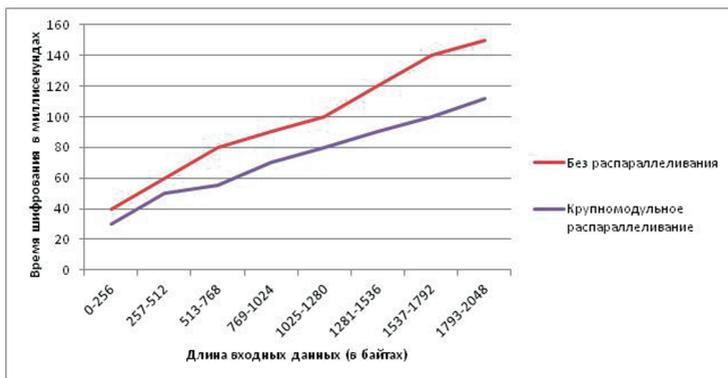
Разработана программная реализация предлагаемого алгоритма на языке программирования C#. В результате работы программы при наличии часто повторяющихся блоков исходного текста получается шифртекст с таким же количеством одинаковых блоков, что соответствует теоретическим результатам.

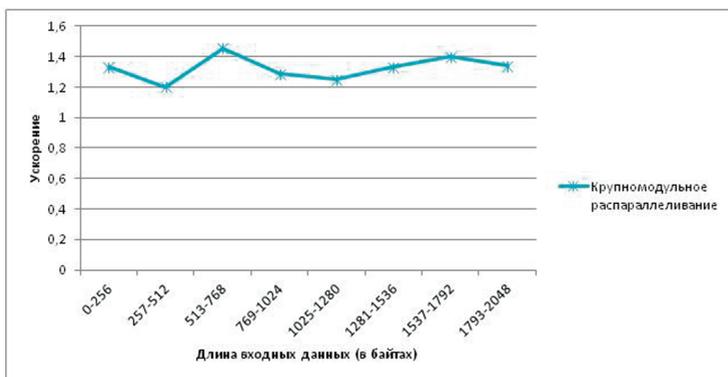
Численный эксперимент позволил оценить время шифрования и ускорение при распараллеливании в зависимости от длины входных данных и от количества раундов преобразований алгоритма.

Разделение по длине входных данных соответствует структуре работы алгоритма CMN89: данные шифруются блоками по 64 байта, подающимися на вход шифратора по 4 вместе в виде единой 256-байтной строки открытого текста. Измерения проводились при работе на 2-х ядерном процессоре Intel(R) Core(TM)2 Duo CPU.

Использование встроенных методов класса System.Threading.Tasks для распараллеливания вычислений позволило достичь определённого ускорения работы алгоритма. Полученные значения подтверждают, что зачастую применение мелко модульного распараллеливания, при котором выполняется параллельная обработка каждого из множества небольших циклов по отдельности (например, при расширении ключа, его наложении на субблоки, табличной замене, перестановках и т. п.), является неэффективным. Причина заключается в том, что накладные расходы на создание и уничтожение потоков вносят большой вклад в суммарное время работы алгоритма, существенно снижая положительный эффект от непосредственного параллельного выполнения этих циклов.

На основании полученных значений построены графики зависимости времени выполнения алгоритма CMN89 в последовательном режиме и в режиме крупномодульного распараллеливания и достигнутого ускорения от размера открытого текста, подвергающегося шифрованию (красным показано время шифрования без распараллеливания, фиолетовым — с распараллеливанием, голубым — ускорение, достигнутое при распараллеливании):





Из приведённых выше графиков видно, что во всех случаях крупномодульного распараллеливания вычислений, когда каждому из ядер назначалась полная обработка одного из 64-байтных блоков открытого текста, было достигнуто ускорение вычислений. Оптимальным на рассматриваемой системе можно считать шифрование (с помощью алгоритма SMN89) открытых текстов размером от 513 до 768 байтов в режиме крупномодульного распараллеливания (в этом случае ускорение достигает более 45%). Наименьший положительный эффект от распараллеливания наблюдается при размере исходного текста от 257 до 512 байтов, когда в среднем алгоритм работает быстрее не более чем на 20%. В среднем по всем указанным диапазонам длин исходных данных ускорение составило примерно 32%.

Время выполнения алгоритма и получаемое при распараллеливании вычислений ускорение зависит от количества раундов преобразований. Далее приведены графики соответствующих зависимостей (зелёным отмечено время шифрования без распараллеливания, голубым – с распараллеливанием, оранжевым — ускорение, достигнутое при распараллеливании). Как видно, при малом количестве раундов может происходить уменьшение скорости работы алгоритма при его распараллеливании (даже в его крупномодульном варианте): даже в случае распределения целых 64-байтных блоков на одно ядро процессора при небольшом числе раундов затраты на создание и уничтожение потоков велики, а также существенное влияние оказывает последовательная часть кода.

С увеличением количества раундов R преобразований алгоритма среднее значение наблюдаемого ускорения устанавливается в окрестности 1.5. Если приводить конкретные значения, то наиболее интересны некоторые точки локальных экстремумов: при $R=2, 11, 12$ ускорение достигло своего абсолютного максимума, равного 2; $R=9$ — единственный случай, при котором наблюдалось заметное ухудшение (замедление в 2 раза) работы алгоритма при распараллеливании; во второй половине диапазона возможных значений числа раундов локальный максимум был достигнут при $R=190$,

когда параллельная версия отработала более чем в 1.7 раза быстрее последовательной, что, скорее всего, объясняется оптимальной загрузкой вычислительных ядер на протяжении всего выполнения алгоритма в этом случае, а также эффективным использованием быстрой кэш-памяти.

6. Оценка криптостойкости

При анализе возможности проведения атак на алгоритм CMN89 были получены следующие оценки:

1. В методе «грубой силы» предполагается наличие у криптоаналитика некоторого шифрующего устройства; ему необходимо узнать «прошитый» в нём ключ; у него также есть нужное количество пар (открытый текст, соответствующий шифртекст) для произведения атаки с известным открытым текстом. В этом методе производится перебор всевозможных вариантов ключа до нахождения искомого. Перед зашифровыванием исходного открытого текста с помощью алгоритма CMN89 производится процедура расширения ключа. В результате «удлинённый» до 256 байтов исходный ключ разбивается на шестнадцать 16-байтных подключей. Каждый из этих подключей используется на определённом подраунде всех раундов преобразований. Таким образом, все биты расширенного ключа задействованы уже в первом раунде обработки 64-байтного блока исходных данных и у «противника» нет возможности найти искомый секретный ключ «по частям», не внося изменений в алгоритм, что предполагается невозможным (у него есть доступ к шифратору, но не к исходному коду).

В данном случае необходимо найти ключ длиной $8 \cdot 256 = 2048$ битов, существует его вариантов. Искомый ключ будет обязательно найден в результате перебора в среднем за тестовых операций шифрования. Предположим, что в распоряжении криптоаналитика будет один из самых мощных суперкомпьютеров современности, например, суперкомпьютер Sequoia, занимающий одну из первых строчек в списке TOP500 на начало 2013 года и имеющий производительность 16,32 петафлопс на 1 572 864 ядрах в тесте LINPACK. В результате подсчётов было установлено, что в среднем при шифровании всего вычислительной системе необходимо совершить 163840^* операций с плавающей точкой для нахождения искомого ключа методом «грубой силы». Потому в случае использования указанного суперкомпьютера время, необходимое на вскрытие среднестатистического варианта алгоритма (нахождение искомого ключа), можно приближённо оценить в 10^{604} секунд, что составляет около $3,1688 \cdot 10^{596}$ лет. Полученная оценка свидетельствует о невозможности осуществления атаки методом «грубой силы» на алгоритм CMN89 за разумное время на текущий момент развития вычислительных систем.

Стоит отметить, что решающий вклад в высокую стойкость алгоритма против атак методом «грубой силы» вносит большая длина ключа.

2. Сдвиговая атака в первоначальном виде на алгоритм SMN89 не применима. На каждом из раундов алгоритма весь ключ шифрования используется полностью, однако раунды не являются идентичными за счёт применения выходного «отбеливания» блока данных (которое заключается в изменении и перемещении байта, положение которого в блоке зависит от номера текущего подраунда: номер байта определяется как $kkk \% 64$, где kkk — глобальный номер текущего подраунда); следовательно, в двух подряд выполняемых раундах алгоритма производятся преобразования с байтами на дизъюнктивных множествах позиций.

Получается, что необходимые для проведения атаки соотношения между результирующими байтами после двух последовательных раундов составить не получится. Можно сделать вывод о том, что применение указанного преобразования в некоторой степени повышает криптостойкость алгоритма к этому виду атак.

3. Атака вида «встреча посередине» может быть осуществлена только при известном количестве раундов преобразований, выполненных при зашифровании. Однако стоит отметить, что в отличие от её применения к алгоритму Double DES, когда при первом проходе шифрования одинарным алгоритмом DES используется первая половина секретного ключа, а при втором — вторая, в алгоритме SMN89 на каждом из раундов используется весь ключ. Поэтому выигрыша, получающегося при проведении подобной атаки на алгоритм Double DES (и заключающегося в возможности разбиения ключа на 2 вдвое меньшие по длине части, которые можно в дальнейшем просто перебрать методом «грубой силы»), здесь получить не удастся: уменьшения области перебора не произойдёт.

4. Метод интерполяции служит для нахождения некоторого алгоритма, эквивалентного зашифрованию данных на данном секретном ключе.

Он заключается в построении полинома, связывающего открытый текст и соответствующий ему шифртекст. Процесс построения часто опирается на особенности используемых преобразований. Обычно используется интерполяционная формула Лагранжа:

$$f(x) = \sum_{i=1}^n y_i * \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j},$$

которая обладает следующим свойством: $f(x_i) = y_i$ при всех $i = 1 \dots n$, где x_i и y_i — заданные значения, причём все x_i различны. При криптоанализе качества x_i и y_i берутся соответствующие байты исходного текста и его шифртекста. Для осуществления такой атаки необходимо только n тестовых операций шифрования при наличии n известных открытых текстов, но при условии, что число n ненулевых коэффициентов многочлена не превосходит для рассматриваемого алгоритма.

Исследование преобразований позволяет сделать вывод о том, что самые сложные действия, которые необходимо выполнить криптоаналитику

для вскрытия алгоритма методом интерполяции, заключаются в построении полинома 63-й степени. Его коэффициенты можно найти методом Гаусса при распараллеливании вычислений. Но важно то, что действие построенного полинома будет распространяться исключительно на данный открытый текст, и алгоритм зашифровывания на его основе никак нельзя считать эквивалентным исходному. Так получается, поскольку все вышеуказанные преобразования производятся 16 раз в течение 1 раунда, причём на каждом новом подраунде в них задействован результат наложения того 16-байтного субблока, который не участвовал в преобразованиях на предыдущем подраунде, на результат этих самых преобразований с другим субблоком, и предполагается невозможность внесения изменений в алгоритм.

Изучение конкретных примеров составления полиномов позволило сделать несколько выводов. Во-первых, в общем случае нельзя пытаться уменьшить степень полинома, т.к. все байты отдельно зашифровываемого 64-байтного блока влияют на результат шифрования. Во-вторых, между отдельными байтами открытого текста и шифртекста, занимающими одинаковые позиции (по номеру от начала текста), невозможно установить биективного соответствия: различные байты могут отображаться в один, и наоборот, образ одного и того же байта в разных текстах может быть различным. В-третьих, подтверждено, что использование указанной интерполяционной формулы Лагранжа при вскрытии алгоритма SMN89 возможно только для каждой конкретной пары открытого текста и соответствующего шифртекста, и полученный многочлен в общем случае будет служить аналогом зашифровывания на рассматриваемом ключе только этого открытого текста.

7. Дальнейшее развитие и исследования

В настоящее время большой интерес представляет создание алгоритмов шифрования, структура преобразований которых сочетает в себе несколько видов, таких, например, как сеть Фейстеля, «квадрат», подстановочно-перестановочная сеть, и их исследование на предмет выявления уязвимостей в плане криптостойкости, а также приспособляемости к различным параллельным вычислительным системам.

Структура алгоритма SMN89 представляет собой как раз такую комбинацию, а именно, модификацию сети Фейстеля, на каждом раунде которой выполняются подстановки, перестановки и другие преобразования с блоками данных, представленными в виде квадратного массива байтов.

На данный момент описанный в работе алгоритм шифрования ещё недостаточно изучен. В будущем желательно провести исследование его криптостойкости к линейному криптоанализу, дифференциальному криптоана-

лизу и его усилению — методу бумеранга. Также возможно приспособление программных реализаций алгоритма к конкретным параллельным архитектурам и проведение подробного анализа их эффективности.

Литература

1. *Панасенко С.П.* Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009. — 576 с.: ил.
 2. *Винокуров А.* Алгоритм шифрования ГОСТ 28147-89, его использование и реализация для компьютеров платформы Intel x86.
 3. *Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T.* Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms. <http://www.mirrors.wiretapped.net> – July 13, 2000.
 4. *Jakobsen T., Knudsen L.R.* The Interpolation Attack on Block Ciphers <http://citeseer.ist.psu.edu> — 1997.
-

ОСОБЕННОСТИ РЕАЛИЗАЦИИ НОВОГО ВАРИАНТА ВЭЙВЛЕТНОГО РАЗЛОЖЕНИЯ

Рудаков С.А.

*студент 543 гр. кафедры параллельного программирования СПбГУ,
sergei.rudakov@gmail.com*

Аннотация: Данная работа освещает возможности и особенности применения нового варианта вэйвлетного разложения. Особое внимание уделяется реализации алгоритма в вопросе сжатия изображений.

Введение

В современном мире, где объемы информации неуклонно растут и требуется все большая скорость передачи данных по каналам связей, разработка эффективных алгоритмов сжатия данных становится особенно важна. Одним семейством таких алгоритмов являются методы, основанные на разложении сжимаемого потока на основной поток, меньший по отношению к исходному, и уточняющий вэйвлетный поток, значения которого обычно достаточно малы и требуют меньшего объема памяти для хранения; иногда вэйвлетный поток отбрасывается вовсе. Далее речь пойдет о конкретном алгоритме такого разложения, предложенного в работах [1] — [2].

Вэйвлетное разложение пространств сплайнов

Рассмотрим упорядоченную сетку узлов:

$$X = \{x_j\}_{j \in Z} : \dots < x_{j-1} < x_j < x_{j+1} < \dots$$

$$x_{j+\frac{1}{2}} = (x_j + x_{j+1}) / 2$$

На этой сетке задаются сплайн-функции:

$$\begin{aligned} \omega_{j,2,X}(x) &= \frac{(x-x_j)^2}{(x_{j+2}-x_j)(x_{j+1}-x_j)} \omega_{j,0,X}(x) + \frac{(x_{j+3}-x)^2}{(x_{j+3}-x_{j+1})(x_{j+3}-x_{j+2})} \omega_{j+2,0,X}(x) + \\ &+ \left(\frac{(x-x_j)(x_{j+2}-x)}{(x_{j+2}-x_j)(x_{j+2}-x_{j+1})} + \frac{(x-x_{j+1})(x_{j+3}-x)}{(x_{j+2}-x_{j+1})(x_{j+3}-x_{j+1})} \right) \omega_{j+1,0,X}(x) \\ \omega_{j,0,X}(x) &= \begin{cases} 1, & x_j < x < x_{j+1}, \\ 0, & x \leq x_j, x \geq x_{j+1} \end{cases} \end{aligned}$$

Их линейные комбинации образуют пространство сплайнов:

$$S_x = \left\{ u \mid u = \sum_{j \in Z} c_j \omega_{j,2,X}, \forall c_j \in R \right\}$$

Задается также оператор проектирования непрерывных функций на это пространство:

$$Pf = \sum_{j \in Z} \langle \lambda_j, f \rangle \omega_{j,2,X}$$

$$\langle \lambda_j, f \rangle = 2f(x_{j+\frac{1}{2}}) - (f(x_{j+1}) + f(x_{j+2}))/2$$

Рассматриваемый метод дает возможность реализовать разложение пространства сплайнов, построенных относительно одной сетки, в прямую сумму двух пространств: пространство вэйвлетов и пространство сплайнов, построенных относительно той же сетки без одного узла.

Обозначим за Y более мелкую сетку, а за X — более крупную, не содержащую $(k+1)$ -го узла Y :

$$Y : \dots < x_k < \varepsilon < x_{k+1} < \dots$$

$$S_y = S_x \oplus W$$

Тогда декомпозицию и реконструкцию пространства сплайнов относительно Y можно построить в соответствии со следующими формулами:

Декомпозиция

$$a_i = d_i \quad \forall i \leq k-2, \quad a_i = d_{i+1} \quad \forall i \geq k,$$

$$a_{k-1} = \sum_{j=k-2}^{k+1} q_{k-1,j} d_j,$$

$$b_j = 0, \quad j \neq k-1, \quad k,$$

$$b_{k-1} = d_{k-1} - p_{k-2,k-1} a_{k-2} - p_{k-1,k-1} a_{k-1},$$

$$b_k = d_k - p_{k-1,k} a_k - p_{k,k} a_k$$

Реконструкция

$$d_i = a_i \quad \forall i \leq k-2, \quad d_i = a_{i-1} \quad \forall i \geq k+1,$$

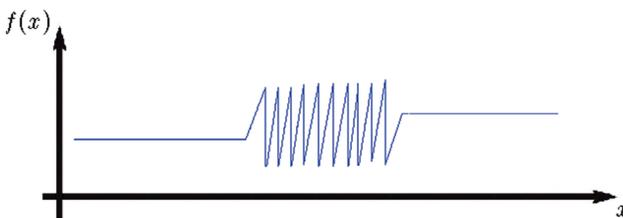
$$d_{k-1} = b_{k-1} + p_{k-2,k-1} a_{k-2} + p_{k-1,k-1} a_{k-1},$$

$$d_k = b_k + p_{k-1,k} a_{k-1} + p_{k,k} a_k$$

$$q_{i,j} = \langle \lambda_i, \omega_{j,2,Y} \rangle$$

$$\omega_{i,2,X}(x) = \sum_{j \in Z} p_{i,j} \omega_{j,2,Y}(x)$$

Здесь d — координаты потока в пространстве S_y , a — в пространстве S_x и b — в пространстве вэйвлетов W . И такое разложение можно проводить рекурсивно, последовательно удаляя узлы сетки. При каждой декомпозиции формируется набор координат a , на единицу меньше входного набора d , а так же две ненулевые координаты b . Для целей сжатия все b обычно полагаются нулевыми. Таким образом если мы хотим аппроксимировать функцию небольшой таблицей ее значений, то сначала разбиваем область аппроксимации достаточно мелкой сеткой (обычно равномерной), затем инициализируем вектор d значениями рассматриваемой функции в узлах сетки. Далее каким-то образом выбираются узлы для отбраковки и последовательно проводится декомпозиция, удаляя по одному из выбранных узлов. Важно заметить, что данный метод не требует равномерности сеток, что является очевидным достоинством, если аппроксимируемая функция на некотором участке быстро осциллирует (см. приводимый график).



В данном случае равномерной сеткой не получится убить двух зайцев сразу — и добиться достаточно точной аппроксимации и оставить минимум значений. Для эффективного сжатия тут следовало бы взять мелкую сетку на участке быстрой осцилляции, а на участках медленного изменения — крупную. Однако есть в неравномерности и свой недостаток: на этапе реконструкции не достаточно знать шага текущей и целевой сетки, необходимо знать все оставшиеся значения сетки. Встает вопрос как определить, какие значения следует удалить.

Методы разреживания сетки

Можно воспользоваться формулой, предложенной в работе [1].

Она позволяет вычислить число, на 1 большее допустимого количества выкидываемых подряд узлов, после сохраненного узла с индексом i , учитывая желаемую степень разряжаемости и максимальное количество выбрасываемых за раз узлов.

$$\bar{S}^*(f', i, q, P, g, \bar{C}) \stackrel{\text{def}}{=} \left\lfloor \frac{\bar{C}}{P \frac{1}{q} \sum_{i'=i}^{i+k-1} |f'(x_{i'})| + \bar{C}/q} \right\rfloor$$

где

i — номер рассматриваемого узла x_i сетки,

P — относительное количество сохраняемых узлов ($0 < P \leq 1$),

q — параметр усреднения (количество усредняемых значений $|f'(x_i)|$ в соседних узлах, $1 \leq q \leq N - i$,

g — параметр ($1 \leq g \leq N - 2$), характеризующий максимальное количество одновременно выбрасываемых узлов (это количество равно $\lfloor g/P \rfloor$)

$$\bar{C} \approx \frac{1}{N} \sum_{i'=0}^{N-1} |f'(x_{i'})|$$

f — аппроксимируемая функция.

В случае числового потока в качестве f можно взять функцию из пространства сплайнов, построенных по текущей сетке с коэффициентами, соответствующими данному потоку. Однако практика показывает, что не всегда подобный подход приводит к желаемому результату. Альтернативой может служить метод, являющийся по сути прямым перебором вариантов того, какой узел стоит выкинуть. А именно, проходя по всем узлам, с начала или с конца сетки проводится декомпозиция, соответствующая удалению этого узла, далее реконструкция и затем замеряются отклонения получившегося вектора от изначального. Если отклонения приемлемы, то узел удаляется, а с оставшимся алгоритм повторяется. И так до тех пор пока находится такой узел, который может быть удален при наших условиях приемлемости. Они могут быть какими угодно в зависимости от области применения — можно ограничить максимальное допустимое отклонение, можно ограничить среднее отклонение. Можно на каждом шаге измерять отклонения, соответствующие удалению каждого узла и удалять в итоге тот, при котором они были минимальны. Плюсы и минусы данного подхода довольно ясны — с одной стороны мы можем гарантировать определенную степень точности и разрежение в рамках этой точности будет близким к оптимальному, а с другой стороны значительно снижается скорость итоговой декомпозиции. Поэтому

использовать такой метод в областях, где важна скорость сжатия навряд ли будет возможно. Один же из вопросов, где это не столь принципиально — сжатие изображений. Здесь единожды проводится операция сжатия, данные записываются в файл и затем могут оттуда многократно считываться. Поэтому тут важнее скорость реконструкции, которая никак не зависит от метода разреживания сетки.

Сжатие изображений

Изображение можно рассматривать как матрицу пикселей, соержащих три значения составляющих цвета и иногда альфа-значение, характеризующее прозрачность. Первым шагом будет вытягивание матрицы в вектор. При этом изображение можно рассматривать целиком или разбить его на блоки определенного размера. В формате JPEG, например, берутся блоки 8×8 пикселей. Вытягивать лучше, не располагая последовательно друг за другом строки или столбцы, а змейкой с тем, чтобы минимизировать количество резких скачков значений. После данного шага мы имеем по сути три последовательности целых чисел, соответствующих цветовым составляющим в какой либо цветовой модели. Это может быть RGB, YCrCb или другая. Часто предпочтение отдается именно модели YCrCb в силу того, что в ней Y-составляющая является наиболее важной и при достаточно сильном сжатии потоков Cг и Cб, качество изображения страдает не сильно. Также может присутствовать четвертый поток при наличии альфа-канала. Над этими потоками и будут проделаны рассмотренные ранее операции.

Вопрос хранения сеток

Вспомним еще раз, что в итоге мы можем получить значения на равномерной сетке, и для реконструкции необходимо знать, какие узлы в этой сетке остались. В виду этого встает вопрос — стоит ли делать свои отдельные сетки для каждого потока составляющих цвета или лучше объединить их сетки в одну общую? Универсального ответа нет. В разных ситуациях оптимальными будут разные решения. В случае когда рассматриваемые сетки имеют достаточное количество общих узлов — выгоднее будет их объединить, если же они почти не пересекаются — лучше хранить по отдельности. Можно выбрать один из этих вариантов, который лучше подходит в большинстве случаев, и использовать его всегда, считая излишки других случаев незначительными. С другой стороны можно каждый раз в конце проводить анализ, по какому пути будет лучше пойти в данной конкретной ситуации.

Заключение

Алгоритмы сжатия, основанные на вэйвлетном разложении с неравномерной сеткой, на текущий момент изучены неполностью, однако при должном внимании к их особенностям имеют большой потенциал успешного использования для сжатия информации.

Литература

1. *Ю.К. Демьянович, В.А. Ходаковский.* Введение в теорию вэйвлетов // Курс лекций ПГУПС — 2007.
 2. *Ю.К. Демьянович, М.В. С. Габр.* Новый вариант вэйвлетного разложения пространств сплайнов // Вестник СПбГУ, серия 10 выпуск — 2009 — С. 58–69.
-

О ЛОКАЛЬНОМ УКРУПНЕНИИ ТРИАНГУЛЯЦИИ

Романовский Л.М.

*аспирант кафедры параллельного программирования
математико-механического факультета СПбГУ, lromanovskiy@gmail.com*

Аннотация: В данной работе рассмотрен алгоритм, позволяющий проводить локальное укрупнение правильной триангуляции на двумерной плоскости с сохранением правильности. Реализована программа, позволяющая курантовскую аппроксимацию поверхности на исходной триангуляции заменить курантовской аппроксимацией поверхности на локально укрупненной триангуляции с учетом особенностей поверхности.

1. Введение

Существующие алгоритмы позволяют строить аппроксимации одномерных функций на неравномерной сетке узлов, что дает возможность учитывать локальные особенности входных данных для последующего сжатия с сохранением допустимой погрешности. Однако, в случае обработки двумерных данных (например, карт высот, 3D-поверхностей) существующие методы не позволяют учитывать локальные особенности данных по нескольким измерениям одновременно. Вместо этого обычно предлагается растягивать входной многомерный поток данных в одномерный, а затем строить по нему аппроксимацию. Такой подход применяется, например, в алгоритме сжатия изображений jpeg. В данной работе рассматривается алгоритм, позволяющий учитывать локальные особенности входного потока данных сразу по двум измерениям, достигая, таким образом, большей эффективности сжатия (см. [1]).

Цель данной работы состоит в реализации программы, позволяющей курантовскую аппроксимацию поверхности на исходной триангуляции заменить на курантовскую аппроксимацию поверхности на локально укрупненной триангуляции с учетом особенностей поверхности. В предлагаемой статье даны краткое описание алгоритма, его реализации на языке программирования Java, атак же некоторые численные результаты работы программы.

2. Основные обозначения

В плоскости R^2 рассматривается декартова система координат (x, y) , точкам этой плоскости сопоставляются радиус-векторы $r = (x, y)$. Треугольник, возможно, криволинейный, с вершинами a, b, c обозначается $|abc|$; порядок следования вершин в этом обозначении роли не играет.

Несколько треугольников объединяются в таблицу, называемую таблицей инциденций. Например, для треугольников $|a, b, c|$, $|d, e, f|$ и $|u, v, w|$ получаем следующую таблицу инциденций:

a	b	c
d	e	f
u	v	w

Порядок следования строк в данном обозначении роли не играет.

3. Триангуляция и ее укрупнение

В плоскости R^2 рассмотрим правильную (возможно, криволинейную) триангуляцию T_0 , которая характеризуется таблицей инциденций

$r_{2i, 2j}$	$r_{2i, 2j+2}$	$r_{2i+1, 2j+1}$
$r_{2i, 2j}$	$r_{2i+2, 2j}$	$r_{2i+1, 2j+1}$
$r_{2i+2, 2j+2}$	$r_{2i+2, 2j}$	$r_{2i+1, 2j+1}$

$$\forall (i, j) \in Z^2$$

Всякую прямолинейную триангуляцию, нульмерный остов которой содержится в множестве вида $\{(ih, jh), (i, j) \in Z^2\}$ называем стандартной триангуляцией.

Укрупнение триангуляции будем проводить объединением двух треугольников исходной триангуляции, имеющих общее ребро, при этом пары для объединения необходимо выбирать таким образом, чтобы топология триангуляции в области укрупнения была аналогична топологии исходной триангуляции. Эквивалентное преобразование таблицы инциденций заключается в том, что из нее исключаются строки, соответствующие объединяемым треугольникам и добавляются строки, соответствующие результатам такого объединения — укрупненным треугольникам. Еще раз подчеркнем, что порядок строк в таблице инциденций не играет никакой роли, поэтому удаление и добавление строк можно осуществлять в любых местах таблицы инциденций. Таким образом, для задания укрупнения достаточно перечислить выбрасываемые и добавляемые строки.

В [1] описан способ локального укрупнения триангуляции T_0 , который приводит к правильной триангуляции T_1 , имеющей в зоне укрупнения прежнюю топологическую структуру.

5. Реализация

Алгоритм реализован автором в виде компьютерной программы. Язык программирования — Java, проект выполнен в среде разработки Eclipse IDE. Для отрисовки триангуляции используется стандартная библиотека Swing. Программа реализована в виде двух проектов: TriangulationEnlargement и TriangulationConsole. TriangulationEnlargement содержит всю логику проекта. TriangulationConsole выполняет загрузку, подготовку данных, запуск вычислений и выгрузку просчитанных данных. Такая организация проекта позволяет облегчить интеграцию приложения в другие проекты.

6. Входные данные программы

В качестве входных данных программы было решено взять графические файлы форматов bmp, jpg и png. Хотя алгоритм ориентирован преимущественно на обработку трехмерных объектов, графические файлы можно рассматривать как карту высот, заданную на сетке узлов, соответствующей размерности входного изображения. При этом координаты пиксела принимаются за координаты по осям (x, y) , а значение цвета в 32-х битном формате — за координату по оси z .

7. Выходные данные

Выходными данными программы является матрица инцидентий, соответствующая триангуляции с проведенным на ней локальным укрупнением.

Вывод матрицы инцидентий производится в консоль программы (или любой другой выходной поток). Для удобства тестирования программы была реализована возможность запуска программы без параметров, в этом случае недостающие параметры запрашиваются в консоли.

При запуске программы порождается окно, в котором отрисовывается исходная триангуляция. При нажатии кнопки «е» происходит укрупнение триангуляции в заданной области, при этом автоматически обновляется изображение триангуляции в окне. Треугольники, над которыми было проведено укрупнение, подсвечиваются красным цветом. Данная функциональность была введена для демонстрационных целей.

8. Особенности реализации

Для поиска смежных треугольников на всей триангуляции была реализована отдельная структура, хранящая треугольники в переменной типа хэш-карта (`HashMap<Integer, Trinagle>`) с целочисленным индексом. Это позволяет обеспечить легкий доступ к треугольникам триангуляции, упростить разработку и сделать алгоритм более гибким, не привязываясь к си-

стематизированной индексации вершин. При этом, однако, возрастают накладные расходы на работу с памятью и удалению/вставку элементов.

В качестве критерия, на основе которого принимается решение о возможности проведения укрупнения, был взят косинус угла между вертикалью и нормалью к треугольникам, которые предлагается объединить. Объединение производится только в том случае, если значение косинуса этого угла больше определенного граничного значения. Такой подход требует доработки, так как не позволяет учитывать случаи, когда достаточно ровная поверхность расположена под углом к вертикали

9. Численные результаты

В нижеследующей таблице представлены результаты тестирования программы. Следует отметить, что процесс вычисления сильно зависит от входных данных, поэтому результаты для тестов одинакового размера могут существенно отличаться.

Размеры	Исходное кол-во треугольников	Кол-во треугольников после укрупнения	Время (миллисекунды)
32×32	1 922	1 826	57
128×100	25 146	20 662	8 841
128×256	64 770	55 954	113 203
16×16	450	402	1
312×456	283 010	233 052	2 130 250
336×188	125 290	102 106	545 793
336×188	125 290	63 686	419 530
200×320	126 962	99 628	497 384
200×320	126 962	69 118	461 204
240×348	165 866	163 410	726 765

10. Заключение

В данной работе построен алгоритм, позволяющий проводить локальное укрупнение правильной триангуляции на двумерной плоскости с сохранением правильности, а также реализована программа, позволяющая курантовскую аппроксимацию поверхности на исходной триангуляции заменить курантовской аппроксимацией поверхности на локально укрупненной с учетом особенностей поверхности триангуляции.

Литература

1. Ю.К. Демьянович, Л.М. Романовский, Локальное укрупнение триангуляции и двумерные сплайн-вэйвлеты. Санкт-Петербург, ВВМ, 2012.

РАСПАРАЛЛЕЛИВАНИЕ СПЛАЙН-ВЭЙВЛЕТНЫХ РАЗЛОЖЕНИЙ

В.В. Петров

*студент 543 группы математико-механического факультета,
кафедра параллельных алгоритмов;
m08pvv@ya.ru*

Санкт-Петербургский государственный университет

Аннотация: Рассматривается общая структура сплайн-вэйвлетного разложения числового потока, исследуются вопросы его распараллеливания и программной реализации. Даны оценки ускорения при распараллеливании.

Введение

Современные потоки информации (которые необходимо обрабатывать, хранить и передавать по коммуникационным средам) имеют электронную форму: чаще всего это поток двоичных данных большого объема. Такие потоки можно быстро обрабатывать лишь в случае, когда имеются большие компьютерные ресурсы (быстродействие, память, широкополосные каналы связи). Задача сокращения объемов цифровой информации за счет отбрасывания несущественных ее составляющих весьма актуальна.

На первом месте среди средств решения этой задачи несомненно находятся вэйвлеты, что подтверждается большим числом приложений в различных технических и научных областях. Имеется много работ по вэйвлетной тематике, в том числе, ряд монографий, исследующих вопрос с различных точек зрения (см. работы [5–7] и имеющуюся в них библиографию).

Цель работы состоит в анализе возможности распараллеливания и в компьютерной реализации алгоритма, описанного в [2–3].

Сплайн-вэйвлетное разложение потока

Имея заданную сетку и вектор-функцию $\phi(t)$ мы можем построить соответствующие сплайны $\omega_j(t)$ с использованием аппроксимационных соотношений.

При $\phi(t) = (1, t, t^2)^T$ имеем

$$\omega_j^B(t) = \frac{(t - x_j)^2}{(x_j - x_{j+1})(x_j - x_{j+2})}, \quad t \in [x_j, x_{j+1}),$$

$$\omega_j^B(t) = ((x_j + x_{j+1} - x_{j+2} - x_{j+3})t^2 - 2(x_j x_{j+1} - x_{j+2} x_{j+3}))t +$$

$$\begin{aligned}
& + x_j x_{j+1} x_{j+2} + x_j x_{j+1} x_{j+3} - x_j x_{j+2} x_{j+3} - x_{j+1} x_{j+2} x_{j+3}) \times \\
& \times (x_{j+2} - x_j)^{-1} (x_{j+2} - x_{j+1})^{-1} (x_{j+3} - x_{j+1})^{-1}, \quad t \in [x_{j+1}, x_{j+2}), \\
& \omega_j^B(t) = \frac{(t - x_{j+3})^2}{(x_{j+3} - x_{j+1})(x_{j+3} - x_{j+2})}, \quad t \in [x_{j+2}, x_{j+3}), \\
& \omega_j^B(t) = 0, \text{ в остальных случаях}
\end{aligned}$$

Введём линейные функционалы $g^{(i)}$, определяемые на непрерывно дифференцируемых функциях по формуле

$$\langle g^{(i)}, u \rangle = u(x_{i+1}) + (x_{i+2} - x_{i+1}) \frac{u'(x_{i+1})}{2}.$$

Данная система функционалов $\{g^{(i)}\}_{i \in \mathbb{Z}}$ биортогональна системе функций $\{\omega_j\}_{j \in \mathbb{Z}}$. Удалив из исходной сетки некоторый набор узлов, получаем новую укрупнённую сетку \tilde{X} . На этой новой сетке сплайны задаются теми же формулами, но с заменой узлов на узлы новой сетки.

Обозначим $P^B(X)$ пространство, являющееся линейной оболочкой функций $\omega_j(t)$

$$P^B(X) = \left\{ \tilde{u} \mid \tilde{u} = \sum_j c_j \omega_j^B \quad \forall c_j \in \mathbb{R}^1 \right\}.$$

Это пространство будем называть пространством B -сплайнов второй степени на сетке X .

Соответственно, $P^B(\tilde{X})$ будем называть пространством B -сплайнов второй степени на сетке \tilde{X} . Справедливо включение $P^B(\tilde{X}) \subset P^B(X)$. Рассмотрим оператор P проектирования пространства $P^B(X)$ на подпространство $P^B(\tilde{X})$, задаваемый формулой

$$P\tilde{u} = \sum_j \langle \tilde{g}^{(i)}, \tilde{u} \rangle \tilde{\omega}_j^B \quad \forall \tilde{u} \in P(X),$$

и введём оператор $Q = I - P$, где I — тождественный оператор. Пространством вэйвлетов здесь является пространство $\tilde{W} = QP^B(X)$, откуда получаем прямое разложение

$$P^B(X) = P^B(\tilde{X}) \oplus \tilde{W},$$

что является сплайн-вэйвлетным разложением пространства $P^B(X)$.

Разложение потока яркости (изображение)

Пусть имеется поток значений яркости монохромного изображения, который задаёт функцию $v(t)$, представляющую собой яркость соответствующего пикселя. Из этой функции получим поток c_i по формуле

$$c_i = \langle g^{(i)}, v \rangle = v(x_{i+1}) + \frac{v'(x_{i+1})}{2} (x_{i+2} - x_{i+1}).$$

Используя формулы численного дифференцирования, получаем

$$c_i = \langle g^{(i)}, v \rangle = \frac{v(x_{i+2}) + v(x_{i+1})}{2}.$$

В качестве функции $u(t)$ возьмём следующую функцию:

$$u(t) = \sum_{x_j \in X} c_j \omega_j(t),$$

где c_i — элементы потока, а X — исходная сетка.

Проекция функции $u(t)$, соответствующая укрупнённой сетке \tilde{X} , будет иметь вид

$$\tilde{u}(t) = \sum_{\tilde{x}_j \in \tilde{X}} \langle \tilde{g}^{(i)}, u \rangle \tilde{\omega}_i(t).$$

Используя определение функционалов $g^{(i)}$, получаем

$$\tilde{u}(t) = \sum_{\tilde{x}_j \in \tilde{X}} a_i \tilde{\omega}_i(t) = \sum_{\tilde{x}_j \in \tilde{X}} \tilde{\omega}_i(t) \sum_{x_j \in X} c_j \left(\omega_j(\tilde{x}_{i+1}) + (\tilde{x}_{i+2} - \tilde{x}_{i+1}) \frac{v'(\tilde{x}_{i+1})}{2} \right),$$

Откуда находятся a_i , являющиеся основным потоком. Вэйвлетный поток получаем по формуле

$$b_s = \langle g_s, u - \tilde{u} \rangle = c_s - \langle g_s, \tilde{u} \rangle = c_s - \sum_{\tilde{x}_j \in \tilde{X}} a_j \left(\tilde{\omega}_i(x_{s+1}) + (x_{s+2} - x_{s+1}) \frac{\tilde{\omega}'_i(x_{s+1})}{2} \right).$$

Из этой формулы также получаем формулы для восстановления исходного потока по полученному разложению, используя формулу

$$\tilde{u}(t) = \sum_{\tilde{x}_j \in \tilde{X}} a_i \tilde{\omega}_i(t).$$

Возможности для распараллеливания

При изложенной последовательности вычислений имеется зависимость по данным: для вычисления вэйвлетного потока необходимо знать значения основного потока.

Значения основного потока можно вычислять независимо, так как они зависят только от значений входного потока с соответствующими индексами. Входной поток обычно достаточно большой для того чтобы обработать его целиком параллельно, соответственно мы можем разбить его на блоки, у которых сначала будет вычисляться основной поток, а затем вэйвлетный, по мере поступления вычисленных значений основного потока.

Вычисления элементов основного и вэйвлетного потоков по соответствующим формулам декомпозиции, а также вычисления элементов исходного потока по формулам реконструкции не зависят друг от друга при различных значениях фигурирующих в них индексов, что позволяет достигнуть высокой степени параллелизма при компьютерной реализации сплайн-вэйвлетных разложений на различных параллельных архитектурах.

Вычислительная сложность каждого вычисления по формулам декомпозиции и реконструкции не зависит от индексов, что облегчает достижение равномерной загрузки.

В системе с общей памятью мы можем достичь равномерной загрузки всех имеющихся процессоров (ядер) путём разбиения входного потока на блоки такого размера, что их можно будет обработать одновременно, при этом основная потеря производительности будет приходиться на работу с памятью и промахи кэша; минимизировав промахи кэша и оптимизировав работу с памятью, мы можем достичь линейного ускорения.

В системах с распределенной памятью мы можем достичь равномерной загрузки всех вычислительных узлов путём разбиения входного потока на блоки, каждый из которых будет обрабатываться отдельным узлом. В такой ситуации потребуется либо дополнительная память для хранения граничных элементов, либо дополнительные пересылки данных. Используя дополнительную память и избегая лишних пересылок данных, мы опять получаем линейное ускорение.

Вариант распараллеливания

Рассмотрим распараллеливание сплайн-вэйвлетного разложения цветного изображения, состоящего из трёх цветовых плоскостей (красной, зелёной и синей) на системе с общей памятью.

Разложение каждой цветовой плоскости не зависит от другой и эти три процесса вычисления можно выполнять параллельно и они имеют примерно равную вычислительную сложность (а если сетки не отличаются, то равную).

При последовательных вычислениях имеем

$$t_{RGB} = t_r + t_g + t_b.$$

При параллельных (если потоков три и более)

$$t_{RGB} = \text{MAX}(t_r, t_g, t_b) + t_{\text{накл.}}$$

где $t_{\text{накл.}}$ — время накладных расходов на распараллеливание

Вычисления элементов основного потока состоят из независимых сумм

$$a_i = \sum_{x_j \in X} c_j \left(\omega_j(\tilde{x}_{i+1}) + (\tilde{x}_{i+2} - \tilde{x}_{i+1}) \frac{\omega'_j(\tilde{x}_{i+1})}{2} \right).$$

Если количество элементов сетки \tilde{X} равно \tilde{N} , а время вычисления одного a_i равно t_{a_i} , то

$$\text{последовательно: } t_{\text{main}} = \tilde{N}t_{a_i}$$

$$\text{параллельно в } p_1 \text{ потоков: } t_{\text{main}} = \frac{\tilde{N}t_{a_i}}{p_1} + t_{\text{накл.}},$$

где $t_{\text{накл.}}$ — накладные расходы на распараллеливание.

Полученное ускорение будет

$$\frac{\tilde{N}t_{a_i}}{\frac{\tilde{N}t_{a_i}}{p_1} + t_{\text{накл.}}} = \frac{p_1}{1 + \frac{t_{\text{накл.}}}{\tilde{N}t_{a_i}}},$$

что весьма близко к линейному.

Количество элементов сетки X обозначим N , что приводит к следующим оценкам возможности распараллеливания вычисления отдельных элементов основного потока a_i

$$\text{последовательно: } t_{a_i} = Nt_{elem}$$

$$t_{elem} = (2t\omega_j + t\omega'_j + t_{mul} + t_{add} + t_{sub}).$$

В силу того что функция $\omega_j(t)$ имеет малый носитель, мы можем не считать по всей сетке, а ограничиться только носителем. Тогда время выполнения вычисления одного элемента основного потока будет выражаться формулами:

$$\text{последовательно: } t_{a_i} = N_{supp} t_{elem}$$

$$t_{elem} = (2t\omega_j + t\omega'_j + t_{mul} + t_{add} + t_{sub})$$

$$\text{параллельно в } p_2 \text{ потоков: } t_{a_i} = \frac{N_{supp} t_{elem}}{p_2} + t_{накл.},$$

где N_{supp} — количество элементов в носителе $\omega_j(t)$, t_{elem} — время вычисления одного слагаемого суммы.

При таком компактном носителе как у сплайнов с минимальным носителем, накладные расходы на распараллеливание будет существенно по сравнению со временем вычисления, кроме случаев достаточно сложной функции $\omega_j(t)$.

Вычисления вэйвлетного потока зависят от вычислений основного, причём мы также можем суммировать не по всей сетке для каждого элемента, а только по элементам носителя функции $\tilde{\omega}_i(t)$.

Время вычисления элементов вэйвлетного потока:

$$\text{последовательно: } t_{b_i} = N_{supp} t_{elem} + t_{sub}$$

$$t_{elem} = (2t\omega_j + t\omega'_j + t_{mul} + t_{add} + t_{sub})$$

$$\text{параллельно в } p_2 \text{ потоков: } t_{a_i} = \frac{N_{supp} t_{elem} + t_{sub}}{p_2} + t_{накл.},$$

где N_{supp} — количество элементов в носителе $\omega_j(t)$, t_{elem} — время вычисления одного слагаемого суммы.

Время вычисления всего вэйвлетного потока:

$$\text{последовательно: } t_{wave} = \tilde{N} t_{b_i}$$

$$\text{параллельно в } p_1 \text{ потоков: } t_{wave} = \frac{\tilde{N} t_{b_i}}{p_1} + t_{накл.},$$

где $t_{накл.}$ — накладные расходы на распараллеливание, t_{b_i} — время вычисления одного элемента вэйвлетного потока.

Полученное ускорение будет

$$\frac{\tilde{N}t_{b_i}}{\frac{\tilde{N}t_{b_i}}{p_1} + t_{\text{накл.}}} = \frac{p_1}{1 + \frac{t_{\text{накл.}}}{\tilde{N}t_{b_i}}},$$

что весьма близко к линейному.

Если пренебречь накладными расходами на распараллеливание, то вычисления основного и вэйвлетного потоков можно распараллелить с линейным ускорением.

Заключение

В данной работе рассматривается общая структура вэйвлетного разложения числового потока, построенного с использованием вложенных сплайновых пространств и специального проектирования. Здесь изучается возможность распараллеливания и даётся оценка ускорения.

Сплайн-вэйвлетные разложения можно применять к числовым потокам большого объёма, эффективно распараллеливая как на системах с общей памятью, так и на системах с распределённой памятью. В случае если накладные расходы на распараллеливание достаточно малы, то ускорение линейно.

Приложение 1. Графики ускорения

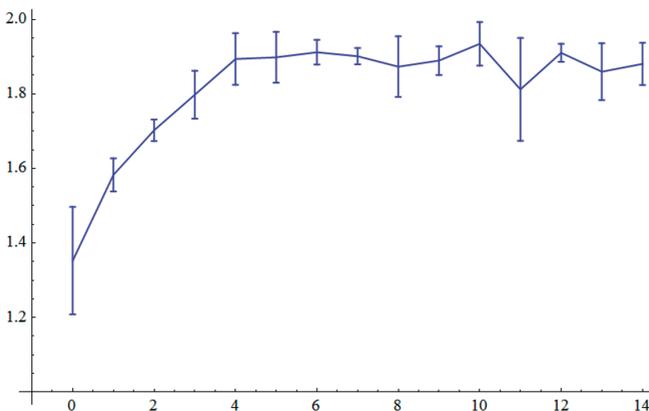


Рис. 1. Зависимость ускорения от размера подзадачи при использовании двух процессорных ядер

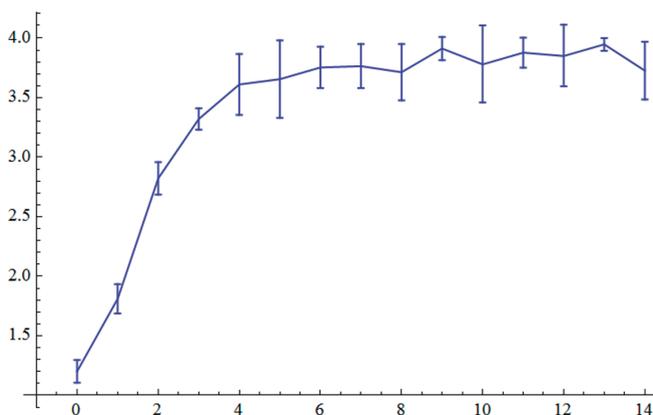


Рис. 2. Зависимость ускорения от размера подзадачи при использовании четырёх процессорных ядер

Литература

1. Демьянович Ю.К. Вэйвлеты & минимальные сплайны. 2003.
2. Демьянович Ю.К., Ходаковский В.А. Введение в вэйвлеты. 2007
3. Демьянович Ю.К., Иванцова О.Н. Новые представления сплайн-вэйвлетных разложений. Ж. Проблемы математического анализа № 46, с. 73–104 (2010).
4. Макаров А.А. Матрицы добавления и удаления узлов для неполиномиальных сплайнов // Вычислительные методы и программирование. 2012. Т. 13. С. 74–86.
5. Добеши И. Десять лекций по вэйвлетам. М.-И. 2004. 464 с.
6. Чуи К. Введение в вэйвлеты. М. 2001. 412 с.
7. Малла С. Вэйвлеты в обработке сигналов. М. 2005. 671 с.

ТРИАНГУЛЯЦИЯ ОБЛАСТИ, ЕЁ ЛОКАЛЬНОЕ УКРУПНЕНИЕ И ИСПОЛЬЗОВАНИЕ

Парахин М.В.

*студент математико-механического факультета СПбГУ,
кафедры параллельных алгоритмов, Mikhail.Parakhin@gmail.com*

Аннотация: Рассмотрен алгоритм построения правильной триангуляции, допускающей локальное укрупнение с сохранением правильности. Приведены адаптивные сплайн-вэйвлетные разложения и соответствующие формулы декомпозиции и реконструкции двумерных числовых потоков. Разработана компьютерная программа, реализующая приведённые алгоритмы. Проведён анализ возможностей распараллеливания компьютерной программы.

Введение

Вэйвлетное преобразование уже долгое время используется как эффективный и качественный способ обработки числовых данных путём разложения числового потока на основной (менее плотный, чем исходный, и лучше подходящий для передачи) и вэйвлетный (используемый, если необходимо восстановить исходный без потери качества). При обработке изображений, когда нет необходимости получать изображение в мельчайших деталях, хорошие результаты дают адаптивные сплайн-вэйвлетные преобразования. Правильным подходом будет построить правильную триангуляцию области с целью дальнейшего локального укрупнения. Затем используются аппроксимационные соотношения, вложенность сплайновых пространств и их проектирование с помощью биортогональных систем функционалов. В случае двумерного потока эффективным методом аппроксимации являются функции Куранта, базирующиеся на правильной триангуляции, допускающей локальное укрупнение с сохранением правильности.

Цель данной работы состоит в создании компьютерной программы и анализе возможностей распараллеливания по формулам из работ [1–3].

1. Построение и укрупнение триангуляции

Рассмотрим плоскость $\{t | t = (x, y) \in \mathbb{R}^2\}$. Необходимо построить правильную триангуляцию области, то есть разбить её на треугольники, причём триангуляция называется правильной, если любая вершина треугольника не лежит внутри стороны другого треугольника. Для описания триангуляции представим её в виде таблицы инцидентий, где каждой строке соответствует один треугольник, а столбцам — координаты вершин.

Введем обозначение $\mathbb{R}^2 = \{(i, j) \mid i, j \in \mathbb{Z}\}$. Пусть $M_{i,j} = (i, j)$, где $(i, j) \in \mathbb{Z}^2$. Точки $M_{i,j}$ служат вершинами триангуляции, которая описана таблицей с тремя столбцами и бесконечным числом строк. Бесконечная таблица получена объединением таблиц:

$$\left\| \begin{array}{ccc} M_{2i,2j} & M_{2i-2,2j} & M_{2i-1,2j+1} \\ M_{2i-2,2j+2} & M_{2i-2,2j} & M_{2i-1,2j+1} \\ M_{-2+2i,2+2j} & M_{2i,2+2j} & M_{2i-1,2j+1} \\ M_{2i,2j} & M_{2i,2j+2} & M_{2i-1,2j+1} \end{array} \right\|, \quad \text{где } (i, j) \in \mathbb{Z}^2 \quad (1.1)$$

Укрупнение триангуляции будем проводить объединением двух соседних, то есть имеющих общую сторону, треугольников. Полученные в результате треугольники будем называть укрупнёнными треугольниками.

Рассмотрим такое укрупнение, при котором вершину $M_{0,0}$ будут окружать только укрупнённые треугольники. Для этого заменим перечисленные ниже треугольники на треугольник, получающийся их объединением. Преобразование таблицы инцидентий состоит в том, что из неё исключаются строки, соответствующие результатам такого объединения — укрупнённым треугольникам. Как было отмечено выше, расположение строк в таблице инцидентий не существенно, и, потому, строки могут быть добавлены между любыми строками упомянутой таблицы. Таким образом, таблицы инцидентий будем задавать указанием двух строк, заменяемых треугольников (в левой части формулы) и указанием строки укрупнённого треугольника (в правой части формулы).

$$\left\| \begin{array}{ccc} M_{0,0} & M_{-2,0} & M_{-1,1} \\ M_{-2,2} & M_{-2,0} & M_{-1,1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{-2,0} \quad M_{-2,2} \right\|, \quad (1.2)$$

$$\left\| \begin{array}{ccc} M_{-2,2} & M_{0,2} & M_{-1,1} \\ M_{0,0} & M_{0,2} & M_{-1,1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{0,2} \quad M_{-2,2} \right\|, \quad (1.3)$$

$$\left\| \begin{array}{ccc} M_{0,0} & M_{0,2} & M_{1,1} \\ M_{2,2} & M_{0,2} & M_{1,1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{0,2} \quad M_{2,2} \right\|, \quad (1.4)$$

$$\left\| \begin{array}{ccc} M_{2,2} & M_{2,0} & M_{1,1} \\ M_{0,0} & M_{2,0} & M_{1,1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{2,0} \quad M_{2,2} \right\|, \quad (1.5)$$

$$\left\| \begin{array}{ccc} M_{0,0} & M_{2,0} & M_{1,-1} \\ M_{2,-2} & M_{2,0} & M_{1,-1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{2,0} \quad M_{2,-2} \right\|, \quad (1.6)$$

$$\left\| \begin{array}{ccc} M_{2,-2} & M_{0,-2} & M_{1,-1} \\ M_{0,0} & M_{0,-2} & M_{1,-1} \end{array} \right\| \rightarrow \left\| M_{0,0} \quad M_{0,-2} \quad M_{2,-2} \right\|, \quad (1.7)$$

$$\left\| \begin{array}{ccc} M_{0,0} & M_{0,-2} & M_{-1,-1} \\ M_{-2,-2} & M_{0,-2} & M_{-1,-1} \end{array} \right\| \rightarrow \left\| \begin{array}{ccc} M_{0,0} & M_{0,-2} & M_{-2,-2} \end{array} \right\|, \quad (1.8)$$

$$\left\| \begin{array}{ccc} M_{-2,-2} & M_{-2,0} & M_{-1,-1} \\ M_{0,0} & M_{-2,0} & M_{-1,-1} \end{array} \right\| \rightarrow \left\| \begin{array}{ccc} M_{0,0} & M_{-2,0} & M_{-2,-2} \end{array} \right\|. \quad (1.9)$$

2. Структура барицентрических звёзд исходной триангуляции

Для построения аппроксимации Куранта важна структура барицентрических звёзд, соответствующих вершинам триангуляции. Для исходной триангуляции имеется два типа барицентрических звёзд. К первому типу отнесём барицентрические звёзды, содержащие восемь треугольников, а ко второму — четыре треугольника. Барицентрические звёзды первого типа имеются у вершин $M_{2i,2j}$. Барицентрические звёзды второго типа имеются у вершин $M_{2i-1,2j+1}$.

3. Структура барицентрических звёзд укрупнённой триангуляции

При локальном укрупнении триангуляции появляются два дополнительных типа барицентрических звёзд. Барицентрическая звезда вершины $M_{0,0}$ состоит из восьми треугольников. Барицентрические звёзды для вершин $M_{0,\pm 2}$ и $M_{\pm 2,0}$ состоят из шести треугольников, а для вершин $M_{\pm 2,2}$ и $M_{\pm 2,-2}$ — из восьми.

4. Функция Куранта

Функция Куранта ассоциированная с выделенной вершиной — называется такая непрерывная функция, равная единице в данной вершине, линейная на каждом треугольнике барицентрической звезды этой вершины и равная нулю вне её. Система функций Куранта — линейно независимая система. Для каждой выделенной вершине $M_{i,j}$ исходной триангуляции введем в соответствие функцию Куранта, обозначенную $\omega_{i,j}$, $(i,j) \in \mathbb{Z}^2$. Как уже говорилось ранее, на исходной триангуляции имеем два типа барицентрических звезд, которым соответствуют два типа функции Куранта: с нечетными индексами, носитель, которых состоит из четырех треугольников и с четными индексами, если носитель состоит из восьми треугольников.

Введём следующее обозначение:

$$E = \{(0,0), (1,1), (-1,1), (1,-1), (-1,-1)\}, E' = E \setminus (0,0),$$

$$2E = \{(0, 0), (2, 2), (-2, 2), (2, -2), (-2, -2)\}.$$

Для укрупнённой триангуляции функцию Куранта обозначим как $\tilde{\omega}_{i,j}(t)$. Заметим, что при укрупнении триангуляции некоторые вершины исходной триангуляции не участвуют: $(i, j) \in \mathbb{Z}^2 \setminus E'$.

Пусть теперь и далее для краткости обозначим некоторые вектора:

$$\alpha \stackrel{\text{def}}{=} (i, j), \theta \stackrel{\text{def}}{=} (0, 0), e \stackrel{\text{def}}{=} (1, 1), e^* \stackrel{\text{def}}{=} (-1, 1).$$

Теорема 4.1. Справедливы следующие соотношения:

$$\tilde{\omega}_\alpha(t) \equiv \sum_{\gamma \in \mathbb{R}^2} p_{\alpha,\gamma} \omega_\gamma(t), \quad \forall \alpha \in \mathbb{Z}^2 \setminus E',$$

где

$$p_{\alpha,\gamma} \stackrel{\text{def}}{=} \delta_{\alpha,\gamma}, \quad \text{при } \alpha \in \mathbb{Z}^2 \setminus E' \setminus 2E', \quad \gamma \in \mathbb{Z}^2,$$

$$p_{2\alpha,2\alpha} = 1, \quad \text{при } \alpha \in E, \quad p_{\theta,\alpha} = 1/2, \quad \text{при } \alpha \in E'.$$

Здесь $\delta_{\alpha,\gamma}$ — символ Кронекера.

Теперь рассмотрим линейные пространства:

$$\mathbb{S} \stackrel{\text{def}}{=} Cl_p \mathcal{L}(\omega_\gamma | \forall \gamma \in \mathbb{Z}^2), \quad \tilde{\mathbb{S}}_\theta \stackrel{\text{def}}{=} Cl_p \mathcal{L}(\tilde{\omega}_\alpha | \forall \alpha \in \mathbb{Z}^2 \setminus E'),$$

где $\mathcal{L}(\dots)$ означает линейную оболочку множества элементов, указанного в круглых скобках, а Cl_p означает замыкание в топологии поточечной сходимости.

Утверждение. Линейное пространство $\tilde{\mathbb{S}}_\theta$ является подпространством в \mathbb{S} :

$$\tilde{\mathbb{S}}_\theta \subset \mathbb{S}.$$

5. Общая структура вэйвлетного разложения

Общая структура вэйвлетного разложения основана на построении оператора проектирования с использованием продолжения системы функционалов, биортогональной координатным функциям вложенного пространства. Зададим в пространстве $C(\mathbb{R}^2)$ систему линейных функционалов g_γ для $\forall \gamma \in \mathbb{Z}$ следующей формулой:

$$\langle g_\gamma, u \rangle \stackrel{\text{def}}{=} u(M_\gamma)$$

Рассмотрим теперь оператор проектирования P_0 пространства $C(\Omega)$ на подпространство $\tilde{\mathbb{S}}_\theta$, задаваемое формулой:

$$P_0 u \stackrel{\text{def}}{=} \sum_{\alpha \in \mathbb{Z}^2 \setminus E'} \langle \tilde{g}_\alpha, u \rangle \tilde{\omega}_\alpha, \quad \forall u \in C(\Omega),$$

и введём оператор $Q_0 = I - P_0$, где I — тождественный оператор в $C(\Omega)$. Пространством вэйвлетов будем называть пространство

$$\mathbb{W}_0 \stackrel{\text{def}}{=} Q_0 \mathbb{S}, \quad (5.1)$$

и мы получаем прямое разложение $\mathbb{S} = \tilde{\mathbb{S}}_\theta \oplus \mathbb{W}_0$ которое называется сплайн-вэйвлетным разложением пространства \mathbb{S} . Пусть $u \in \mathbb{S}$. Приравнявая правые части двух представлений этого элемента,

$$u = \sum_{\gamma \in \mathbb{Z}^2} c_\gamma \omega_\gamma,$$

$$u = \sum_{\alpha \in \mathbb{Z}^2 \setminus E'} a_\alpha \tilde{\omega}_\alpha + \sum_{\beta \in \mathbb{Z}^2} b_\beta \omega_\beta,$$

где $a_\alpha = \langle \tilde{g}_\alpha, u \rangle$, $b_\beta, c_\gamma \in \mathbb{R}^1$, и используя формулы калибровочных соотношений функций Куранта, а также в виду линейной независимости системы $\{\omega_\gamma\}_{\gamma \in \mathbb{Z}^2}$, получим формулу реконструкции:

$$c_\gamma = \sum_{\alpha \in \mathbb{Z}^2 \setminus E'} a_\alpha p_{\alpha, \gamma} + b_\gamma, \quad \forall \gamma \in \mathbb{Z}^2. \quad (5.2)$$

Из формул реконструкции выразим a_α и b_γ , получив тем самым соответствующие формулы декомпозиции:

$$a_\alpha = \sum_{\gamma \in \mathbb{Z}^2} c_\gamma \langle \tilde{g}_\alpha, \omega_\gamma \rangle + b_\gamma, \quad \alpha \in \mathbb{Z}^2 \setminus E', \quad (5.3)$$

$$b_\gamma = c_\gamma - \sum_{\alpha \in \mathbb{Z}^2 \setminus E'} a_\alpha p_{\alpha, \gamma}, \quad \forall \gamma \in \mathbb{Z}^2. \quad (5.4)$$

6. Вэйвлетное разложение при укрупнении триангуляции

Теорема 6.1. При локальном укрупнении триангуляции в вэйвлетном разложении (5.1) формулы декомпозиции (5.3) — (5.4) имеют вид

$$b_\alpha = 0, \quad a_\alpha = c_\alpha, \quad \alpha \in \mathbb{Z}^2 \setminus E', \quad (6.1)$$

$$b_{-e} = c_{-e} - \frac{1}{2} c_{-2e} - \frac{1}{2} c_\theta, \quad b_{-e} = c_{-e} - \frac{1}{2} c_{-2e} - \frac{1}{2} c_\theta, \quad (6.2)$$

$$b_e = c_e - \frac{1}{2}c_{2e} - \frac{1}{2}c_\theta, \quad b_{e^*} = c_{e^*} - \frac{1}{2}c_{2e^*} - \frac{1}{2}c_\theta. \quad (6.3)$$

Теорема 6.2. Для локального укрупнения триангуляции вэйвлетному разложению (5.1) соответствуют формулы реконструкции:

$$c_\alpha = a_\alpha, \quad \forall \alpha \in \mathbb{Z}^2 \setminus E', \quad (6.4)$$

$$c_{-e^*} = b_{-e^*} + \frac{1}{2}a_{-2e^*} + \frac{1}{2}a_\theta, \quad c_{-e} = b_{-e} + \frac{1}{2}a_{-2e} + \frac{1}{2}c_\theta, \quad (6.5)$$

$$c_e = b_e + \frac{1}{2}a_{2e} + \frac{1}{2}a_\theta, \quad c_{e^*} = b_{e^*} + \frac{1}{2}a_{2e^*} + \frac{1}{2}a_\theta. \quad (6.6)$$

7. Компьютерная реализация алгоритма укрупнения триангуляции

Дан исходный двумерный числовой поток, который представляет из себя прямоугольную область с известными значениями в точках $\{(i, j) \mid (i, j) \in \mathbb{Z}^2\}$. Дана область, в которой необходимо провести локальное укрупнение триангуляции, заданная в виде $(c-A)^2 + (y-B)^2 = C^2$.

Шаги алгоритма:

1. Построение триангуляции вида (1.1). Триангуляция хранится в виде массива `Triangles` — список треугольников, содержащий строки, соответствующие различным треугольникам, и три столбца — вершины соответствующих треугольников, и массива `Vertices` — список вершин, содержащий строки соответствующие вершинами и два столбца — координаты вершин.

2. Составление списка вершин первого типа, то есть тех, которым соответствуют барицентрические звёзды из восьми треугольников.

3. Проведение укрупнения для вершин, полностью лежащих в заданной области вместе с соответствующими им барицентрическими звёздами, путём удаления исходных треугольников и добавления новых в массив `Triangles` аналогично формулам (1.2) – (1.9).

4. Разложение исходного потока согласно формулам декомпозиции для укрупнённой триангуляции.

8. Анализ возможностей распараллеливания

Так как укрупнение исходной триангуляции относительно вершин второго типа, то есть вершин, барицентрические звёзды которых состоят из восьми треугольников, укрупнение соответствующих звёзд независимо и можно передавать на обработку различным вычислительным модулям.

Компьютерная реализация сплайн-вейвлетного разложения базируется на формулах декомпозиции и реконструкции и вычислительный процесс по этим формулам может происходить независимо. Значения основного потока можно вычислять независимо, для этого используются только значения исходного потока. Таким образом можно разбить исходный поток на блоки и передавать блоки для обработки различным параллельным вычислительным модулям. Значения вэйвлетного потока используют значения основного, так что его вычисление будет производиться постепенно, по мере поступления значений основного потока.

Заключение

Рассмотрен алгоритм построения правильной триангуляции, допускающей локальное укрупнение с сохранением правильности. Приведены адаптивные сплайн-вейвлетные разложения и соответствующие формулы декомпозиции и реконструкции двумерных числовых потоков. Разработана компьютерная программа, реализующая приведённые алгоритмы. Проведён анализ возможностей распараллеливания компьютерной программы.

Литература

1. *Демьянович Ю.К.* Аппроксимация на многообразии и минимальные сплайны. 1994.
 2. *Арсентьева Е.П., Демьянович Ю.К.* Адаптивные сплайн-вейвлетные разложения двумерных потоков числовой информации. СПб. Проблемы математического анализа. 2011. Вып. 56. С.3–17.
 3. *Романовский Л.М.* Локальное укрупнение триангуляции и калибровочные соотношения. Труды XLII Международной конференции аспирантов и студентов. Санкт-Петербург, 4–7 апреля 2011 года. Изд-во СПбГУ. 2011. С.338–434.
-

ВЕЙВЛЕТНЫЕ ПРЕОБРАЗОВАНИЯ

Т.О. Мухамедов

*студент 543 группы кафедры параллельных алгоритмов, timur@mbrooks.ru
Санкт-Петербургский государственный университет*

Введение

Современные потоки информации в процессе обработки, хранения и передачи представляются в виде последовательности нулей и единиц огромной длины (10^{12} — 10^{16} символов).

Такие последовательности можно быстро обрабатывать лишь в случае, когда имеются большие компьютерные ресурсы (быстродействие, память, мощные каналы связи).

В связи с этим, задача сокращения объема числового потока за счет выявления и отбрасывания несущественных (с той или иной точки зрения) частей — становится весьма актуальной, а степень важности эффективности решения этой задачи постоянно растет.

Обработка числового потока может производиться в двух направлениях:

- в направлении сохранения аппроксимации
- в направлении сохранения основных спектральных характеристик

Классическим аппаратом обработки в первом направлении являются сплайны, а во втором — всплески (за рубежом чаще употребляют термин вэйвлеты).

Оказалось, что сплайны и всплески тесно связаны между собой: каждая цепочка вложенных пространств минимальных сплайнов порождает всплесковое разложение.

Поскольку мощность множества упомянутых цепочек — континуум, то такова же мощность множества предлагаемых всплесковых разложений.

Преобразования числовых потоков

Для наглядной иллюстрации идеи вейвлет-преобразования представим себе, что рассматриваемый числовой поток кодирует некоторое изображение, выводимое на экран компьютера.

Предположим, что экран представляет собой прямоугольную матрицу из большого числа пикселей маленьких прямоугольников которые нанесены на прозрачную поверхность (стекло).

Пиксели светятся под воздействием попадающих на них электронов. Для простоты рассматриваем лишь одноцветные изображения (черно-бе-

лый экран). Пиксели перенумерованы последовательно по строкам, которые предварительно выстроены одна за другой в прямую линию.

Таким образом, пиксели приобретают номера:

$$0, 1, 2, 3, 4, 5, \dots, N-1; N=M \times K,$$

где M — число строк рассматриваемой матрицы, а K — число ее столбцов.

Для определенности будем считать N четным (пусть $N=2L$, где L — натуральное число).

Каждому пикселю предписывается определенная яркость, выражаемая некоторым числом (обозначим это число для j -го пикселя через c_j).

Таким образом, кодировка изображения производится с помощью числового потока:

$$c_0, c_1, c_2, c_3, c_4, c_5, \dots, c_{2L-1} \quad (2.1)$$

Поток (2.1) может быть передан по линиям связи и при подаче на экран компьютера (телевизора) может быть превращен в исходное изображение. Если исходное изображение передается с большой точностью, то N весьма велико, и передача даже одного такого изображения представляет значительные технические трудности (на практике требуется передавать миллионы таких изображений с большой скоростью). Поэтому возникает задача уменьшения количества передаваемых чисел.

Предполагая, что соседние числа в (2.1) близки, можно было бы предположить передавать, например, только числа с нечетными номерами в (2.1), т. е. числа:

$$c_1, c_3, c_5, \dots, c_{2L-1} \quad (2.1)$$

Такое преобразование называется прореживанием исходного числового потока (*английский термин *upsampling* — разрежение или разрежающая выборка*). Вместо потока (2.1) передают в два раза более короткий поток (2.2).

Далее приемное устройство расширяет полученный числовой поток (2.2) дублированием принятых значений так, чтобы в результате на местах с четным и со следующим нечетным номером находились одинаковые числа. В результате на экране воспроизводится изображение, полученное с помощью числового потока вида

$$c_1, c_1, c_3, c_3, c_5, c_5, \dots, c_{2L-1}, c_{2L-1} \quad (2.3)$$

Тем самым восстановление (2.3) исходного потока (2.1) производится с погрешностью, причем информация теряется необратимым образом (*т. е.*

без передачи дополнительной информации приемное устройство, вообще говоря, не в состоянии восстановить поток (2.1)).

Такой прием (английский эквивалент *downsampling* — сужение) оправдан, если полученное изображение мало отличается от исходного.

Недостатки описанного подхода состоят в следующем:

- он применим лишь к достаточно медленно меняющемуся потоку,
- отсутствует учет характеристик числового потока (в некоторых частях числовой поток может меняться очень медленно, и можно было бы выбрасывать много чисел подряд, а в других частях при быстром изменении потока любые выбрасывания чисел могут существенно испортить передаваемое изображение)
- нет средств для уточнения передаваемого потока.

Идея вэйвлетного подхода:

Из числового потока (2.1) формируется два числовых потока

$$a_j = (c_{2j} + c_{2j+1})/2; \quad b_j = (c_{2j} - c_{2j+1})/2, \quad \text{где } j=0, 1, \dots, L-1 \quad (2.4)$$

Нетрудно заметить, что:

$$c_{2j} = a_j + b_j, \quad c_{2j+1} = a_j - b_j, \quad j=0, 1, \dots, L-1 \quad (2.5)$$

Таким образом, если поток (2.1) заменить двумя потоками (2.4), то после их передачи можно восстановить исходный поток (2.1), используя формулы (2.5).

Возникает вопрос, в чем же польза от замены потока (2.1) на два потока (2.4), если общее количество чисел в потоках (2.4) совпадает с количеством чисел в (2.1). Для ответа на этот вопрос заметим, что если соседние числа в (2.1) близки, то второй из потоков в (2.4) состоит из чисел, близких к нулю, так что может оказаться, что второй поток вообще не нужен и его можно отбросить.

Однако, если некоторые фрагменты первого потока из (2.4) не дают достаточной точности, то можно использовать соответствующие фрагменты (с теми же диапазонами индексов) второго потока, и произвести расчеты по формулам (2.5). Это приведет к точному восстановлению исходного потока (2.1) на соответствующих участках.

Подобная технология передачи используется, в частности, при передаче изображений в Интернете: сначала появляются основные контуры изображения, позволяющие оценить его содержание и прервать передачу, если в

ней нет необходимости, и лишь затем происходит уточнение, и окончательное завершение передачи изображения.

- Поток чисел $a_0, a_1, a_2, a_3, a_4, a_5, \dots, a_{l-1}$ (2.6) — называют основным потоком.
- Поток чисел $b_0, b_1, b_2, b_3, b_4, b_5, \dots, b_{l-1}$ (2.7) — называют вэйвлетным потоком.

Полученный основной поток (2.6) можно рассматривать как сжатие исходного потока (2.1), а поток (2.7) как поправку к основному потоку, позволяющую восстановить исходный поток.

Если поток (2.6) все еще велик для передачи, то аналогичной процедурой его расщепляют на два потока: поток, являющийся основным для потока (2.6) (будем его называть нулевым приближением к исходному потоку (2.1) или просто нулевым потоком) и соответствующий вэйвлетный поток (его назовем первой поправкой к нулевому потоку или первым вэйвлетным потоком). В этом случае поток (2.7) можно назвать второй поправкой (или вторым вэйвлетным потоком). Возможно дальнейшее продолжение процесса расщепления; на k -м шаге получим расщепление исходного потока на $k+1$ потоков: нулевой поток (основной результат сжатия) и k вэйвлетных потоков, последовательное добавление которых к нулевому потоку приводит к последовательному уточнению результата сжатия вплоть до полного восстановления исходного потока.

Заключение

Результатом исследования является алгоритм сокращения объема числового потока. Излагаемая методика похожа на разложение по формуле Тейлора, где производные заменены соответствующими разностями. Такой процесс расщепления иногда применяют и к вэйвлетным потокам; получающийся результат называют вэйвлет-пакетом.

Литература

1. Демьянович Ю.К., Ходаковский В.А. Введение в вэйвлеты. 2007.
 2. Демьянович Ю.К. Вэйвлеты и минимальные сплайны. 2003
-

СТРУКТУРА СПЛАЙН-ВЭЙВЛЕТНЫХ РАЗЛОЖЕНИЙ¹

Мирошниченко И.Д.

*ст. преп. кафедры параллельных алгоритмов, irina_mir_@mail.ru
Санкт-Петербургский государственный университет*

Аннотация: Рассматриваются условия вложенности (вообще говоря, разрывных и неполиномиальных) сплайновых пространств, получающихся при удалении группы узлов (гнезд), дается вэйвлетное разложение упомянутых пространств, строятся матрицы вложения и продолжения, а также соответствующие формулы декомпозиции и реконструкции. Результаты могут быть применены для сплайн-вэйвлетного разложения рассмотренных пространств.

1. Введение

Ранее в работе авторов (см. [1]) рассматривались негладкие сплайн-вэйвлетные разложения первого порядка, получаемые методом последовательного удаления узлов, что приводило к упрощению формулы для вычисления таких сплайн-вэйвлетных разложений с сохранением свойств адаптивности к обрабатываемому потоку. В той же работе была установлена независимость вэйвлетного разложения от порядка удаления узлов. Но в машинной арифметике с плавающей точкой численная реализация процесса последовательного удаления большого количества узлов приводит к быстрому накоплению ошибок округления, поэтому такой подход оправдан лишь при вычислениях в реальном масштабе времени, когда запаздывание не допустимо.

В данной работе, как и в работе [2] рассматривается ситуация, когда возможно запаздывание при обработке поступающего числового потока; в этом случае возможно одновременное удаление групп последовательных узлов (гнезд), что позволяет применить распараллеливание процесса вычислений. В предлагаемой статье продолжено рассмотрение условий вложенности (вообще говоря, разрывных и неполиномиальных) сплайновых пространств, получающихся при удалении группы узлов (гнезд), дается их вэйвлетное разложение, строятся матрицы вложения и продолжения, а также соответствующие формулы декомпозиции и реконструкции. Изложение ведется для одного гнезда, но в заключении показано, как распространить полученные результаты на множество гнезд; последнее важно для параллельной обработки числовых потоков.

2. Вложенность пространств

Пусть задана полная цепочка двумерных векторов $A_N \stackrel{\text{def}}{=} \{a_j\}_{j \in J'_{N-1}}$ определены функции $\omega_j(t)$, $t \in G_n$, $j \in J'_{N-1}$ с помощью аппроксимационных соотношений (согласно определениям, приведённым в работе [2])

¹ Работа частично поддержана грантами РФФИ 10-01-00297 и 10-01-00245.

$$\sum_{j \in J'_{N-1}} a_j \omega_j(t) = \varphi(t) \quad \forall t \in G_N \quad \omega_j(t) \equiv 0 \quad \forall t \in G_N \setminus S_j, j \in J'_{N-1},$$

то есть на множестве G_N определены функции $\omega_{-1}(t)$, $\omega_j(t)$, $\omega_{N-1}(t)$.

$$\omega_{-1}(t) = \begin{cases} \frac{\det(\varphi(t), a_0)}{\det(a_1, a_0)}, & \text{при } t \in (x_0, x_1) \\ 0 & \text{при } t \in [a, b] \cap G_N \setminus S_{-1} \end{cases} \quad \text{для } j \in J_{N-2}$$

$$\omega_j(t) = \begin{cases} \frac{\det(a_{j-1}, \varphi(t))}{\det(a_{j-1}, a_j)}, & \text{при } t \in (x_j, x_{j+1}) \\ \frac{\det(\varphi(t), a_{j+1})}{\det(a_j, a_{j+1})}, & \text{при } t \in (x_{j+1}, x_{j+2}) \\ 0 & \text{при } t \in [a, b] \cap G_N \setminus S_j \end{cases}$$

$$\omega_{N-1}(t) = \begin{cases} \frac{\det(a_{N-2}, \varphi(t))}{\det(a_{N-2}, a_{N-1})}, & \text{при } t \in (x_{N-1}, x_N) \\ 0 & \text{при } t \in [a, b] \cap G_N \setminus S_{-1} \end{cases}$$

В пространстве $S_N \stackrel{\text{def}}{=} Cl_p \mathcal{L}\{\omega_j\}_{j \in J'_{N-1}}$, где $\mathcal{L}\{\dots\}$ — линейная оболочка функций, указанных в фигурных скобках, а Cl_p — замыкание в топологии точечной сходимости, ввиду того, что $\varphi(t)$, $t \in G_N$, $j \in J'_{N-1}$ — двухкомпонентная вектор-функция с линейно независимыми компонентами на любом интервале $(a', b') \in G_N$, функции $\omega_j(t)$, $j \in J'_{N-1}$ линейно независимы на интервале G_N и, следовательно, являются базисом пространства S_N , поэтому, $\dim S_N = N + 1$.

В случае удаления совокупности узлов (гнезда)

$$\Gamma = \Gamma(k, s) \stackrel{\text{def}}{=} \{x_{k-s}, x_{k-s+1}, \dots, x_k, x_{k+1}\},$$

$$x_j^* \stackrel{\text{def}}{=} x_j \quad \text{при } j \leq k - s - 1,$$

$$x_j^* \stackrel{\text{def}}{=} x_{j+s+2} \quad \text{при } j \geq k - s, \quad j \in J'_{N-s-2}$$

$$\text{на сетке } S_j^* \stackrel{\text{def}}{=} (x_j^*, x_{j+1}^*) \cup (x_{j+1}^*, x_{j+2}^*), \quad j \in J'_{N-s-4},$$

$$G^* \stackrel{\text{def}}{=} \bigcup_{j \in J'_{N-s-3}} (x_j^*, x_{j+1}^*)$$

задается полная цепочка двумерных векторов $\{a_j^*\}_{j \in J'_{N-s-3}}$ и строится система функций $\{\omega_j^*\}_{j \in J'_{N-s-3}}$ из соотношений

$$\sum_{j \in J'_{N-s-3}} a_j^* \omega_j^*(t) = \varphi(t) \quad \forall t \in G^* \quad \omega_j^*(t) \equiv 0 \quad \forall t \in G^* \setminus S_j^*, j \in J'_{N-s-3},$$

Тогда $\dim \mathbb{S}_{N(\Gamma)} = N - s - 1$, где $\mathbb{S}_{N(\Gamma)} \stackrel{\text{def}}{=} Cl_p \mathcal{L} \{\omega_j^*\}_{j \in J'_{N-s-3}}$.

Обозначим (B) условие: при $\forall t \in G, j \in J'_{N-s-3}$ справедливы соотношения

$$\begin{aligned} a_j^* \omega_j^*(t) &= a_j \omega_j(t) \quad \text{при } j \leq k - s - 3, \\ a_j^* \omega_j^*(t) &= a_{j+s+2} \omega_{j+s+2}(t) \quad \text{при } j \geq k - s. \end{aligned}$$

Справедливы следующие теоремы.

Теорема 1. При условии (B) выполняются тождества

$$\begin{aligned} \omega_{k-s-2}^*(t) &\equiv \sum_{j=k-s-2}^{k+1} p_{k-s-2,j} \omega_j(t) \quad \text{где } p_{k-s-2,j} \stackrel{\text{def}}{=} \frac{\det(a_j, a_{k-s-1}^*)}{\det(a_{k-s-2}^*, a_{k-s-1}^*)} \\ \omega_{k-s-1}^*(t) &\equiv \sum_{j=k-s-2}^{k+1} p_{k-s-1,j} \omega_j(t) \quad \text{где } p_{k-s-1,j} \stackrel{\text{def}}{=} \frac{\det(a_{k-s-2}^*, a_j)}{\det(a_{k-s-2}^*, a_{k-s-1}^*)}, \\ &j = k - s - 2, \dots, k + 1. \end{aligned}$$

Обозначим для удобства

$$J'_{s,k,N} \stackrel{\text{def}}{=} \{-1, 0, \dots, k-s-2\} \cup \{k+1, k+2, \dots, N-1\}.$$

Обозначим условие C:

$$\begin{aligned} a_j &= c_j^* a_j^* \quad \text{при } j \leq k - s - 2 \quad \text{где } c_j^* \neq 0, j \in J'_{s,k,N} \\ a_{j+s+1} &= c_{j+s+1}^* a_{j-1}^* \quad \text{при } j \geq k - s. \end{aligned}$$

Теорема 2. Для выполнения условия (B) необходимо и достаточно выполнения условия (C).

Теорема 3. *Если выполнено условие (C), то пространство $\mathbb{S}_{N(\Gamma)}$ содержится в пространстве \mathbb{S}_M .*

Введем вектор-функции

$$\omega_*(t) \stackrel{\text{def}}{=} (\omega_{-1}^*(t), \omega_0^*(t), \dots, \omega_{N-s-3}^*(t))^T$$

$$\omega(t) \stackrel{\text{def}}{=} (\omega_{-1}(t), \omega_0(t), \dots, \omega_{N-1}(t))^T$$

Теорема 4. *Если выполнено условие (C), то справедливы калибровочные соотношения $\omega_* = \mathfrak{B}_{N(\Gamma)}\omega$ где результат транспонирования матрицы $\mathfrak{B}_{N(\Gamma)}$ может быть представлен особым видом матрицы, где на главной диагонали и на $k-s$ прилегающих к ней диагоналях стоят ненулевые элементы, а остальные элементы — нули.*

$$\mathfrak{B}_{N(\Gamma)}^T = \begin{pmatrix} -1 & \dots & k-s-3 & k-s-2 & & k-s-1 & k-s & \dots & N-s-3 \\ c_{-1}^* & \dots & 0 & 0 & & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & & \dots & \dots & \dots & \dots \\ 0 & \dots & c_{k-s-3}^* & \dots & & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & c_{k-s-2}^* & & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & p_{k-s-2, k-s-1} & p_{k-s-1, k-s-1} & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & p_{k-s-2, k-s} & p_{k-s-1, k-s} & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & \dots & 0 & p_{k-s-2, k} & p_{k-s-1, k} & 0 & \dots & 0 & k \\ 0 & \dots & 0 & 0 & c_{k+1}^* & 0 & \dots & 0 & k+1 \\ 0 & \dots & 0 & 0 & 0 & c_{k+2}^* & \dots & 0 & k+2 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & N-1 \end{pmatrix}$$

Рассмотрим матрицу размером $N-s-1 \times N+1$ с элементами

Доказывается, что эта $\mathfrak{Q}_{N(\Gamma)}^T = \{q_{i,j}\}$ матрица может быть представлена в псевдо диагональном виде, где $q_{i,j} \stackrel{\text{def}}{=} \langle g_i^*, \omega_j \rangle$ при $i \in J_{N-s-3}^*$, $j \in J_{N-1}$

$$\mathfrak{Q}_{N(\Gamma)}^T \stackrel{\text{def}}{=} \begin{pmatrix} -1 & \dots & k-s-3 & k-s-2 & & k-s-1 & k-s & \dots & k+1 & k+2 & \dots & N-1 \\ 1/c_{-1}^* & \dots & 0 & 0 & & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & & \dots \\ 0 & \dots & 1/c_{k-s-3}^* & 0 & & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1/c_{k-s-2}^* & & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & q_{k-s-1, k-s-2} & q_{k-s-1, k-s-1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 1/c_{k+2}^* & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1/c_{N-1}^* \end{pmatrix}$$

Матрица $\mathfrak{Q}_{N(\Gamma)}$ называется *матрицей продолжения*.

Теорема 5. Матрица $\mathfrak{Q}_{N(\Gamma)}$ является левой обратной к матрице $\mathfrak{B}_{N(\Gamma)}^T$.
Рассмотрим матрицу вида

$$\mathfrak{Q} = \begin{pmatrix} -1 & \dots & k-s-3 & k-s-2 & k-s-1 & k-s & \dots & k+1 & k+2 & \dots & N-1 \\ z_{-1} & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & z_{k-s-3} & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & z_{k-s-2} & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & x & y & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & z_{k+2} & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & z_{N-1} \end{pmatrix} \begin{matrix} -1 \\ \dots \\ k-s-3 \\ k-s-2 \\ k-s-1 \\ k-s \\ \dots \\ N-s-3 \end{matrix}$$

где x, y, z_i — некоторые вещественные числа, $(i \in \{-1, 0, \dots, k-s-2\} \cup \{k+2, k+3, \dots, N-1\})$ а $s+2$ столбца с номерами $k-s, \dots, k+1$ состоят из нулей.

Лемма. Для матрицы $\mathfrak{B}_{N(\Gamma)}^T$ существует единственная левая обратная матрица выше указанного вида.

2. Сплайн-вэйвлетное разложение пространства \mathbb{S}_N .

Рассмотрим оператор $P_{N(\Gamma)}$ проектирования пространства \mathbb{S}_N на подпространство $\mathbb{S}_{N(\Gamma)}$:

$$P_{N(\Gamma)}u \stackrel{\text{def}}{=} \sum_{j \in J'_{N-s-3}} \langle g_j^*, u \rangle \omega_j^* \quad \forall u \in \mathbb{S}_N.$$

Введем оператор $Q_{N(\Gamma)} = I - P_{N(\Gamma)}$, где I — тождественный в \mathbb{S}_N оператор.

Пространством вэйвлетов (всплесков) называется пространство

$$W_{N(\Gamma)} \stackrel{\text{def}}{=} Q_{N(\Gamma)} \mathbb{S}_N.$$

$\mathbb{S}_N = \mathbb{S}_{N(\Gamma)} W_{N(\Gamma)}$ — прямое разложение или сплайн-вэйвлетное разложение пространства \mathbb{S}_N .

Пусть $u \in \mathbb{S}_N$. Учитывая формулу прямого разложения получаем два представления элемента u .

$$u = \sum_{j \in J'_{N-1}} c_j \omega_j, \quad u = \sum_{i \in J'_{N-s-3}} a_i \omega_i^* + \sum_{j \in J'_{N-1}} b_j \omega_j,$$

где $a_i \stackrel{\text{def}}{=} \langle g_i^*, u \rangle, \quad b_j, c_j \in R^1.$

Приравняем их и получим

$$\sum_{j \in J'_{N-1}} c_j \omega_j = \sum_{i \in J'_{N-s-3}} a_i \sum_{j \in J'_{N-1}} p_{i,j} \omega_j^* + \sum_{j \in J'_{N-1}} b_j \omega_j,$$

Так как $\{\omega_j\}_{j \in J'_{N-1}}$ — линейно независимая система, получим **формулы реконструкции**

$$c_j = \sum_{i \in J'_{N-s-3}} a_i p_{i,j}^{k+1} + b_j \quad \forall j \in J'_{N-1},$$

Перепишем эту формулу как

$$c_j = \sum_{i \in J'_{N-s-3}} c'_j \langle g_i^*, \omega_j \rangle p_{i,j} + b_j \quad \forall j \in J'_{N-1}.$$

Учитывая, что $q_{i,j} \stackrel{\text{def}}{=} \langle g_i^*, \omega_j \rangle$, получим

$$b_j = c_j - \sum_{i \in J'_{N-s-3}} \sum_{j \in J'_{N-1}} c_j q_{i,j} p_{i,j} \quad \forall j \in J'_{N-1}.$$

Откуда

$$a_i = \sum_{j \in J'_{N-1}} c_j q_{i,j} \quad \forall j \in J'_{N-s-3}.$$

Формулы для b_j и a_i — представляют **формулы декомпозиции**.

Запишем в матричной форме. Введем вектор столбцы

$$a \stackrel{\text{def}}{=} (a_{-1}, \dots, a_{N-s-3})^T, \quad b \stackrel{\text{def}}{=} (b_{-1}, \dots, b_{N-1})^T, \quad c \stackrel{\text{def}}{=} (c_{-1}, \dots, c_{N-1})^T.$$

Тогда $b = c - B_{N(\Gamma)}^T Q_{N(\Gamma)} c$, $a = Q_{N(\Gamma)} c$.

Вектор a называют основной, а вектор b — взвешенной составляющей исходного вектора (потока) c .

3. Представления взвешенного разложения

Введем обозначение для натурального $M \geq 2$.

$$\mathfrak{F}_M \stackrel{\text{def}}{=} \{f \mid f = (f_{-1}, f_0, f_1, \dots, f_{M-2}, f_{M-1})^T, f_i \in \mathfrak{R}^1, i = -1, 0, 1, \dots, M-1\}.$$

Рассмотрим линейный оператор $A_{N(\Gamma)} = \mathfrak{F}_{N-s-2}$ из пространства $\mathfrak{C}_N \stackrel{\text{def}}{=} \mathfrak{F}_{N \vee}$ пространство $B_{N(\Gamma)}$, определяемый в нем матрицей $\mathfrak{Q}_{N(\Gamma)}$.

Обозначим ядро этого оператора $B_{N(\Gamma)} \stackrel{\text{def}}{=} \{b \mid b \in \mathfrak{C}_N, \mathfrak{Q}_{N(\Gamma)} b = 0\}$.

Теорема 6. *Множество вэйвлетных составляющих совпадает с пространством $B_{N(\Gamma)}$.*

Теорема 7. *Операторы $\mathfrak{D}_{N(\Gamma)}$ и $\mathfrak{R}_{N(\Gamma)}$ взаимно обратны; они реализуют линейный изоморфизм пространств \mathfrak{C}_N и $E_{N(\Gamma)}$.*

Литература

1. Ю.К. Демьянович, И.Д. Мирошниченко. «Негладкие сплайн-вэйвлетные разложения на отрезке», //Проблемы математического анализа. 63, 23–40 (2012).
 2. И.Д. Мирошниченко. «Гнездовые укрупнения и вложенность сплайновых пространств», //Материалы всероссийской научной конференции по проблемам информатики СПИСОК-2012, СПб, 166–169.с.
-

О СРАВНЕНИЕ СПЛАЙН-ВЭЙВЛЕТНЫХ РАЗЛОЖЕНИЙ С НЕКОТОРЫМИ ВЭЙВЛЕТНЫМИ РАЗЛОЖЕНИЯМИ И. ДОБЕШИ

Зубарева М.В.

*студентка математико-механического факультета СПбГУ,
mari.zubareva@gmail.com*

Аннотация: В работе рассмотрено сравнение эффективности классического подхода (с использованием вэйвлетов Добеши D4) и адаптивных сплайн-вэйвлетных разложений (с использованием полиномиальных B -сплайнов второй степени).

Введение

Из-за тенденций развития общества и технологий становится всё актуальнее задача обработки больших потоков цифровой информации. Её важная часть — проблема сокращения объёмов передаваемых данных за счёт отбрасывания несущественных (по тем или иным критериям) составляющих. На первое место среди средств решения этой задачи выходят вэйвлеты (wavelets).

Основной результат теории вэйвлетов — эффективные алгоритмы обработки больших потоков информации, то есть экономное (с точки зрения экономии ресурсов компьютера: памяти и времени обработки) разложение потока информации на основной и вэйвлетный информационные потоки. Как правило, основной информационный поток значительно менее плотный, чем исходный поток информации, поэтому его можно передать быстро. Вэйвлетный информационный поток можно передавать частично либо вообще отбросить.

Классический подход заключается в использовании равномерной сетки и ортогонального проектирования (в стандартном гильбертовом пространстве последовательностей).

На измельчающейся сетке полиномиальные B -сплайны образуют систему вложенных пространств, на основе которой строятся вэйвлетные разложения. Для улучшения приближения могут понадобиться различные степени измельчения сетки на разных частях рассматриваемого промежутка. Использование преобразования Фурье, обычного для равномерной сетки, в случае неравномерной затруднено, однако применение биортогональной системы функционалов позволяет построить вэйвлетные разложения при произвольном измельчении сетки — то есть на неравномерной адаптивной сетке.

Цель работы состоит в сравнении эффективности классического подхода (с использованием вэйвлетов Добеши D4) и адаптивных сплайн-

вэйвлетных разложений (с использованием полиномиальных B -сплайнов второй степени).

Вэйвлеты Добеши

Вэйвлеты Добеши — семейство ортогональных вэйвлетов с компактным носителем. Их вид зависит от числа ненулевых элементов в масштабирующем уравнении.

Рассмотрим вэйвлеты Добеши-D4, имеющие 4 ненулевых коэффициента в масштабирующем уравнении.

Введём обозначения: исходный поток $c = \{c_j\}$, основной — $a = \{a_i\}$, вэйвлетный — $b = \{b_i\}$.

Формулы декомпозиции

$$a_i = \frac{1 + \sqrt{3}}{4\sqrt{2}} c_{2i} + \frac{3 + \sqrt{3}}{4\sqrt{2}} c_{2i+1} + \frac{3 - \sqrt{3}}{4\sqrt{2}} c_{2i+2} + \frac{1 - \sqrt{3}}{4\sqrt{2}} c_{2i+3},$$

$$b_i = \frac{3 - \sqrt{3}}{4\sqrt{2}} c_{2i} - \frac{3 - \sqrt{3}}{4\sqrt{2}} c_{2i+1} + \frac{3 + \sqrt{3}}{4\sqrt{2}} c_{2i+2} - \frac{1 + \sqrt{3}}{4\sqrt{2}} c_{2i+3}.$$

Формулы реконструкции

$$c_{2i} = \frac{3 - \sqrt{3}}{4\sqrt{2}} a_{i-1} + \frac{3 + \sqrt{3}}{4\sqrt{2}} b_{i-1} + \frac{1 + \sqrt{3}}{4\sqrt{2}} a_i + \frac{1 - \sqrt{3}}{4\sqrt{2}} b_i,$$

$$c_{2i+1} = \frac{1 - \sqrt{3}}{4\sqrt{2}} a_{i-1} - \frac{1 + \sqrt{3}}{4\sqrt{2}} b_{i-1} + \frac{3 + \sqrt{3}}{4\sqrt{2}} a_i - \frac{3 - \sqrt{3}}{4\sqrt{2}} b_i.$$

Аппроксимация основным потоком

φ — масштабирующая функция Добеши D4.

$$\varphi(0) = 0, \varphi(1) = \frac{1 + \sqrt{3}}{2}, \varphi(2) = \frac{1 - \sqrt{3}}{2}, \varphi(3) = 0,$$

$$\varphi(t) = \frac{1 + \sqrt{3}}{4} \varphi(2t) + \frac{3 + \sqrt{3}}{4} \varphi(2t-1) + \frac{3 - \sqrt{3}}{4} \varphi(2t-2) + \frac{1 - \sqrt{3}}{4} \varphi(2t-3),$$

$$\varphi_{jk}(t) = \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t}{2^j} - k\right).$$

Пусть $u(t)$ — функция, генерирующая основной поток. Приближение основным потоком классического разложения ($\tilde{u}(t)$) вычисляется по формуле:

$$\tilde{u}(t) = \sum_k a_k \varphi_{jk}(t).$$

Адаптивные сплайн-вэйвлетные разложения

При использовании адаптивных сплайн-вэйвлетных разложений возникают задачи определения номеров выкидываемых узлов (необходимо для построения «крупной» сетки) и построения соответствующего разложение исходной функции.

Предварительные сведения

На интервале (α, β) вещественной оси рассмотрим сетку:

$$X: \dots < x_{-1} < x_0 < x_1 < \dots$$

На этой сетке зададим полиномиальные B -сплайны ω_j^B второй степени:

$$\omega_j^B = \begin{cases} \frac{(t-x_j)^2}{(x_{j+1}-x_j)(x_{j+2}-x_j)}, & \text{если } t \in [x_j, x_{j+1}); \\ (x_{j+2}-x_j)^{-1}(x_{j+2}-x_{j+1})^{-1}(x_{j+3}-x_{j+1})^{-1}[(x_j-x_{j+2}-x_{j+3}+x_{j+1})t^2 - \\ - 2(x_{j+1}x_j - x_{j+2}x_{j+3})t + x_jx_{j+1}x_{j+3} - x_jx_{j+2}x_{j+3} + x_jx_{j+1}x_{j+2} - \\ - x_{j+1}x_{j+2}x_{j+3}], & \text{если } t \in [x_{j+1}, x_{j+2}); \\ \frac{(t-x_{j+3})^2}{(x_{j+3}-x_{j+2})(x_{j+3}-x_{j+1})}, & \text{если } t \in [x_{j+1}, x_{j+2}); \\ 0, & \text{если } t \notin [x_j, x_{j+3}] \end{cases}$$

Таким образом, $supp \omega_j^B = [x_j, x_{j+3}]$.

В $C^1(\alpha, \beta)$ рассмотрим линейные функционалы $g^{(i)}$:

$$\langle g^{(i)}, u \rangle \stackrel{\text{def}}{=} u(x_{i+1}) + \frac{1}{2}(x_{i+2} - x_{i+1})u'(x_{i+1}) \text{ для } \forall u \in C^1(\alpha, \beta).$$

Исходный поток генерируется по формуле:

$$\langle g^{(i)}, u \rangle = c_i.$$

Алгоритм выбора сетки

Пусть $X = \{x_j\}$ — равномерная сетка, введённая на промежутке вещественной оси $[\sigma, \beta]$. С учётом характера функции u , рассматриваемой на данном промежутке, получим из равномерной сетки X новую сетку \tilde{X} :

$$\tilde{x}_0 = x_0, \quad \tilde{x}_1 = x_{S(x_0)}, \quad \tilde{x}_2 = x_{(S(x_0) + S(x_{S(x_0)}))}, \quad \dots$$

Чтобы получить новую сетку, удаляем из X вслед за очередным сохраняемым узлом $x_j = \tilde{x}_k$ количество узлов, равное $(S(x_j) - 1)$, то есть $\tilde{x}_{k+1} = x_{j+S(x_j)}$.

Функция S :

- N — число узлов в исходной сетке;
- i — номер рассматриваемого узла исходной сетки, $0 \leq i \leq N-1$;
- P — относительное число сохраняемых узлов, $0 < P \leq 1$;
- q — число усредняемых значений $|u'(x_j)|$ в соседних узлах, $1 \leq q \leq N-i$;
- g — максимальное число одновременно выбрасываемых узлов, $1 \leq g \leq N-2$;

$$C = \sum_{i=0}^{N-1} |u'(x_i)|, \quad j = \left\lfloor \frac{CP}{\sum_{r=i}^{\min\{i+q-1, N\}} \frac{|u'(x_r)|}{q} + \frac{C}{g}} \right\rfloor$$

$$S(x_i) = \begin{cases} 1, & \text{если } j < 1, \\ g, & \text{если } j > g, \\ j, & \text{иначе.} \end{cases}$$

Формулы декомпозиции и реконструкции

При удалении узлов из сетки X получаем новую, более «крупную» сетку \tilde{X} . Для неё построим новую систему полиномиальных B -сплайнов второй степени, причём «новые» сплайны выражаются через «старые»:

$$\omega_j^B = \sum_j d_{ij} \omega_j^B.$$

Соотношения такого вида называются *калибровочными соотношениями*.

$$d_{ij} = \langle g^{(j)}, \omega_j^B \rangle.$$

В матрице $D = d_{ij}$ в каждом столбце будет не более 3 ненулевых элементов. Рассмотрим функцию

$$\tilde{u}(t) = \sum_{x_j \in X} c_j \omega_j^B(t).$$

Декомпозиция:

$$a_i = \langle \tilde{g}^i, \tilde{u} \rangle$$

$$b_j = c_j - \sum_i d_{ij} a_i.$$

Реконструкция:

$$c_j = \sum_i a_i d_{ij} + b_j.$$

Аппроксимация основным потоком

Проекция функции $\tilde{u}(t)$ на пространство сплайнов (то есть риблизение), построенных на «крупной» сетке \tilde{X} имеет вид:

$$\mu(t) = \sum_{\tilde{x}_j \in \tilde{X}} a_i \tilde{\omega}_j^B(t).$$

Сравнение двух подходов

На измелъчѐнном в 100 раз интервале [0,1) рассматривались скорость построения приближения по основному потоку и его отклонение от исходной функции.

Рассматривалась функция $u(t) = \frac{1}{1+t^2}$.

- N — размер исходного потока;
- *Норма1* — погрешность приближения при классическом подходе;
- *Норма2* — погрешность приближения при сплайн-вэйвлетном подходе;
- $N1$ — размер основного потока при классическом подходе;
- N — размер основного потока при сплайн-вэйвлетном подходе;
- *Время1* — среднее время построения приближения при классическом подходе;
- *Время2* — среднее время построения приближения при сплайн-вэйвлетном подходе.

N	N1	N2	Норма1	Норма1
2048	23	1024	4.84228e-004	5.15041e-004
4096	31	2048	2.56519e-004	2.5752e-004
8192	40	4096	1.49131e-004	1.2876e-004
16384	54	8192	8.30436e-005	6.438e-005
32768	77	16384	3.7989e-005	3.219e-005
65555	117	32768	1.37469e-005	3.219e-005

Таблица 1. Сравнение эффективности классического и сплайн-вэйвлетного подходов. Размер основного потока

N	Норма1	Норма2	Время1, с	Время2, с
2048	4.84228e-004	5.15041e-004	1.26796e-007	4.4255e-007
4096	2.56519e-004	2.5752e-004	1.30312e-007	4.4255e-007
8192	1.49131e-004	1.2876e-004	1.29543e-007	4.42835e-007
16384	8.30436e-005	6.438e-005	1.31991e-007	4.40217e-007
32768	3.7989e-005	3.219e-005	1.30772e-007	4.38356e-007
65555	1.37469e-005	3.219e-005	1.36441e-007	4.40899e-007

Таблица 2. Сравнение эффективности классического и сплайн-вэйвлетного подходов. Скорость восстановления

Заключение

По сравнению с классическим подходом сплайн-вэйвлетное разложение на неравномерной сетке показывает большую эффективность.

Литература

1. Демьянович Ю.К. Вэйвлеты & минимальные сплайны. — 2003.
 2. Демьянович Ю.К., Ходаковский В.А. Введение в вэйвлеты. — 2007.
-

РЕАЛИЗАЦИЯ СЖАТИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ СПЛАЙН-ВЕЙВЛЕТНОГО РАЗЛОЖЕНИЯ И ТЕХНОЛОГИЙ OPENCL И CUDA

Д.М. Гусев

*студент 543 группы математико-механического факультета,
кафедра параллельных алгоритмов, jos.shad@gmail.com*

Аннотация: Рассматривается разработка программ на GPU, общая структура сплайн-вейвлетного разложения, описывается алгоритм сжатия изображений и видеозаписей на GPU с использованием сплайн-вейвлетного разложения

Введение

На сегодняшний день вейвлеты стали, чуть ли не главным математическим инструментом в области исследования и анализа структуры сигналов. Вейвлетное преобразование уже давно зарекомендовало себя как эффективный и качественный способ разложения числовых данных. Эффективность вейвлетного преобразования данных достигается за счет экономного использования ресурсов компьютера (памяти и времени обработки) для разбиения потока числовой информации на его составляющие: основную часть потока и уточняющую, так называемую, вейвлетную часть. При этом основной поток, как правило, менее плотный, чем исходный, а, следовательно, и более быстрый для передачи его по линиям связи, в которых уже обязательно использовать каналы с широкой полосой пропускания. Вейвлетная же часть нужна для того чтобы, восстановить исходный числовой поток без потери качества. Эта часть бывает не всегда необходимой и ее можно передавать фрагментально по мере необходимости. Естественно сразу же голову приходит идея использования вейвлетной обработки данных для разложения цифрового изображения или видео. Действительно, порой нет необходимости получать данные об изображении или видео в мельчайших деталях, достаточно увидеть картину в целом, не вдаваясь в подробности. Тем самым облегчая и увеличивая скорость передачи данных, а также уменьшая нагрузку, при всем этом не теряя качества исходного изображения.

В компьютерной графике задача повышения эффективности обработки изображения, в частности вывод алгоритмов вейвлетных разложений для качественного сжатия гладких потоков информации, является актуальной и востребованной на сегодняшний день. Требования к максимальной реалистичности изображения, а также ее адекватной скорости передачи по каналам связи постоянно растут.

Цель данной работы в создании программы сжатия изображений и видеозаписей при использовании сплайн-вейвлетного разложения на GPU, с использованием CUDA и OpenCL.

Обзор вычислений на GPU

Существует два типа архитектуры параллельных систем – SIMD и MIMD. SIMD — Single Instruction stream, Multiple Data stream. Управляющий узел один, он отправляет инструкции всем остальным процессорам.

Каждый процессор имеет свой набор данных для работы. MIMD — Multiple Instruction stream, Multiple Data Stream. Каждый процессор имеет свой собственный управляющий юнит, каждый процессор может выполнять разные инструкции.

Почти все CPU, используемые в повседневной жизни, – это параллельные системы с MIMD архитектурой и общей памятью. Высокопроизводительные кластеры (например IBM Blue Gene) – это MIMD-машины с распределенной памятью. GPU (Graphics Processing Unit, термин, впервые введенный корпорацией Nvidia для обозначения того, что графический ускоритель, первоначально используемый только для ускорения трехмерной графики, стал мощным программируемым устройством (процессором), пригодным для решения широкого класса задач, никак не связанных с графикой) в отличие от обычных CPU, по своей сути является SIMD-машиной с общей памятью.

Из-за различия архитектур GPU и CPU, эффективно работающий на CPU алгоритм может оказаться абсолютно неэффективным на GPU. Если еще обратить внимание на то, что работа с памятью и синхронизация между вычислительными ядрами также очень сильно различаются, то становится понятно, что для эффективной работы алгоритма на GPU необходимо провести полный анализ алгоритма для CPU.

Для работы с видеоускорителем необходимы специализированные средства, позволяющие эффективно писать код. Такими средствами являются технологии CUDA и OpenCL. По сути своей разработка на обеих технологиях очень похожа (более того, реализацией OpenCL на видеокартах Nvidia является CUDA). Отличие заключается в том, что CUDA немного быстрее, а OpenCL является технологией для гетерогенной разработки (что позволяет писать программы, работающие на GPU и CPU одновременно).

Общая структура вейвлетного разложения

Рассмотрим функцию $u(t) \in C(a, b)$. Будем называть её исходной функцией, указанное множество целых точек — исходной сеткой, а целые точки — узлами исходной сетки. На интервале (a, b) вещественной оси рассмотрим сетку

$$X: \dots < x_{-1} < x_0 < x_1 < \dots$$

Задача состоит в сжатии исходной функции $u(t)$ путем укрупнения сетки и разложении в основной и вейвлетный поток. Таким образом, возникает

две задачи: определение номеров выбрасываемых узлов (укрупнение) и соответствующее разложение исходной функции. Первая задача решается с помощью формул

$$\bar{S}^* = (f', i, q, P, h, g, \bar{C}) \stackrel{\text{def}}{=} \left\| \frac{\bar{C}}{P \left(\frac{1}{g} \sum_{i'=i}^{i+q-1} \left| f'(i'h) + \frac{C}{g} \right| \right)} \right\|,$$

где $S-1$ — количество следующих выбрасываемых узлов; i — номер рассматриваемого узла x_i сетки; P — относительное количество сохраняемых узлов; q — параметр усреднения; g — параметр, характеризующий максимальное количество одновременно выбрасываемых узлов;

$\bar{C} = \frac{1}{N} \sum_{i=0}^{N-1} f'(x_i)$ константа C может вычисляться приближенно. Параметр P характеризует степень сжатия изображения; он выбирается из интервала $(0, 1)$.

Опишем алгоритм разложения исходного потока на основную и вейвлетную составляющие. Рассмотрим полиномиальный B -сплайн ω_j^B второй степени на сетке X ; его можно задать следующим образом:

$$\omega_j^B(t) = (t-x_j)^2(x_{j+1}-x_j)^{-1}(x_{j+2}-x_j)^{-1}, \text{ при } t \in [x_j, x_{j+1}),$$

$$\omega_j^B(t) = (x_{j+2}-x_j)^{-1}(x_{j+2}-x_{j+1})^{-1}(x_{j+3}-x_{j+1})^{-1}[(x_j-x_{j+2}-x_{j+3}+x_{j+1})t^2 - 2(x_{j+1}x_j-x_{j+2}x_{j+3})t + x_jx_{j+1}x_{j+3} - x_jx_{j+2}x_{j+3} + x_jx_{j+1}x_{j+2} - x_{j+1}x_{j+2}x_{j+3}],$$

при $t \in [x_{j+1}, x_{j+2})$,

$$\omega_j^B(t) = (t-x_{j+3})^2(x_{j+3}-x_{j+2})^{-1}(x_{j+3}-x_{j+1})^{-1}, \text{ при } t \in [x_{j+2}, x_{j+3}],$$

$$\omega_j^B(t) = 0, \text{ при } t \notin [x_j, x_{j+3}].$$

Далее понадобятся линейные функционалы g^i , определяемые на непрерывно дифференцируемых функциях по формуле

$$g^i = u(x_{i+1}) + (x_{i+2} - x_{i+1}) \frac{u'(x_{i+1})}{2}.$$

В качестве функции $u(t)$ рассмотрим функцию $u(t) = \sum_{x_j \in X} c_j \omega_j(t)$, где c_j — элементы исходного потока, а $X = \{x_j\}$ — исходная сетка.

Проекция функции $u(t)$ спомощью укрупненной сетки \tilde{X} имеет вид

$$\tilde{u} = \sum_{\tilde{x}_j \in \tilde{X}} \langle \tilde{g}^i, u \rangle \tilde{\omega}_j(t).$$

Используя определение функционалов g^i , получаем

$$\tilde{u}(t) = \sum_{\tilde{x}_j \in \tilde{X}} a_j \tilde{\omega}_j(t) = \sum_{\tilde{x}_j \in \tilde{X}} \tilde{\omega}_j(t) \sum_{(x_i \in X)} c_i \left(\omega_i(\tilde{x}_{j+1}) + (\tilde{x}_{j+2} - \tilde{x}_{j+1}) \frac{\omega'_j(x_{j+1})}{2} \right).$$

Из этой формулы находятся числа a_i , которые представляют собой основной поток. Вейвлетный поток рассчитывается по следующей формуле $b_s = \langle g^s, u - \tilde{u} \rangle$.

Поскольку в полученных потоках содержится большое количество подряд идущих нулей (в связи с укрупнением сетки), то алгоритмы сжатия потоков данных начинают работать с большей эффективностью.

Зная только основной поток можно восстановить исходный поток с потерей данных по формуле

$$\tilde{u}(t) = \sum_{\tilde{x}_j \in \tilde{X}} a_j \tilde{\omega}_j(t).$$

Алгоритм сжатия видеозаписей

Современная компьютерная графика делится на два класса: растровая и векторная; простейшее растровое изображение может быть представлено в виде матрицы пикселей. для примера рассмотрим несжатое изображение в формате BMP с цветовой моделью RGB — оно представляет собой ряд специальных заголовков, содержащих дополнительную информацию об изображении, и само изображение в виде массива пикселей, записанных построчно снизу вверх. Каждый пиксель занимает три байта и представляет собой цвет в модели RGB, т.е. три целых числа от 0 до 255 включительно. Несжатый видеопоток представляет собой последовательность изображений в формате BMP.

Для того, чтобы оценить размеры несжатых видеозаписей приведена таблица ниже:

	720	1080
24 кадра в секунду	1 900 Мб	4 270 Мб
30 кадров в секунду	2 370 Мб	5 340 Мб

Таблица 1. Размер 30-секундной записи

Поскольку в настоящий момент используются мониторы достаточно больших размеров, то видеозаписи с меньшим разрешением выглядят зернисто. Однако передавать HD видеозаписи неудобно из-за их размера. В связи с этим такие потоки данных необходимо сжимать.

Каждый кадр исходной видеозаписи переводится из RGB в YCbCr. После преобразования имеем 3 «слоя» данных. Наиболее важный «слой» — слой

насыщенности. Цветовые слои несут гораздо меньше информации. Каждый слой мы можем представить в качестве потока информации и соответственно обрабатывать его. Поскольку более качественное сплайн-вейвлетное разложение получается при плавном изменении потока, то поток из изображения необходимо получать не построчно, а «змейкой».

Поскольку в цветовых слоях информации содержится меньше, то можно сделать сетку для разложения более редкой.

Алгоритм, занимающийся собственно самим разложением и определением разреженной сетки, исполняется на видеокарте. Поток делится на равные участки, каждый из участков передается своему ядру видеокарты (общими точками потока будут только 2 точки в конце каждого участка потока и 1 точка в начале). После окончания действия алгоритма, каждый вычислительный модуль имеет основной и вейвлетный поток. Оба потока передаются на CPU, где происходит сжатие потоков и сохранение на жесткий диск.

Результаты времени работы алгоритма сплайн-вейвлетного разложения для одного кадра приведены ниже, в скобках указано количество потоков:

	CPU (1)	CPU (4)	CPU (256)
1920×1080	25 сек	8 сек	1 сек
1280×720	11 сек	3 сек	1 сек

Таблица 2. CPU — Intel i5-2400, GPU — AMD Radeon HD 6850

С уменьшением размера кадра GPU начинает проигрывать CPU, поскольку затраты на компиляцию программы на видеокарте и копирование данных в память видеокарты и из нее начинает занимать больше времени, чем само исполнение программы.

В результате работы программы было исходная несжатая видеозапись размером 4 Гб была разложена и сжата на основной поток (450 Мб) и вейвлетный (600 Мб).

Восстановление исходной видеозаписи являет собой, по сути, выполненные в обратном порядке шаги алгоритма сжатия. Для каждого кадра берутся 3 соответствующих слоям сжатых потока. Каждый поток распаковывается, а затем отправляется на видеокарту, где происходит процесс восстановления (с использованием или без использования вейвлетного потока), восстановленные потоки расворачиваются из «змейки» в построчное представление, а затем применяется преобразование из YCbCr в RGB. Полученное изображение дописывается к видеопотоку.

Заключение

В результате работы было проведено сжатие видеозаписи с использованием сплайн-вейвлетного разложения и GPU. Было установлено значительное ускорение разложения на видеопроцессоре по сравнению с CPU.

Представляет интерес дальнейшее развитие сжатия видеопоследовательностей. Необходимо использовать более сложные и качественные алгоритмы сплайн-вейвлетного разложения. Также узким местом системы оказалось непосредственное сжатие информационного потока стандартными средствами (gzip, bzip). Было бы интересным реализация алгоритмов сжатия на GPU.

Литература

1. *Демьянович Ю.К., Ходаковский В.А.* Введение в вейвлеты. 2007
 2. *Миано Дж.* Форматы и алгоритмы сжатия изображений в действии. М. 2003. 336 с.
 3. www.khronos.org/opencv/
 4. http://www.nvidia.ru/object/cuda_home_new_ru.html
-

ОПРЕДЕЛЕНИЕ И РЕАЛИЗАЦИЯ ВЫЧИСЛЕНИЙ ДЛЯ СТРУКТУР JSON

Д. Гершун

*студент 543 группы, Математико-Механический факультет
Санкт-Петербургский государственный университет*

Аннотация: Язык JSON считается форматом описания данных. В данном докладе делается попытка добавления в него способа описывать алгоритмы. В теории, реализация этой идеи даст возможность моделировать произвольные системы на языке JSON.

1. Введение

Язык JSON позволяет описывать структуры и массивы данных, не прибегая к эмуляции с помощью списков. В основе работы лежит модификация языка для задания алгоритмов. Делается попытка получить средство для описания поведения JSON-овских объектов. Результатом подобной системы может стать возможность моделирования совершенно произвольной системы, так как к возможности стандартного языка JSON описывать любые структуры добавляется возможность определять алгоритмы.

Ничего о реализациях этой задачи ранее не известно, вероятно, их не было. Говоря о структурах, нельзя не сказать о языке LISP, вычисляющий структуры на списках. Однако еще ближе в представленной теме лежит CLIPS, являющийся, в некотором роде, «надстройкой» над LISP. Система CLIPS моделирует структуры в стиле списка пар «ключ : значение», которые принимают вид набора правил «условия \Rightarrow действия».

Целью работы является, в идеале, разработка системы для моделирования систем реального мира с определенной точностью. Более конкретно — разработка языка для выполнения алгоритмов с добавлением специальных примитивов для изменения той структуры данных, частью которой данная программа является.

Планировалось сделать прототип системы, которая может осуществлять простейшие арифметические операции.

2. Программное обеспечение

Язык JSON выбран как наиболее лаконичный язык, подходящий для данной задачи. Для ядра системы выбран язык программирования Python 3, как один из наиболее читаемых и высокоуровневых языков. Удобство разработки и безопасность с точки зрения сохранения данных поддерживается системой контроля версий git и веб-сервисом для хостинга проектов github.

3. Структура системы и используемые алгоритмы.

Система состоит из двух основных файлов и одного вспомогательного.

Одним из основных файлов является т.н. «ядро». В «ядре», которое представляет из себя программу на языке Python, исполняются элементарные базовые команды, необходимые для функционирования системы. Они включают в себя элементарные арифметические операции, обработку вызова функции, а также расширяющие возможности системы команды, вроде предоставления пользователю возможности модифицировать систему в непосредственно в процессе выполнения.

Во втором файле находится сама программа на языке JSON; в нем же находится и определение функций, используемых в программе. Кроме этого, в этом файле хранится структура, исполняющая функции стека, а также структура, через которую пользователь может задать входные параметры. Программа представляет из себя рекурсивные операции-функции в формате JSON. Они исполняются с помощью базовых операций из «ядра».

Третий, вспомогательный файл, представляет собой набор шаблонов-примеров различных функций. Его можно рассматривать, как часть документации по пользованию системой. Этот файл был добавлен в процессе работы, как хранилище уже реализованных функций, которые не требуются на данном этапе разработки.

4. Реализация.

Далее разберем работу системы более подробно.

Первым шагом «ядро» загружает файл с программой в формате JSON и, с помощью встроенных библиотек, превращает ее в обычные для Python структуры данных (массивы, словари). В этом файле отдельно выделены внешние данные, стек и, собственно, само описание программы.

Во структуре, вольно названной «внешними данными» хранятся данные, которые пользователь передает функциям в виде параметров.

Одной из наиболее занимательных частей работы является организация работы стека. Стек состоит из массива стековых фреймов, указателя активного фрейма, указателей на текущую функцию и текущую инструкцию. В зависимости от вызовов вложенных функций и окончания их работы эти данные изменяются.

В конце исполнения программы производится обратная запись получившейся структуры в файл в формате JSON.

5. Что уже реализовано и «планы на будущее»

На данный момент реализованы базовые операции «ядра», в том числе арифметические и логические. Они позволяют выполнять указанные эле-

ментарные действия при явно заданных аргументах. Кроме них написана обработка операции `goto`, дающей возможность определять циклы.

Также есть взаимодействие с пользователем, который может изменять структуру JSON прямо в процессе выполнения программы.

Практически реализована работа со стеком, который позволяет вызывать вложенные функции, не теряя информацию о текущих значениях и месте выполнения программы. Эти функции позволяют определять, к примеру, операции, у которых в качестве аргументов заданы не явные значения, а выражения или другие функции.

В данный момент ведется работа над определением арифметических операций в терминах JSON. В будущем планируется расширить систему определив циклы, различные операции с рекурсивными операндами, написать на JSON алгоритм Евклида, как пример возможностей реализованного расширения языка JSON. В целях оптимизации временных затрат на выполнение возможно также переписать ядро на языке C.

Литература

1. Хювёнен Э., Сеппенен И. «Мир Lisp'a.» М.: Мир, 1990
 2. N. Wirth, Algorithms and Data Structures = Programs, 1985 edition, updated for Oberon in August 2004
-

СТРУКТУРА РАЗРЕЖЕННОГО ВЭЙВЛЕТНОГО ГРЕБНЯ

Бухтин Е.В.

*студент 5 курса кафедры системного программирования СПбГУ,
mr.johnson.spb@gmail.com*

Аннотация: В данном докладе рассматривается разреженная структура сплайн-вейвлетного разложения и один алгоритм сжатия цифровой информации, основанный на сплайн-вейвлетном разложении первого порядка.

Введение

В современном мире технологического прогресса и инновационных технологий огромное значение придается информации, ее качеству, а также скорости ее передачи. И если в некоторых случаях высокое качество может быть не основным критерием, то скорость передачи всегда существенна.

В свете огромных размеров современных потоков информации (10^{12} – 10^{16} символов), задача сокращения объемов цифровой информации за счет отбрасывания несущественных ее составляющих весьма актуальна, причем степень важности эффективного решения этой задачи постоянно возрастает.

Среди известных инструментов решения данной задачи важное место занимают вейвлеты. Основной результат теории вейвлетов — эффективные алгоритмы обработки больших потоков информации. Под эффективностью в данном случае понимают экономное с точки зрения использования вычислительных ресурсов разложение потока информации на составляющие — основной и вейвлетный информационный потоки. Как правило, основной информационный поток значительно менее плотный, чем исходный поток информации, поэтому его можно передать быстро. Вейвлетный же информационный поток можно передавать фрагментарно, либо вообще отбросить, в зависимости от конкретной области применения.

Цель данной работы заключается в создании пакета программ сжатия и восстановления числовой информации, реализующего алгоритм, основанный на сплайн-вейвлетных разложениях первого порядка и в анализе возможностей распараллеливания указанного алгоритма.

Дадим небольшую справку по теории вопроса в согласии с работами [1]–[4]:

Пусть N — натуральное число, и на отрезке $[a, b]$ имеется сетка

$$X : a = x_0 < x_1 < \dots < x_{N-2} < x_{N-1} < x_N = b,$$

$$G \equiv \bigcup_{i \in \{0, 1, \dots, N-1\}} (x_i, x_{i+1}), \quad S_i \equiv (x_i, x_{i+1}) \cup (x_{i+1}, x_{i+2}), \quad i \in \{0, 1, \dots, N-1\},$$

$$P_u \equiv \sum_{i \in \{-1, 0, 1, \dots, N-1\}} \text{::} \langle g_i, u \rangle \underline{\omega}_i \quad \forall u \in S,$$

где g_i — система функционалов, биортогональная системе $\underline{\omega}_i$ (достаточные условия существования ее приведены в [4]).

Введем также оператор $Q = I - P$; пространством вэйвлетов называется пространство $W \equiv QS$. Разложение же $S = \underline{S} \otimes W$ называется сплайн-вэйвлетным разложением пространства S .

Пусть $u \in S$. Имеем два представления элемента u :

$$u = \sum_{i \in \{-1, 0, 1, \dots, N-1\}} \text{::} c_i \omega_i, \text{ и}$$

$$u = \sum_{i \in \{-1, 0, 1, \dots, N-1\}} \text{::} a_i \underline{\omega}_i + \sum_{i \in \{-1, 0, 1, \dots, N-1\}} \text{::} b_i \omega_i.$$

Далее в [3] выводятся формулы реконструкции

$$c_j = \sum_{i \in \{-1, 0, 1, \dots, N-1\}} \text{::} a_i p_{ij} + b_j, \quad \forall i \in \{-1, 0, 1, \dots, N-1\},$$

и декомпозиции

$$b = c - P^T Qc, \quad a = Qc,$$

где a, b, c — составленные из a_i, b_i, c_i вектор-столбцы, P, Q — матрицы определенного вида, p_{ij} — элементы матрицы P . При этом вектор a называется основным потоком, а вектор b — вэйвлетным потоком при сплайн-вэйвлетном разложении исходного потока c .

Работа алгоритма происходит следующим образом:

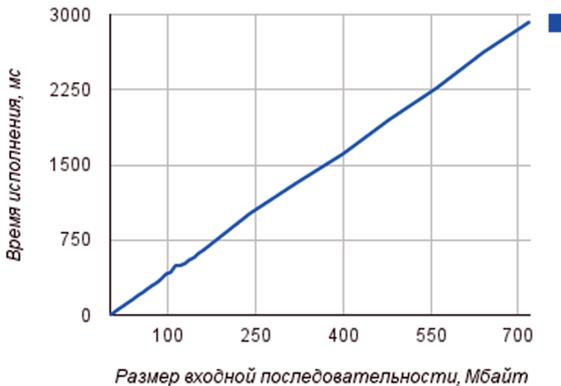
- Укрупнение сетки, получение основного и вэйвлетного потоков по формулам декомпозиции.
- Построение сплайна по основному потоку, приближенное восстановление входного потока (восстановление с потерей информации), оценка уклонения от исходного потока.
- Восстановление входного потока по формулам реконструкции (восстановление без потери информации), оценка уклонения от исходного потока (должно быть точное совпадение).

В результате сравнения исходного и восстановленного потоков была установлена корректность реализации алгоритма (восстановление без потери информации); также возможно измерить время исполнения программы на различном количестве процессоров.

Программа корректно работает для любого допустимого входного потока.

Явный вид формул декомпозиции и реконструкции позволяет вычислять элементы основного и вэйвлетного потоков попарно, что позволяет выполнять вычисления параллельно.

Приведем график скорости работы в зависимости от размера исходного потока на одном процессоре:



В данной работе был реализован алгоритм сжатия цифровой информации на основе сплайн-вэйвлетного разложения первого порядка. Он может применяться для многих задач сжатия на практике. Вычисления по предложенным формулам декомпозиции и реконструкции не зависят друг от друга при различных значениях фигурирующих в них индексов; это позволяет эффективно распараллелить вычислительный процесс.

Литература

1. Демьянович Ю.К. Всплески & минимальные сплайны. Курс лекций. СПб., 2003. 200 с.
2. Демьянович Ю.К., Ходаковский В.А. Введение в теорию вэйвлетов. Учеб. пособие. СПб.: Петербургский государственный университет путей сообщения, 2008. — 51 с.
3. Демьянович Ю.К., Дронь В.О. О вэйвлетном гребне на нерегулярной сетке // Проблемы математического анализа 2011. (в печати).
4. Демьянович Ю.К., Мирошниченко И.Д. Гнездовые сплайн-вэйвлетные разложения // Проблемы математического анализа 2012. Вып. 64. С. 51–61.

ОЦЕНКИ АППРОКСИМАЦИИ НЕПОЛИНОМИАЛЬНЫМИ СПЛАЙНАМИ ВТОРОГО ПОРЯДКА¹

Ю.К. Демьянович

Yuri.Demjanovich@gmail.com

О.Н. Иванцова

ivantsova@front.ru

Аннотация: В работе дано представление остатка аппроксимации интерполяционными неполиномиальными сплайнами лагранжева типа и получены оценки погрешности аппроксимации с указанием констант. Упомянутые оценки точны на компонентах порождающей вектор-функции $\varphi(t)$.

1. Введение

Известно, что для полиномиальных сплайнов оценки аппроксимации получаются из принципа подобия (см., например, [1]). Для неполиномиальных сплайнов второго порядка такие оценки получены в работе [2] обработкой асимптотик координатных функций; на этом пути пришлось преодолеть значительные аналитические трудности.

В данной работе применяется другой прием; в результате полученные оценки точны на компонентах порождающей векторфункции $\varphi(t)$ и содержат легко вычисляемые константы.

2. Представление остатка аппроксимации

На интервале (α, β) рассмотрим сетку вида

$$X : \dots < x_{-1} < x_0 < x_1 < \dots,$$

$$\lim_{j \rightarrow -\infty} x_j = \alpha, \quad \lim_{j \rightarrow +\infty} x_j = \beta.$$

Положим

$$G \stackrel{\text{def}}{=} \bigcup_{j \in \mathbb{Z}} (x_j, x_{j+1}), \quad h \stackrel{\text{def}}{=} \sup_{j \in \mathbb{Z}} (x_{j+1} - x_j).$$

¹ Работа частично поддержана грантом РФФИ 13-01-00096

Пусть $s \in \{1, 2\}$, а $\varphi(t) = (\varphi_0(t), \varphi_1(t), \varphi_2(t))^T$ — трехкомпонентная вектор-функция с компонентами из пространства $C(\alpha, \beta)$, причем функции $\varphi_0(t), \varphi_1(t), \varphi_2(t)$ образуют чебышевскую систему на (α, β) .

Определим функции $\omega_j(t)$ из аппроксимационных соотношений

$$\text{supp } \omega_j \subset [x_{j-s}, x_{j+3-s}),$$

$$\begin{aligned} \varphi(x_{k+s-2})\omega_{k+s-2}(t) + \varphi(x_{k+s-1})\omega_{k+s-1}(t) + \varphi(x_{k+s})\omega_{k+s}(t) &= \varphi(t) \\ \forall t \in (x_k, x_{k+1}) \quad \forall k \in \mathbb{Z}. \end{aligned}$$

Известно (см., например, [1]), что функции $\omega_j(t)$ определяются однозначно, они линейно независимы, непрерывны и образуют интерполяционный базис, так что $\omega_j(x_i) = \delta_{ij}$.

Рассмотрим линейное пространство $\mathbb{S}_{X,\varphi} \stackrel{\text{def}}{=} Cl_p \mathcal{L}\{\omega_j\}$, где $\mathcal{L}\{\dots\}$ — линейная оболочка множества элементов, находящихся в фигурных скобках, а Cl_p — замыкание этой линейной оболочки в топологии поточечной сходимости. Пространство $\mathbb{S}_{X,\varphi}$ называется пространством элементарных (X, φ) -сплайнов лагранжева типа.

Пусть

$$A_k \stackrel{\text{def}}{=} (\varphi(x_{k+s-2}), \varphi(x_{k+s-1}), \varphi(x_{k+s})), \quad \langle g_i, u \rangle \stackrel{\text{def}}{=} u(x_i).$$

Рассмотрим интерполяцию $\tilde{u}(t)$ функции $u \in C(\alpha, \beta)$ в узлах сетки X :

$$\tilde{u}(t) \stackrel{\text{def}}{=} \sum_{j \in \mathbb{Z}} u(x_j) \omega_j(t), \quad t \in (\alpha, \beta).$$

Теорема 1. При $t \in [x_k, x_{k+1}]$, $k \in \mathbb{Z}$, верно тождество

$$u(t) - \tilde{u}(t) = \det A_k^{-1} \begin{vmatrix} \varphi(x_{k+s-2}) & \varphi(x_{k+s-1}) & \varphi(x_{k+s}) & \varphi(t) \\ u(x_{k+s-2}) & u(x_{k+s-1}) & u(x_{k+s}) & u(t) \end{vmatrix};$$

здесь второй сомножитель правой части — определитель квадратной матрицы четвертого порядка, записанной в блочном виде.

3. Оценки аппроксимации сплайнами лагранжева типа

Предположим, что $u, \varphi \in C^3[\alpha, \beta]$ и выполнено условие (А)

$$|\det(\varphi, \varphi', \varphi'')(t)| \geq c > 0. \tag{3.1}$$

Ввиду равномерной непрерывности рассматриваемых функций на отрезке $[a, b]$ из (3.1) заключаем, что для любого $\varepsilon \in (0, c)$ существует такое $h_0(\varepsilon)$, что при $h \in (0, h_0(\varepsilon))$ и $\xi_1, \xi_2 \in [x_{k+s-2}, x_{k+s}]$ верно неравенство

$$|\det(\varphi(x_k), \varphi'(\xi_1), \varphi''(\xi_2))(t)| \geq c_\varepsilon > 0, \quad (3.2)$$

где $c_\varepsilon = c - \varepsilon$.

Теорема 2. Если $u, \varphi \in C^3[\alpha, \beta]$ и выполнено неравенство (3.2), то при $t \in (x_k, x_{k+1})$ справедлива оценка

$$|u(t) - \tilde{u}(t)| \leq \frac{1}{6c_\varepsilon} \cdot |(x_{k+s-2} - t)(x_{k+s-1} - t)(x_{k+s} - t)| \cdot D_k(\varphi, u),$$

где

$$D_k(\varphi, u) \stackrel{\text{def}}{=} \max \left| \det \begin{pmatrix} \varphi(x_{k+s-2}) & \varphi'(\xi_1) & \varphi''(\xi_2) & \varphi'''(\xi_3) \\ u(x_{k+s-2}) & u'(\xi_1) & u''(\xi_2) & u'''(\xi_3) \end{pmatrix} \right|,$$

а максимум берется при условиях $\xi_1, \xi_2, \xi_3 \in [x_{k+s-2}, x_{k+s}]$.

Следствие 1. Интерполяция $\tilde{u}(t)$ функции $u(t)$ точна на элементах пространства $\Phi \stackrel{\text{def}}{=} \mathcal{L}\{\varphi_0, \varphi_1, \varphi_2\}$, т.е. $u(t) \equiv \tilde{u}(t) \quad \forall t \in (\alpha, \beta) \quad \forall u \in \Phi$.

Литература

1. Бурова И.Г., Демьянович Ю.К. Теория минимальных сплайнов. Издательство СПбГУ. 2000. 316 с.
2. Демьянович Ю.К. Минимальные сплайны лагранжева типа//Сб. Проблемы математического анализа. Вып.50. 2010. С.21–64.

Теория и практика кодирования информации



Крук
Евгений Аврамович

д.т.н., профессор
декан факультета информационных систем и защиты информации
заведующий кафедрой
безопасности информационных систем СПбГУАП



Трифонов
Петр Владимирович

к.т.н., доцент СПбГПУ

ДЕКОДИРОВАНИЕ ПОЛЯРНЫХ КОДОВ С ПОМОЩЬЮ АЛГОРИТМА ДУМЕРА-ШАБУНОВА

Морозов Р.А.

*студент кафедры распределённых вычислений и компьютерных сетей
СПбГПУ, rmorozov@dcn.fik.spbstu.ru*

Аннотация: Рассматривается задача списочного декодирования полярных кодов. Предложена модификация алгоритма Думера-Шабунова декодирования кодов Рида-Маллера, обеспечивающая большую корректирующую способность по сравнению с методом последовательного исключения Арикана.

Введение

Полярные коды — первый класс кодов, достигающих пропускной способности канала при длине кода, стремящейся к бесконечности, имеющий сложность кодирования и декодирования $O(n \log n)$, где n — длина кода. Однако корректирующая способность полярных кодов с практически значимыми длинами оказывается значительно хуже, чем, например, в случае кодов с малой плотностью проверок на чётность (LDPC). Одной из причин этого является субоптимальность метода последовательного исключения, предложенного Ариканом для декодирования полярных кодов. В данной работе представлена модификация алгоритма Думера-Шабунова декодирования кодов Рида-Маллера, использование которой обеспечивает намного большую корректирующую способность.

Алгоритм Думера-Шабунова

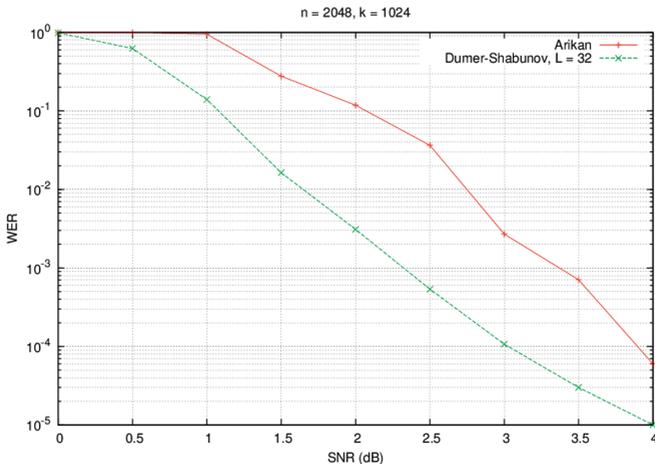
Алгоритм Думера-Шабунова декодирования кодов Рида-Маллера использует их представление, основанное на конструкции Плоткина. Задача декодирования кода $RM(r, m)$ рекурсивно сводится к декодированию кодов с повторением и тривиальных кодов со скоростью 1. Для каждого из них формируется список из не более чем 4 наиболее вероятных кодовых слов.

Декодирование полярных кодов

Полярные коды являются подкодами кода Рида-Маллера, поэтому для их декодирования можно также применить алгоритм Думера-Шабунова с небольшими модификациями. Для заданного полярного кода $P(m, k)$ длины 2^m и размерности k находится наименьший код $RM(r, m)$, содержащий в себе данный код. Таким образом, заданный полярный код — это код Рида-Маллера с некоторыми «замороженными» позициями в информационном слове, т.е.

некоторые информационные символы принимают заведомо известные значения (например, нули). Вследствие этого некоторые коды с повторениями, возникающие в разложении кода $RM(r, m)$, преобразуются в коды с нулевой скоростью, а некоторые коды со скоростью 1 перестают быть тривиальными. Для декодирования полученных таким образом кодов $(2^g, 2^g - f)$, где f — число замороженных символов в этом коде, предлагается воспользоваться универсальным алгоритмом декодирования линейных блочных кодов *Vox & Match*. В процессе декодирования этот алгоритм перебирает множество наиболее вероятных кодовых слов, отсекая слова с меньшим эллиптическим весом, чем у лучшего найденного на данный момент слова. Для получения списка наиболее вероятных кодовых слов была предложена модификация этого алгоритма, состоящая в том, что в качестве порога отсечения используется эллиптический вес s -го наиболее вероятного уже найденного кодового слова.

На рисунке представлен график зависимости вероятности ошибки на кодовое слово от отношения сигнал-шум в канале с АБГШ для стандартного алгоритма декодирования полярных кодов и модифицированного алгоритма Думера-Шабунова.



Литература

1. *E. Arikan*. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. // IEEE Transactions on Information Theory, Vol. 55, No. 7. — Июль 2009. — С. 3051–3073.
2. *I. Dumer, K. Shabunov*. Soft-Decision Decoding of Reed–Muller Codes: Recursive Lists. // IEEE Transactions on Information Theory, Vol. 52, No. 3. — Март 2006. — С. 1260–1266.
3. *A. Valembois, M. Fossorier*. Box and Match Techniques Applied to Soft-Decision Decoding. // IEEE Transactions on Information Theory, Vol. 50, No. 5. — Май 2004. — С. 796–810.

ОБОБЩЕНИЕ НАДЕЖНЫХ КОДОВ

Алексеев М.О.

*аспирант кафедры аэрокосмических компьютерных технологий СПбГУАП,
alexeev@vu.spb.ru*

Еганын А.В.

инженер института ВКуСТ, СПбГУАП, eganyan.artur@gmail.com

Аннотация: В данной работе ставится задача защиты криптографических устройств от алгебраических манипуляций — изменения обрабатываемых данных. Приводятся существующие методы защиты, после чего описывается новая кодовая конструкция, являющаяся обобщением надежных кодов, обнаруживающих алгебраические манипуляции. Производится сравнение данной конструкции с аналогами.

Введение

Классическим подходом к обеспечению защиты вычислительных устройств и каналов связи является использование линейных кодов [1]. Линейные коды концентрируют свои обнаруживающую и корректирующую способности на определенных конфигурациях ошибок (в том числе, ошибках заданной кратности). Простота реализации алгоритмов кодирования и декодирования при условии согласованности кода с каналом обеспечивают высокую эффективность этого подхода.

При условии, что конфигурация ошибок не поддается строгому описанию, использование линейных кодов не способно обеспечить необходимый уровень защиты данных. Например, если конфигурация ошибок контролируется злоумышленником, то, воспроизводя ошибки, кратность которых заведомо превышает корректирующие или исправляющие способности используемого кода, злоумышленник может успешно атаковать вычислительное устройство или канал передачи данных, внедряя ошибки [2].

Одной из моделей внесения ошибки является модель алгебраической манипуляции. В этой модели подразумевается, что злоумышленник способен изменять значение некоторого абстрактного устройства хранения данных, не имея при этом доступа на чтение этих данных. Данная модель может использоваться как для описания атак защищенной памяти [9, 11], так и для других систем, например схем разделения секрета с мошенниками [12].

Применительно к защищенной памяти алгебраическая манипуляция представляет собой некоторую аддитивную ошибку, воздействующую на содержимое памяти. Конфигурация ошибок в данном случае абсолютно непредсказуема и зависит от возможностей злоумышленника и метода возбуждения помех. Далее понятия ошибки, атаки и алгебраической манипуляции будут рассматриваться как синонимы.

Естественной областью применения алгебраических манипуляций является криптоанализ и, в частности, атаки с привнесением помех. Атаки с привнесением помех являются одним из наиболее эффективных типов атак по сторонним каналам, используемых для получения дополнительной информации о реализации конкретного криптографического алгоритма. Алгебраические манипуляции могут приводить к ошибкам вычисления конкретных блоков криптографического устройства. Далее, работа устройства в условиях помех анализируется, например, с помощью метода дифференциального анализа помех [3]. Данный подход к взлому криптографических алгоритмов является крайне эффективным, на данный момент большинство симметричных шифров было взломано с его помощью, например AES [4, 5].

Коды, обнаруживающие алгебраические манипуляции

Надежные коды

Для защиты от алгебраических манипуляций был разработан класс надежных кодов, обнаруживающих алгебраические манипуляции [6]. Эти коды определяются как коды, которые обнаруживают любую конфигурацию ошибок с заданной вероятностью. Это выполняется за счет использования совершенно нелинейных (perfect nonlinear, PN) и почти совершенно нелинейных (almost perfect nonlinear, APN) функций для вычисления проверочных символов кода [8].

Для систематического надежного кода со скоростью $1/2$, кодовое слово которого может быть записано как $(x|f(x))$, вероятность обнаружения любой ошибки $e \neq 0$ ограничена сверху следующим выражением: $P_{\text{det}} \leq 1 - P_f$, где через P_f обозначена степень нелинейности используемой функции $f(x)$ [8]. Минимальное значение нелинейности для функций $f(x): GF(p^k) \rightarrow GF(p^r)$ равно $P_f = p^{-r}$ при $p \neq 2$. Такие функции называются совершенно нелинейными функциями. Функции, достигающие минимальное значение $P_f = 2^{-r+1}$ при $p = 2$, называются почти совершенно нелинейными.

Надежные коды эффективны при условии, что злоумышленник не имеет информации о закодированном сообщении. В противном случае, надежность этих кодов сводится к нулю.

Сильно защищенные коды, обнаруживающие алгебраические манипуляции

Для защиты данных при сильной модели атаки, когда злоумышленник обладает информацией об обрабатываемых данных или даже может их контролировать, были предложены сильно защищенные коды, обнаруживающие алгебраические манипуляции (strongly secure algebraic manipulation codes) [7, 9]. Для краткости будем называть их AMD кодами.

Для построения AMD кодов используются различные математические объекты: коды аутентификации, разностные структуры, помехоустойчивые коды [7]. Одной из наиболее развитых и эффективных конструкций сильных AMD кодов является конструкция, основанная на полиномах. Наиболее полно этот класс AMD кодов описан в [9]. Значительная часть конструкций, предлагаемых в этой работе, являются оптимальными в смысле вероятности обнаружения ошибки.

Далее будут рассматривать коды над полями характеристики 2. Кодовые слова систематического AMD кода представляют собой конкатенацию информационного сообщения $y \in GF(2^k)$, некоторого случайного числа $x \in GF(2^m)$ и значения нелинейной функции $f(y, x) \in GF(2^r)$, заданной в виде полинома. Сами AMD коды определяются как коды, для которых не существует такой конфигурации ошибок $e = (e_y, e_x, e_f)$ и такого значения y , при возникновении которых равенство $f(y, x) + e_f = f(y + e_y, x + e_x)$ выполнится при всех возможных значениях x . Данное равенство называется уравнением маскирования ошибки (УМО). Способность AMD кода обнаруживать ошибки напрямую зависит от вида его УМО: максимальное количество решений УМО относительно x для всех возможных комбинаций y, e_x, e_y, e_f и будет максимальным количеством необнаруживаемых кодом ошибок. Вероятность обнаружения ошибки ограничена снизу следующим выражением:

$$P_{\text{det}} \geq 1 - \frac{|\{x: f(y, x) + e_f = f(y + e_y, x + e_x)\}|}{2^m}.$$

Сильно защищенные коды на базе надежных кодов

Кодовая конструкция

В данной статье предлагается новая конструкция AMD кодов, основанная на надежных кодах, обнаруживающих ошибки.

Как уже было сказано, слабостью надежных кодов при условии сильной модели атаки является детерминированность вычисляемой проверочной части. Наиболее естественным методом её устранения является привнесение случайности в кодируемое сообщение.

Рассмотрим следующую кодовую конструкцию.

Пусть используется надежный систематический код $(x | f(x))$, где $f(x): GF(2^r) \rightarrow GF(2^r)$ — некоторая APN функция. Пусть вектор x представляет собой конкатенацию информационного сообщения $a \in GF(2^k)$ и случайного числа $b \in GF(2^m)$, или $r = k + m$.

Таким образом, кодовое слово рассматриваемого кода можно записать как

$$(x | f(x)) = (a | b | f(a | b)).$$

Теорема. Данный код при $m > 1$ является сильным AMD кодом с вероятностью обнаружения ошибки $P_{\text{det}} \geq 1 - 2^{-m+1}$.

Доказательство:

Как уже было написано, обнаруживающая способность AMD кода напрямую зависит от максимального количества корней его уравнения маскирования ошибки. УМО рассматриваемого кода выглядит следующим образом:

$$f(x) + e_f = f(x + e_x).$$

Проанализируем количество корней уравнения маскирования ошибки. Согласно теории надежных кодов, количество решений УМО не превышает двух (за счет использования APN функции). То есть, для любой конфигурации ошибок $e = (e_x, e_f)$ существует не более двух значений x , при которых выполнится равенство в УМО.

Так как вектор x представляет собой конкатенацию векторов a и b , существует не более двух их комбинаций, при которых ошибка не будет обнаружена. Следовательно, существуют не более двух значений b , при которых алгебраическая манипуляция не будет обнаружена. В случае, когда b может принимать больше двух значений ($m > 1$), для данного кода не существует необнаруживаемых ошибок, любая ошибка обнаруживается с вероятностью не менее $P_{\text{det}} \geq 1 - \frac{2}{2^m} = 1 - 2^{-m+1}$. ■

Рассмотрим пример: пусть $k=4$, $m=2$, тогда $r=k+m=6$. В качестве функции кодирования выберем возведение в третью степень в поле Галуа, $f(x) = f(a, b) = (a|b)^3$. Итоговый код будет иметь скорость $R = k/(k+m+r) = 1/3$, гарантируя при этом вероятность обнаружения любой сильной атаки (алгебраической манипуляции) $P_{\text{det}} \geq 0.5$.

Стоит отметить, что при равномерно распределенной ошибке $e \neq 0 \in GF(2^n)$ (слабая модель атаки) вероятность обнаружения ошибки при использовании предлагаемого кода будет равна $P_{\text{det}} \geq 1 - 2^{1-r}$ против $P_{\text{det}} \geq 1 - 2^{1-m}$ в случае сильной атаки. Это следует из свойств надежных кодов [6].

Сравнение с AMD-кодами на базе обобщенных кодов PM

В данном разделе приводится сравнение описываемой кодовой конструкции с AMD кодами из [9], которые основаны на обобщенных кодах Риды-Маллера.

При фиксированных параметрах k и m коды из [9] обеспечивают вероятность обнаружения ошибки $P_{\text{det}} \geq 1 - (\lceil k/m \rceil + 1)/2^m$. Из этого следует, что при $k \leq m$ сравниваемые коды обладают одинаковой вероятностью обнаружения ошибок. В случае $k > m$ предлагаемый код обеспечивает более высокую вероятность обнаружения ошибки. В обоих случаях избыточность предлагаемого кода больше на k бит.

Сравнение кодов с одинаковой скоростью. Пусть есть описанный код с $y \in GF(2^k)$, $x \in GF(2^m)$ и $f(x, y) \in GF(2^{r-k+m})$, $P_{\text{det}} \geq 1 - 2^{-m+1}$. Согласно [11] можно построить код (увеличив размер случайного числа и проверочной части так, чтобы скорость кодов совпадали) с $y \in GF(2^k)$, $y \in GF(2^{m+\lfloor k/2 \rfloor})$ и $f(x, y) \in GF(2^{m+\lfloor k/2 \rfloor})$, вероятность обнаружения ошибки которого ограничена снизу величиной $P_{\text{det}} \geq 1 - (\lceil k/(m + \lfloor k/2 \rfloor) \rceil + 1)/2^{m+\lfloor k/2 \rfloor}$. Из этого следует, что при заданной скорости кода конструкция из [9] демонстрирует более высокий уровень защиты от алгебраических манипуляций.

Стоит отметить, что при слабой модели атаки, когда вносимые ошибки равновероятны, предлагаемый код не обнаруживает ошибки с вероятностью в 2^k раз меньше, чем коды из [9].

Заключение

Данную кодовую конструкцию можно рассматривать как обобщение надежных кодов на случай сильной модели атаки. С увеличением размера r случайного числа обеспечивается рост вероятности обнаружения сильных атак. Частным случаем данной конструкции при $m=0$ являются надежные коды, обнаруживающие только слабые атаки. Несмотря на то, что кодовые конструкции из [11] обладают меньшей избыточностью, обобщенные надежные коды обладают рядом преимуществ:

- возможность и простота динамического изменения скорости кода за счет изменения соотношения количества информационных и случайных битов в кодируемом сообщении. Фактически, кодек обобщенных надежных кодов может содержать в себе некоторый набор кодов с разными характеристиками. Данная особенность позволяет использовать один кодек для различных типов памяти или различных режимов работы устройства;
- процедура исправления повторяющихся ошибок, описанная для надежных кодов в [10]. За счет нее предлагаемый код способен исправлять ошибки, которые проявляются на 3 и более тактах работы устройства, за счет решения системы уравнений. Подобная процедура для кодов из [9] требует значительных аппаратных, информационных и вычислительных затрат [11];
- высокая помехоустойчивость при слабой модели атаки.

Описанный обобщенный надежный код совмещает в себе как надежный, так и сильно защищенный AMD коды, обеспечивая вероятность обнаружения ошибок $P_{\text{det}}^{(w)} \geq 1 - 2^{-r+1}$ и $P_{\text{det}}^{(s)} \geq 1 - 2^{-m+1}$, соответственно.

Литература

1. J. MacWilliams and N.J.A. Sloane. The Theory of Error-Correcting Codes. North-Holland, Amsterdam, 1977, 762 p.

2. *J.J. Quisquater and F. Koeune*. Side channel attacks. Report, October 2003.
 3. *Biham E., Shamir A.*, Provable Security Against Differential Cryptography // *Crypto 1990, Lecture Notes in Computer Science*. 1991. V. 537. P. 2–21.
 4. *C.N. Chen and S.M. Yen*. «Differential Fault Analysis on AES Key Schedule and Some Countermeasures». *ACISP 2003, LNCS 2727*, pp. 18–129, 2003.
 5. *P. Dusart, G. Letourneux, O. Vivolo*. «Differential Fault Analysis on AES». *Cryptology ePrint Archive*, Report 2003/010.
 6. *K.D. Akdemir, Z. Wang, M.G. Karpovsky, and B. Sunar*. «Design of Cryptographic Devices Resilient to Fault Injection Attacks Using Nonlinear Robust Codes». *Fault Analysis in Cryptography*, M. Joye Editor, 2011.
 7. *E. Jongsma*. «Algebraic Manipulation Detection Codes». *Bachelorscriptie, Mathematisch Instituut, Universiteit Leiden*, 6 maart 2008.
 8. *М.Э. Тужилин*. Почти совершенные нелинейные функции // *ПДМ*, 2009, №3, 14–20.
 9. *Z. Wang and M.G. Karpovsky*, «Algebraic Manipulation Detection Codes and Their Application for Design of Secure Cryptographic Devices», *Proc of Int. Symp. on On-Line Testing*, 2011.
 10. *K. Kulikowski, M.G. Karpovsky*, «Robust Correction of Repeating Errors by Non-linear Codes», *Communications, IET*, Vol 5 pp. 2317–2327, 4, 2011.
 11. *Z. Wang and M.G. Karpovsky*, «Reliable and Secure Memories Based on Algebraic Manipulation Correction Codes», *Proc Int Symp. on On-line Testing*, June 2012.
 12. *R. Cramer, Y. Dodis, S. Fehr, C. Padro, and D. Wichs*, «Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors,» in *Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology, ser. EUROCRYPT'08*, 2008, pp. 471–488.
-

СХЕМА СЖАТИЯ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ С ПРИМЕНЕНИЕМ АДАПТИВНОГО ДВОИЧНОГО АРИФМЕТИЧЕСКОГО КОДИРОВАНИЯ

Егоров Н.Д.

*студент кафедры безопасности информационных систем ГУАП,
negorov.91@gmail.com*

Соколовский Р.А.

*студент кафедры безопасности информационных систем ГУАП,
sokolovskiy.roman@gmail.com.*

Аннотация: В работе была предложена схема кодирования изображений без потерь на основе бинаризации и контекстного моделирования с использованием двоичного арифметического кодирования. Приведены результаты экспериментов, подтверждающие эффективность вносимых улучшений. Представлено сравнение улучшенной схемы с существующими решениями.

Введение

В современном мире во многих областях деятельности требуется работать с изображениями в высоком разрешении. В частности такие изображения используются в медицине. Данные изображения, как правило, занимают значительное место на жестком диске и требуют сжатия. При этом операция сжатия должна быть полностью обратимой, т. е. мы должны иметь возможность восстановления исходного изображения без потерь.

При сжатии изображений без потерь приходится сталкиваться с двумя основными проблемами:

1. наличие избыточности, обусловленной статистической зависимостью кодируемых данных. В цветных изображениях будем выделять пространственную (между пикселями одной компоненты) и межкомпонентную избыточность;
2. нестационарность потоков данных, обусловленная изменением статистических свойств.

В рамках данной работы будут рассмотрены типовое решение и ряд улучшений для уже существующей схемы сжатия без потерь, учитывающих статистические особенности медицинских цветных изображений.

1. Обзор существующих подходов к кодированию цветных изображений без потерь

В качестве типового решения рассмотрим схему кодирования, представленную на Рис. 1.

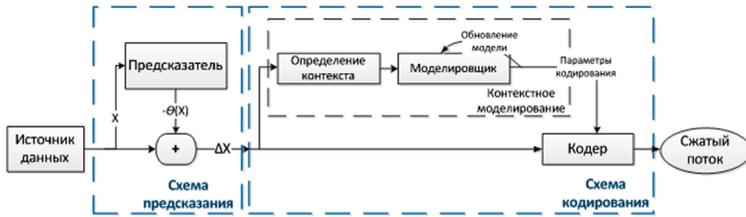


Рис. 1. Типовая схема кодирования изображений без потерь

Схема предсказания отвечает за решение проблемы избыточности данных. В результате кодироваться будет не само значение кодируемой величины, а разность между ним и предсказанным значением.

Схема кодирования решает проблемы избыточности и нестационарности данных. Для решения описанных проблем в схеме кодирования используется процедура контекстного моделирования с последующей адаптацией к нестационарности потоков данных. Под контекстной моделью на этапе энтропийного кодирования понимается правило присвоения вероятности текущему символу на основе некоторой предыстории.

Среди существующих решений можно выделить двух лидеров:

- JPEG-LS [1] — кодек для сжатия изображений. Показал высокую эффективность при сжатии медицинских изображений. Время работы кодека существенно меньше, чем у альтернативных решений.
- RAQ [2] — универсальный кодек для сжатия данных. Показал наивысшие результаты при сжатии медицинских изображений. Время работы на порядок больше, чем у конкурентов.

В рамках предыдущей работы [3] была доказана эффективность использования декоррелирующих цветовых преобразований совместно с JPEG-LS кодеком. В данной работе за основу берется схема с альтернативным предсказанием Inter Component Prediction (ICP) [4] и схема кодирования JPEG-LS кодека с отключенным пространственным предсказанием. Данная схема представлена на Рис. 2.

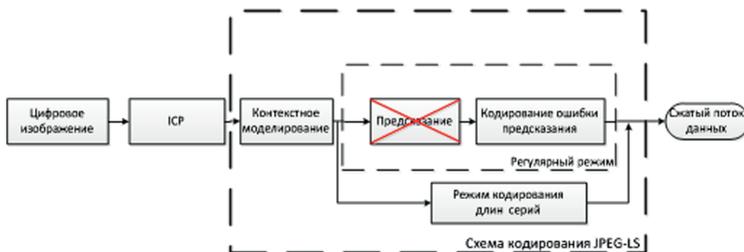


Рис. 2. Базовая схема кодирования

2. Улучшения схемы кодирования изображений без потерь

В качестве улучшений к существующей схеме кодирования было предложено:

1. добавление процедуры двоичного адаптивного арифметического кодирования;
2. изменение существующей процедуры определения контекста;
3. изменение режима кодирования длин серий.

2.1. Добавление процедуры двоичного арифметического адаптивного кодирования

Для работы арифметического кодирования необходима оценка вероятностей для каждого кодируемого символа. Способ получения оценок вероятностей не регламентирован и может быть выбран с учетом нестационарности потока данных и отличительных статистических особенностей кодируемого источника. Оценка вероятности появления кодируемого символа производится по формуле (1) Кричевского-Трофимова [5], где n_i — это счётчик попадания i -го символа в контекст. Для учёта нестационарности статистических характеристик обрабатываемого потока используется схема с экспоненциальным забыванием [6]. В рамках этой схемы при достижении порогового значения производится уменьшение счётчиков попадания символов в контекст в несколько раз. Подобная схема уменьшает влияние предыдущей статистики по экспоненциальному закону, что позволяет энтропийному кодеку приспосабливаться к изменениям статистики в потоке.

$$p_1 = \frac{n_1 + 1/2}{n_0 + n_1 + 1} \quad (1)$$

Двоичное арифметическое кодирование используется в виду вычислительной простоты по сравнению с недвоичным арифметическим кодированием. Также существуют уже описанные эффективные реализации двоичного арифметического кодирования [7]. Но при переходе к использованию двоичного арифметического кодера для кодирования символов недвоичного источника данных необходимо вводить процедуру бинаризации — представления кодируемых символов в виде строки двоичных символов.

При использовании арифметического кодирования ключевым параметром эффективности бинаризации будет сумма энтропий кодируемых двоичных источников, а не средняя длина двоичной последовательности бинаризованного источника. В качестве бинаризации возможно использование унарного кода, так как для данного способа доказана справедливость равенства из формулы (2).

$$\geq (\text{исходный источник}) = \sum \geq (\text{двоичный источник}) \quad (2)$$

Унарный код — двоичный префиксный код переменной длины для представления натуральных чисел. Операция кодирования унарным кодом некоторого числа показана на Рис. 3.

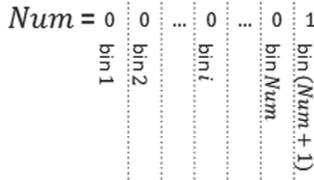


Рис. 3. Кодирование унарным кодом

2.2. Изменение существующей процедуры контекстного моделирования

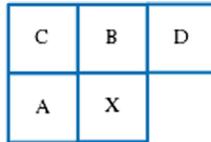


Рис. 4. Контекст рассматриваемого значения X

В кодеке JPEG-LS контекстное моделирование производится на основе квантованных значений локальных градиентов (3) соседних пикселей, представленных на Рисунке 4. Квантованные градиенты формируют 365 контекстов. Моделирование выполняется на этапе предсказания и является его частью.

$$\begin{cases} g_1 = d - b \\ g_2 = b - c \\ g_3 = c - a \end{cases} \quad (3)$$

При переходе к новой схеме предсказания ICP исчезла необходимость в коррекции ошибок предсказания: гистограмма ошибок имеет чётко выраженную концентрацию в нуле. Значение ошибки предсказания кодируется в следующем контексте:

$$\text{ctx} = \max[BC(a), BC(b), BC(c), BC(d)], \quad (2)$$

где $BC(x)$ — это индекс старшего значащего бита числа «x», который будем называть битовой категорией. Битовая категория для ошибок предсказания может принимать девять различных значений, поэтому для кодирования ошибок предсказания используется девять контекстных потоков.

2.3. Изменение режима кодирования длин серий

Режим кодирования длин серий используется для эффективного кодирования повторяющихся значений. При использовании этого режима в битовый поток укладывается информация о значении текущего символа и о количестве его повторений.

Благодаря эффективной работе схемы предсказания в кодируемом потоке данных наблюдается группировка множественных нулевых значений. Режим кодирования длин серий был скорректирован для обработки таких последовательностей.

Был также изменён блок принятия решений относительно использования режима кодирования длин серий: для каждой компоненты решение принимается на основе её статистических свойств. Кодирование длин серий используется в случае, если дисперсия значений ошибок внутри компоненты оказывается ниже некоторого порога.

3. Результаты проведенных экспериментов

Предлагаемая схема кодирования представлена на Рис. 5.

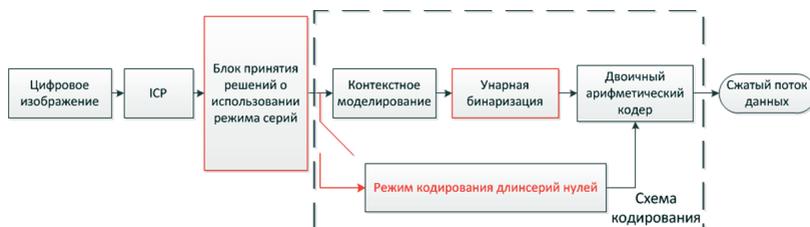


Рис. 5. Предлагаемая схема кодирования

Представленная схема проходила проверку на тестовом множестве, состоящем из 332 изображений:

- 95 оригинальных медицинских изображений (кардиограммы, МРТ, рентгеновские снимки);
- 95 уменьшенных и 95 увеличенных в несколько раз копий медицинских изображений;
- 41 цветных фотореалистичных изображений.

Результаты проведенных экспериментов приведены в таблице 1.

Оценка выигрыша по сжатию рассматриваемого кодека производилась согласно формуле (5).

$$TG(XJPEG-LS) = \frac{\text{Сжатые данные } X - \text{Сжатые данные } JPEG-LS}{\text{Сжатые данные } JPEG-LS} * 100, \quad (5)$$

где $Total\ Gain(TG)$ — величина, отражающая выигрыш кодека X по сжатию в процентах относительно кодека $JPEG-LS$.

Дополнения к схеме кодирования	Размер сжатого тестового множества (байт)	TG (X, JPEG-LS), %	Время обработки тестового множества
JPEG-LS + цветное преобразование	596430373	38,7	6 мин
JPEG-LS + ICP	569600771	41,4	13 мин
Предлагаемая схема	530948899	45,4	14 мин
PAQ	441992366	54,5	5ч. 35мин.

Таблица 1. Результаты сжатия опробованных схем кодирования.

Заключение

В данной работе были представлены методы улучшения схемы кодирования, связанные с применением процедуры адаптивного двоичного арифметического кодирования и изменением контекстного моделирования. Как видно из результатов экспериментов, эффективность сжатия предлагаемой схемы приблизилась к результатам PAQ, но при этом сохраняет существенное преимущество в скорости.

Литература

1. ITU-T Recommendation T.87/ISO 14495-1 «Information technology — Lossless and near-lossless compression of continuous-tone still images — Baseline», 06. 1998.
2. *Matthew V. Mahoney*, «The PAQ1 Data Compression Program», Draft, Jan. 20, 2002, revised Mar. 19.
3. *Егоров Н.Д.* «Анализ обратимых преобразований компонент цветных изображений», ГУАП, СНК 2012
4. *Новиков Д.В.* «Метод сжатия цветных изображений без потерь», ГУАП, СНК 2012.
5. *R.E. Krichevski and V.E. Trofimov*, «The performance of universal encoding» IEEE Trans. Inform. Theory, vol. IT-27, pp 199–207, Mar. 1981.
6. *Eado Meron, Meir Feder, Fellow*, «Finite-Memory Universal Prediction of Individual Sequences», IEEE Transactions on Information Theory, vol. 50, NO.7, JULY 2004, pp 1506–1523
7. *Detlev Marpe, Heiko Schwarz, and Thomas Wiegand* «Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard», IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 13, NO. 7, JULY 2003

О ЗАДАЧЕ ВЫБОРА ПАРАМЕТРОВ ПРЕОБРАЗОВАНИЯ НА ОСНОВЕ ПОЛИНОМОВ ЧЕБЫШЕВА С ЦЕЛЬЮ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ СЖАТИЯ ИЗОБРАЖЕНИЙ

Филиппов Б.С.

асп. кафедры инфокоммуникационных систем ГУАП, FilippovBoris@gmail.com

Егоров Н.Д.

*студент кафедры безопасности информационных систем ГУАП,
negorov.91@gmail.com*

Аннотация: Данная работа посвящена изучению преобразования, основанного на ортогональных полиномах Чебышева. Предложен эвристический метод выбора параметров преобразования, для обеспечения более эффективного кодирования. Приведены примеры, подтверждающие рациональность предложенного метода. Представлено сравнение исследуемого метода кодирования с кодированием по стандарту JPEG [1, 2].

Введение

В большинстве современных схем кодирования изображений в качестве декоррелирующего преобразования используется Дискретное Косинусное Преобразование (ДКП). В данной работе рассматривается альтернативное преобразование, основанное на применении разложения по базису ортогональных полиномов Чебышева, схожего по своим свойствам с ДКП. Использование этого преобразования в задачах кодирования изображений обладает своими преимуществами и недостатками. К преимуществам можно отнести возможность выбора произвольного размера блока восстанавливаемого изображения, а также более высокие показатели PSNR на низких битовых скоростях по сравнению с ДКП. Недостатком являются дополнительные ошибки, вносимые преобразованием. Данная работа посвящена анализу этих ошибок. Предлагается метод выбор параметров преобразования для уменьшения вносимой ошибки.

Преобразование на основе полиномов Чебышева

В серии работ Ю.С. Радченко ([3], [4] и др.) представил метод кодирования изображений, основанный на разложении сигнала по базису ортогональных полиномов Чебышева [5]. Чтобы описать процедуру кодирования, введём понятия изображения, сигнала и отсчётов. Под отсчётами будем понимать значения непрерывной функции на конечном множестве точек.

Будем рассматривать изображение как двухмерный непрерывный сигнал $s(x, y)$, для которого известны только отсчёты в узлах равномерной сетки, что соответствует пикселям изображения. Тогда суть предложенного метода кодирования изображений будет заключаться в разложении сигнала $s(x, y)$ по базису ортогональных полиномов Чебышева. Формулы прямого и обратного преобразований для одномерного случая имеют вид:

$$C_m = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} s(z_n) \cos\left(\pi \cdot m \frac{n+0.5}{N}\right), \quad C_0 = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} s(z_n)$$

$$S_M(z) = \sqrt{\frac{2}{N}} \sum_{m=0}^{M-1} g_m C_m T_m(z) = \sqrt{\frac{2}{N}} \sum_{m=0}^{M-1} g_m C_m \cos(m \cdot \arccos(z))$$

$$g_m = \begin{cases} \sqrt{0.5}, & m=0 \\ 1, & m>0 \end{cases},$$

где $T_m(z)$ обозначает полином Чебышева степени m ; $z_n = \cos\left(\pi \frac{n+0.5}{N}\right)$ — значение n -го нуля полинома Чебышева степени N . Прямое преобразование по виду совпадает с прямым Дискретным Косинусным Преобразованием (ДКП) [6] с точностью до узлов, в которых вычисляются значения сигнала $s(z)$. Для ДКП эти узлы представляют собой сетку равномерных отсчётов. В случае преобразования с использованием ортогональных полиномов Чебышева, выражение $s(z_n)$ означает значения сигнала, вычисленные на неравномерной сетке, узлы которой соответствуют нулям полиномов Чебышева. Так как на практике сигнал $s(z)$ обычно задаётся отсчётами, вычисленными на равномерной сетке (например, значение пикселей изображений), в этом случае значения $s(z_n)$ можно вычислять, например, применяя методы интерполяции к значениям в узлах равномерной сетки, ближайшим к точке z_n .

При выполнении обратного преобразования восстановленный сигнал можно вычислять на произвольной сетке аргументов. Если взять такую сетку, узлы которой задаются по правилу $z_j = \cos\left(\pi \frac{j+0.5}{L}\right)$, то обратное преобразование совпадёт с обратным дискретным косинусным преобразованием.

В силу схожести предложенного преобразования с ДКП, автор назвал это преобразование GDCT (Generalized Discrete Cosine Transform — Обобщённое Дискретное Косинусное Преобразование).

Распространяя приведённые формулы на двумерный случай, можно выписать выражения для прямого и обратного преобразований в матричном виде: $C = \Phi S \Phi^T$, $R = \Psi C \Psi^T$, где матрица прямого преобразования Φ вычисляется по формуле:

$$\Phi = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{0.5} \\ \cos\left(\pi \cdot m \frac{i+0.5}{N}\right) \end{bmatrix}, \quad \begin{matrix} m=0 \dots M-1, \\ i=0 \dots N-1 \end{matrix}$$

Матрица обратного преобразования Ψ вычисляется по формуле:

$$\Psi = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{0.5} \\ \cos\left(m \cdot \arccos\left(\frac{2n}{L-1} - 1\right)\right) \end{bmatrix}, \quad \begin{matrix} m=0 \dots M-1, \\ i=0 \dots N-1 \end{matrix}$$

Таким образом, процедуру GDCT преобразования для кодирования и восстановления изображений формально можно описать последовательно следующими шагами:

- 1) Выделение блока изображения размером $N1 \times N1$
- 2) Вычисление блока $N \times N$, соответствующего значениям сигнала в нулях полинома Чебышева
- 3) Нахождение блока спектральных коэффициентов размером $M \times M$ путём вычисления прямого преобразования
- 4) Применение обратного преобразования для нахождения блока восстановленного изображения размером $L \times L$

Значения $N1$, N , M , L могут быть различны, что является преимуществом преобразования, т. к. в случае необходимости изменения размера изображения, достаточно просто выполнить обратное преобразование с другим значением L . Для изменения размера восстановленного изображения при использовании ДКП придётся прибегать к методам интерполяции.

Анализ искажений, вносимых преобразованием

При применении GDCT-преобразования к изображениям, восстановленное изображение будет отличаться от кодируемого, т. е. можно говорить об ошибке восстановления. Если считать, что в схеме кодирования не применяются процедуры округления и квантования спектральных коэффициентов, то ошибка восстановления будет складываться из двух составляющих: $E = E_{zn} + E_{int}$, где E_{zn} — это ошибка вычисления значений сигнала в узлах, соответствующих нулям полиномов Чебышева, а E_{int} означает ошибку интерполяции сигнала при восстановлении блока произвольного размера $L \times L$. Данная работа посвящена вопросу минимизации ошибки восстановления за счёт уменьшения E_{zn} .

Описание ошибки вычисления значения сигнала в нулях полиномов Чебышева

Разберём более подробно этап вычисления значений сигнала в нулях полиномов Чебышева. На рисунке 1 изображён график сигнала $s(x)$. Для вы-

полнения прямого GDCT-преобразования требуется посчитать значение $s(x)$ в узлах $\{z_0, z_1, z_2, z_3\}$, соответствующих позициям нулей полинома Чебышева третьей степени. В случае кодирования изображений нам известны значения сигнала только в узлах равномерной сетки $\{x_0, x_1, x_2, x_3\}$, что соответствует значениям пикселей блока изображения. Для вычисления $s(z_i)$ можно воспользоваться методами интерполяции. На рисунке 1 сплошные кресты обозначают истинные значения $s(z_i)$, а полыми крестами обозначены оценочные значения $\hat{s}(z_i)$, вычисленные методом линейной интерполяции по двум ближайшим точкам $s(x_{j-1})$ и $s(x_j)$. Различие $\hat{s}(z_i)$ и истинного значения $s(z_i)$ приводит к ошибке вычисления спектральных коэффициентов, что повлечёт ошибку восстановления изображения. Таким образом, неточность вычисления истинного значения $s(z_i)$ определяет ошибку E_{z_n} , описанную выше.

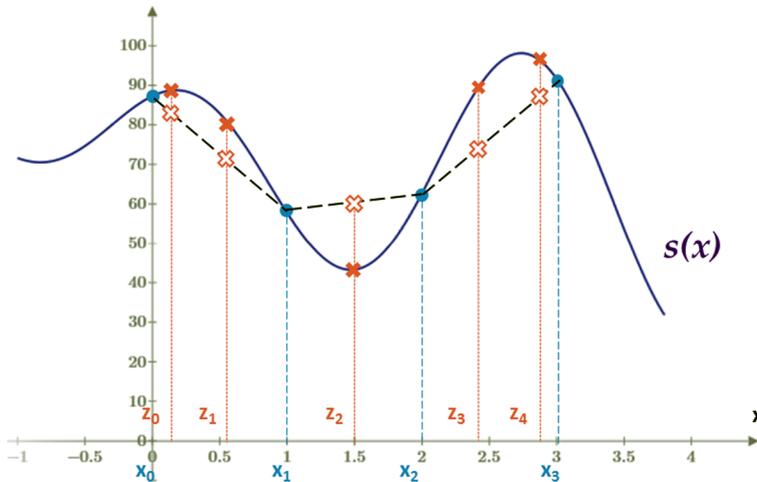


Рис. 1. Иллюстрация сигнала $s(x)$ и его значений в узлах равномерной сетки, а также истинных значений сигнала в нулях полиномов Чебышева и их оценок

Будем обозначать за N_1 размер исходного блока отсчётов сигнала на равномерной сетке, а за N размер блока отсчётов сигнала, вычисленных в нулях полинома Чебышева. В соответствии со схемой GDCT-преобразования, значения N_1 и N можно выбирать произвольным образом. Для различных пар (N_1, N) , ошибка E_{z_n} получается различной. Возникает вопрос о выборе таких параметров N_1, N , для которых E_{z_n} будет минимальна.

Способ выбора параметров N_1, N

Как было упомянуто в предыдущих разделах, уменьшить ошибку восстановления $E = E_{z_n} + E_{\text{int}}$ можно за счёт выбора таких параметров N_1, N , для

которых составляющая $E_{z_{in}}$ будет наименьшей. Для решения вопроса выбора параметров преобразования N_1, N , было выдвинуто предположение о том, что точность интерполяции при вычислении $\hat{s}(z_i)$ будет тем меньше, чем ближе узлы z_i расположены к узлам равномерной сетки. В соответствии с этим

предположением введём эвристическую метрику $\omega = \frac{1}{N} \sum_{i=0}^{N-1} \min_j (|z_i - x_j|)$, по-

казывающую среднее расстояние от узлов полиномов Чебышева z_i до ближайшего узла равномерной сетки x_j при заданных параметрах N_1, N . Тогда можно сформулировать оптимизационную задачу: для заданного значения N_1 выбрать такое значение N , при котором метрика ω будет минимальна. На Рисунке 2 изображены значения метрики ω для различных N ($N_1 = 9$). На рисунке 3 изображены графики «скорость-искажение», полученные в результате кодирования тестового изображения *lena.bmp* с параметрами $N_1 = 9$ и $N = 10, 11, 12$. Кодирование осуществлялось по схеме, аналогичной кодированию по стандарту JPEG [1, 2]; в качестве энтропийного кодирования использовался арифметический кодек.

Из приведённых графиков видно, что при одинаковой степени сжатия кривая «скорость-искажение» для $N = 11$ соответствует лучшему качеству, чем при выборе $N = 10$ или $N = 12$. Из графика на рис. 3 видно, что значение $N = 11$ соответствует меньшему значению метрики ω , что подтверждает предложенный метод выбора N для заданного N_1 . Аналогичные графики были построены для различных значений N_1 и так же не опровергли предложенный метод.



Рис. 2. Значение метрики ω для различных N при $N_1 = 9$

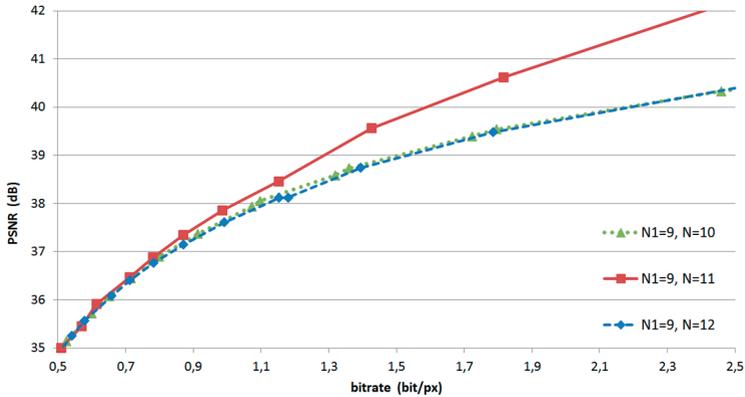


Рис. 3. Графики «скорость-искажение» при кодировании тестового изображения *lena.bmp* для различных соотношений N_1/N

Заключение

В данной работе был проведён анализ ошибок, вносимых GDCT-преобразованием при кодировании изображений. Был предложен метод выбора параметров преобразования, позволяющий уменьшить эту ошибку. Были построены графики по результатам кодирования тестового изображения с использованием параметров GDCT-преобразования, полученных предложенным методом. Графики показали, что выбор параметров предложенным методом позволяет получить выигрыш порядка 0,5 дБ на низких битовых скоростях и порядка 4 дБ на высоких битовых скоростях.

Литература

1. CCITT Recommendation T.81. Information technology. Digital compression and coding of continuous-tone still images. Requirements and guidelines. 1992
2. Wallace G.K. The JPEG still picture compression standard // Commun. ACM. 1991.— April. Vol. 34. Pp. 30–44
3. Радченко Ю.С. Алгоритм сжатия изображений на основе полиномиальных преобразований (алгоритм GDCT) // Цифровая обработка сигналов. 2002. С. 2–6
4. Радченко Ю.С., Радченко Т.А., Булыгин А.В. Сравнение алгоритмов сжатия изображений на основе Чебышевского (GDCT) и дискретного косинусного (DCT) ортогональных преобразований // Цифровая обработка сигналов и ее применение. 2006
5. Суетин П.К. «Классические ортогональные многочлены.» – 3-е изд., М.: ФИЗМАТЛИТ, 2005
6. N. Ahmed, T. Natarajan and K.R. Rao, «Discrete Cosine Transform» // IEEE Transactions on Computers, Jan. 1974, pp. 90–93

ОБ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ АЛГОРИТМА SURF В ЗАДАЧЕ ИДЕНТИФИКАЦИИ ЛИЦ

Новиков Д.В.

студент 5-го курса кафедры защиты информации ГУАП, dn@vu.spb.ru

Аннотация: В данной работе проанализирована эффективность применения алгоритма SURF в задаче идентификации лиц. Рассмотрены наиболее популярные алгоритмы идентификации лиц. Предложен собственный метод идентификации лиц на основе алгоритма SURF. Проведен сравнительный анализ предложенного метода и существующих решений.

Введение

Технологии автоматического распознавания лица используются в таких системах как: биометрическая идентификация, зрение роботов, видеоконференции. Задача, которая стоит перед алгоритмами таких систем — это идентификация человека по лицу. Алгоритмы идентификации в этих системах должны показывать уверенную работу при различных условиях освещенности, иметь низкие показатели ложной идентификации. Алгоритм должен быть подготовлен к тому, что лицо человека в разные моменты времени может иметь другой масштаб и быть повернуто.

Обзор существующих решений задачи идентификации лиц

В этом разделе рассматриваются основные принципы работы наиболее популярных алгоритмов идентификации лиц.

1. Алгоритм EigenFaces [2]

Работа алгоритма основана на методе главных компонент. Предположим, что имеется база данных лиц, где изображения имеют размер $N \times N$ пикселей. Каждое изображение из базы данных представляют точкой в пространстве размерностью $N \times N$. Основная идея алгоритма состоит в том, чтобы найти такой базис меньшей размерности, после проекции в который максимально сохраняется информация по осям с большой дисперсией и теряется информация по осям с маленькой дисперсией. Это нужно для того, чтобы оставить только ту информацию, которая бы характеризовала различия лиц и удалить ненужную информацию, которая может помешать правильно идентифицировать человека.

Процедура идентификации выполняется в новом базисе с использованием Евклидовой метрики.

Основными недостатками алгоритма EigenFaces является отсутствие устойчивости к изменению условий освещенности и отсутствие инвариантности к аффинным преобразованиям.

2. Алгоритм FisherFaces [3]

Алгоритм предполагает наличие множества фотографий при разных условиях освещенности у каждой персоны в базе данных. В алгоритме, как и в EigenFaces, предполагается поиск базиса, но такого, который позволил бы максимизировать дисперсию между множествами изображений лиц и одновременно минимизировать дисперсию внутри каждого множества.

За счет множества фотографий каждой персоны алгоритм получается устойчивым к изменениям условий освещенности, но сохраняет недостаток алгоритма EigenFaces в отсутствии инвариантности к аффинным преобразованиям.

Алгоритм SURF (Speeded-Up Robust Features) для выделения и описания ключевых особенностей изображения [1]

В этом разделе рассматриваются ключевые моменты алгоритма SURF и анализируются его основные преимущества и недостатки.

1. Поиск ключевых точек изображения

Ключевая точка изображения — точка, которая имеет некие признаки, существенно отличающие ее от основной массы точек (резкие перепады освещенности, углы и т. д.). Ключевые точки изображения ищутся путем анализа определителя матрицы Гёссе для каждого пикселя изображения:

$$\det(H) = \frac{\partial^2 f}{\partial x^2} * \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \quad (1)$$

Для того, чтобы алгоритм был устойчив к масштабированию изображения, используют разные размеры области, по которой берутся вторые производные.

Если значение определителя превысило специально установленный порог, то считается, что найдена ключевая точка. Такая операция выполняется для каждого пикселя изображения.

2. Нахождение ориентации ключевой точки изображения

Для каждой найденной ключевой точки в ее окрестности ищется направление вектора максимального перепада функции интенсивности изо-

бражения вблизи данного пикселя с использованием фильтров Хаара. Стоит отметить, что размеры фильтров, а также размеры анализируемой окрестности выбираются в соответствии с размером области взятия вторых производных.

3. Построение дескриптора ключевой точки изображения

Вокруг ключевой точки описывается прямоугольная область. Эта область разбивается на 16-ти квадрантов одинаковых размеров. Прямоугольная область затем поворачивается в соответствии с ориентацией ключевой точки. На следующем шаге считаются оценки для каждого из 16-ти квадрантов области с помощью фильтров Хаара:

$$\sum dx, \sum |dx|, \sum dy, \sum |dy|, \quad (2)$$

Размеры прямоугольной области, а также размеры фильтров Хаара зависят от размера области взятия вторых производных.

В результате получается вектор из 64-х чисел. К описанию точки также добавляют след матрицы Гёссе. Вектор и след матрицы вместе образуют дескриптор ключевой точки. На данном этапе алгоритм SURF заканчивается.

4. Анализ преимуществ и недостатков алгоритма

За счет использования вторых производных алгоритм SURF становится инвариантен к перепадам яркости. Благодаря использованию разных размеров области взятия вторых производных и вычислению ориентации ключевых точек алгоритм становится инвариантен к изменению масштаба и повороту лица человека в плоскости изображения [1].

Схема идентификации на основе алгоритма SURF (Speeded-Up Robust Features)

В этом разделе описывается способ идентификации лиц на основе алгоритма SURF. Схема идентификации представлена на Рисунке 3.

Входом схемы является изображение лица. Выходом схемы является индекс человека в базе данных, который был обнаружен на этом изображении. На выходе модуля SURF получается набор из ключевых точек и их дескрипторов. Из базы данных поступает множество ключевых точек и дескрипторов всех содержащихся там изображений. Модуль сравнения дескрипторов ищет пару для каждого дескриптора входного изображения среди всех дескрипторов базы данных, используя для этого вспомогательные методы:

1. **Метод сетки.** Считается, что ключевые точки соответствующих изображений должны иметь схожее пространственное положение.

2. **Анализ масштаба фильтров.** Ключевые точки должны быть получены при близких размерах фильтров вторых производных.
3. **Анализ ориентации ключевых точек.** Считается, что отклонение ориентации ключевых точек не должно превысить некоторый порог.
4. **Евклидова метрика.** Вычисляется для каждой пары дескрипторов.
5. **Гистограммный метод.** Позволяет разрешать конфликтные ситуации, когда более одного дескриптора из исходного изображения выбрали себе в пару один и тот же дескриптор из базы данных. Выбирается пара с наиболее похожими гистограммами окрестностей ключевых точек.

Вычисляются характеристики совокупности пар дескрипторов ключевых точек каждого изображения базы данных. Далее, на основе полученных результатов принимается решение о том, к какому изображению в базе данных ближе всего входное изображение и выдается индекс такого изображения.

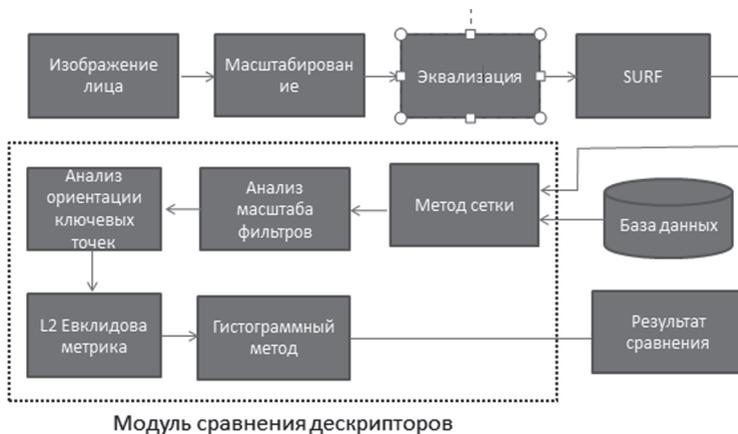


Рис. 3. Схема идентификации на основе алгоритма SURF

Сравнение алгоритмов идентификации лиц

В этом разделе анализируется эффективность использования предложенной схемы идентификации лиц на основе алгоритма SURF (в Таблицах 1 и 2 схема обозначена как S_{our}) путем сравнения его показателей с аналогичными показателями для алгоритмов S_{orig} , EigenFaces и FisherFaces. Под S_{orig} подразумевается схема идентификации, использующая оригинальный алгоритм SURF для получения ключевых точек и их дескрипторов и Евклидову метрику для сопоставления этих дескрипторов. Первым показателем является вероятность события, при котором человек с видеоролика будет верно идентифицирован. Вторым показателем является вероятность события, при

котором человек с видеоролика не будет верно идентифицирован. Стоит отметить, что в сумме оба показателя не дают единичную вероятность.

Для сравнения использовались два тестовых набора: случай естественного и искусственного освещения. База данных в количестве девяти изображений лиц делалась при естественном освещении. В качестве тестового набора были использованы 9 видеозаписей, снятых при естественном освещении и 3 видеозаписи, снятых при искусственном освещении. Каждая видеозапись содержит одну персону. Все алгоритмы идентификации принимали на вход одинаковый набор данных. Результаты работы алгоритмов на тестовых множествах с естественным и искусственным освещением представлены в Таблицах 1 и 2 соответственно.

№	Вероятность верной идентификации				Вероятность ложной идентификации			
	S_our	S_orig	EigenFaces	FisherFaces	S_our	S_orig	EigenFaces	FisherFaces
1	0,94	0,78	0,36	0,53	0,03	0,08	0,33	0,36
2	0,89	0,67	0,00	0,00	0,04	0,13	0,56	0,82
3	1,00	0,87	0,69	0,65	0,00	0,07	0,31	0,18
4	0,97	0,97	0,77	0,55	0,03	0,02	0,14	0,30
5	1,00	0,92	0,94	0,78	0,00	0,08	0,06	0,14
6	1,00	1,00	0,97	0,96	0,00	0,00	0,00	0,01
7	0,71	0,10	0,24	0,71	0,14	0,24	0,38	0,24
8	1,00	1,00	0,97	0,56	0,00	0,00	0,03	0,44
9	1,00	0,21	0,00	0,00	0,00	0,43	0,93	0,93

Таблица 1. Результаты алгоритмов идентификации лиц на тестовом множестве с естественным освещением

№	Вероятность верной идентификации				Вероятность ложной идентификации			
	S_our	S_orig	EigenFaces	FisherFaces	S_our	S_orig	EigenFaces	FisherFaces
1	0,34	0,34	0,02	0,15	0,34	0,24	0,55	0,45
2	0,88	0,78	0,00	0,00	0,04	0,10	0,92	0,94
3	0,84	0,47	0,00	0,00	0,11	0,21	1,00	1,00

Таблица 2. Результаты алгоритмов идентификации лиц на тестовом множестве с искусственным освещением

Таблицы показывают, что предложенный способ идентификации на основе метода SURF (S_our) показал хорошие результаты. Только на видеоролике № 1 при искусственном освещении не удалось верно идентифицировать человека.

Заключение

В статье были рассмотрены наиболее популярные алгоритмы идентификации лиц. Был рассмотрен алгоритм SURF. Предложена процедура иденти-

фикации лиц на основе алгоритма SURF. Продемонстрирована эффективность работы полученной схемы. Предложенная схема идентификации лиц обладает лучшими показателями вероятностей верной и ложной идентификации среди рассмотренных алгоритмов на двух тестовых множествах.

Литература

1. *Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool* Speeded-Up Robust Features (SURF), 2008.
 2. *Matthew Turk and Alex Pentland* Eigenfaces for Recognition, 1991.
 3. *Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman* Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, 1997.
-

ОЦЕНКА МЕТОДА МАСКИРОВАНИЯ В ИЗУАЛЬНЫХ ИСКАЖЕНИЙ НА СТОРОНЕ ДЕКОДЕРА ДЛЯ УМЕНЬШЕНИЯ ВЛИЯНИЯ ПОТЕРЬ ПРИ ПЕРЕДАЧЕ ВИДЕОДАНЫХ ПО СЕТИ

Веселов А.И.

ассистент кафедры инфокоммуникационных систем ГУАП, felix@vu.spb.ru

Ястребов В.А.

*студент кафедры безопасности информационных систем ГУАП,
victor.yastrebov@vu.spb.ru*

Аннотация: В статье рассматривается задача обработки визуальных искажений, возникающих при потере пакетов видеоданных в сети. Предлагается метод борьбы с такими искажениями, основанный на применении алгоритмов пространственной интерполяции потерянных областей с использованием методов восстановления изображений. Приводятся результаты оценки предложенного метода на простой модели системы передачи видеоданных, демонстрируется целесообразность применения предложенного подхода.

Введение

В настоящее время существует большое количество приложений, требующих обработку и передачу видеоданных в режиме реального времени. Примерами таких приложений являются видеоконференции, видео звонки, просмотр потокового видео и т. д. Типовая схема системы передачи видеоданных для обозначенных выше сценариев приведена на рисунке 1А. Исходное видео, являющееся последовательностью кадров, подвергается сжатию (кодированию), сжатые данные разбиваются на пакеты и передаются декодеру, который по мере получения пакетов восстанавливает кадры видеопоследовательности. В процессе передачи по сети могут возникать ошибки, которые отражаются в искажении или потере пакетов, что может приводить к существенному падению качества восстановленного видео (рисунок 3Б, рисунок 3Г). В данной статье предлагается метод маскирования визуальных искажений, позволяющий уменьшить влияние такого рода ошибок на качество восстановленных видеоданных. Производится оценка предложенного метода в рамках простой модели передачи данных.

Описание рассматриваемой модели обработки и передачи видеоданных по сети

Введем в рассмотрение модель системы обработки и передачи видеоданных (рисунок 1А), построенную на основе следующих допущений.

В качестве кодека видеоданных используется эталонная реализация кодека стандарта H.264/AVC [1] — JM 16.1 [2]. Кодер (передатчик) считается фиксированным, т.е. никакие модификации в него вносить нельзя. Сжатый поток формируется следующим образом. Первый кадр видеопоследовательности кодируется в режиме Intra (I кадр), все остальные кадры обрабатываются как Inter кадры (P кадры). Одному кадру ставится в соответствие один слайс (slice). Стандарт H.264/AVC предоставляет уровень абстракции между сжатыми данными и средой передачи (NAL, Network Abstraction Layer). На этом уровне сжатые данные упаковываются в блоки, называемые NALU (NAL Units). В рассматриваемой модели предполагается, что в один NALU попадают сжатые данные только от одного макроблока, и в одном пакете передается только один NALU. Вводится модель ошибок, возникающих при передаче NALU пакетов по сети, основанная на следующих допущениях:

- Любые искажения в пакете будут обнаружены;
- Обнаруженные искажения трактуются как полная потеря пакета;
- При передаче I кадра возникают ошибки, которые приводят к потере только подряд идущих пакетов (соответствующих соседним в кадре макроблокам);
- При передаче P кадров ошибки не возникают.

Декодер получает из сети пакеты со сжатыми данными и осуществляет восстановление видеопоследовательности. Потеря пакета приводит к появлению черного квадрата на восстановленном кадре по координатам соответствующим координатам потерянного макроблока. При этом считается, что ошибка не распространяется по I кадру. На P кадрах возникают искажения при декодировании макроблоков, использующих явно или неявно потерянный макроблок в качестве базового.

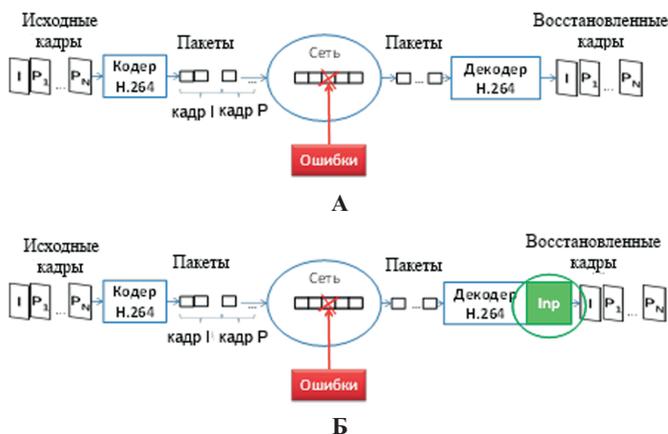


Рис. 1. А) Модель кодека и модель ошибок.
Б) Предложенная модификация декодера

Перед рассмотрением предложенного метода следует отметить, что стандарт H.264 AVC предлагает ряд способов борьбы с ошибками передачи:

- Гибкий выбор порядка следования макроблоков;
- Произвольный порядок выдачи слайсов;
- Генерация избыточных слайсов.

Первые два подхода позволяют уменьшить визуальное влияние ошибки, «распространив» её по кадру. Третий метод позволяет в некотором роде бороться с проблемой потери пакетов за счет увеличения объема передаваемых данных. Тем не менее, использование третьего метода не гарантирует, что все пакеты, необходимые для безошибочного декодирования кадра, успешно дойдут до декодера, т.к. возможна ситуация, что произойдет такая ошибка, которая исказит и избыточные слайсы. В связи с этим адекватная обработка потерянных пакетов декодером является актуальной задачей.

Описание предложенного метода

Для решения поставленной задачи предлагается использовать подходы, основанные на методах восстановления изображений (image inpainting [3]). Эти подходы позволяют произвести интерполяцию поврежденной или недоступной части изображения. Предлагается модифицировать декодер видеоданных, добавив в него блок восстановления базового кадра (рисунок 1Б). Этот блок используется для детектирования потерянных макроблоков и их последующей интерполяции. Обобщенная схема предлагаемого алгоритма восстановления приведена на рисунке 2.



Рис. 2. Схема работы алгоритма восстановления

Предложенный алгоритм восстановления базируется на статье [4]. Основное отличие состоит в том, что на каждом шаге изменению (интерполяции) подвергается только один пиксель, а не несколько пикселей сразу.

Введем понятия, которые будут использоваться для описания алгоритма. Исходная область — область на изображении, которая остается неизменной в ходе работы алгоритма. Целевая область — область на изображении, которую необходимо изменить (интерполировать).

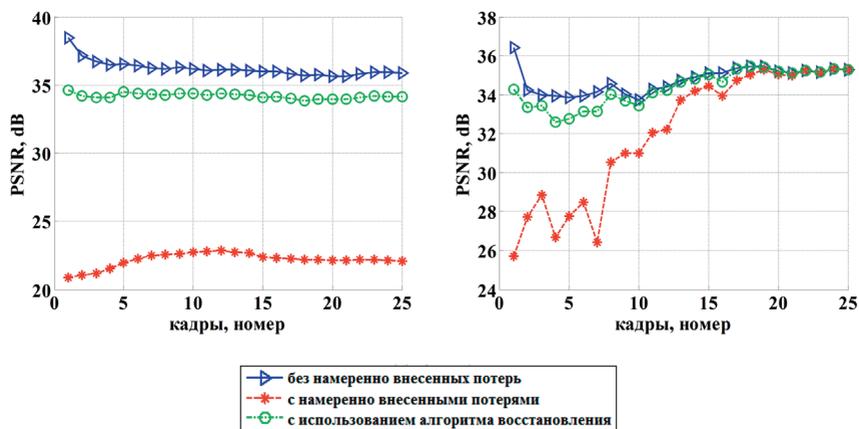
В ходе работы алгоритма осуществляется замена каждого пикселя из целевой области наиболее похожим пикселем из исходной области изображения. Степень похожести определяется путем анализа апертур (окружения) целевого и исходного пикселей с использованием критерия SAD (Sum of Absolute Differences, сумма абсолютных разностей). В случае если несколько исходных пикселей имеют равный минимальный SAD, то апертюра каждого из них анализируется заново, уже с использованием критерия SSD (Sum of Squared Differences, сумма квадратов разностей). Более подробное описание алгоритма приведено в [5].

Следует отметить, что описанная ранее модель системы передачи данных обладает рядом существенных упрощений, не соответствующих реальности. Тем не менее, целесообразность применения метода маскирования искажений и потенциальный выигрыш от предложенного метода можно оценить и на такой простой модели. Также, следует сказать, что существуют работы, например [6], в которых рассматриваются схожие подходы, основанные на пространственной интерполяции потерянных областей. Отличие данной статьи заключается в том, что в качестве метода интерполяции предлагается использовать подход, показавший хорошие результаты при решении схожей задачи — задачи восстановления поврежденных изображений. Сравнение с альтернативными методами интерполяции является предметом дальнейших исследований.

Оценка предложенного метода

Оценка предложенного метода была проведена на наборе стандартных тестовых видеопоследовательностей [7]. Результаты проведенных экспериментов для двух последовательностей (с медленным и быстрым движением) отображены на рис. 3.

На протестированных последовательностях удалось получить существенный выигрыш по критерию PSNR (до 13 дБ). При этом на малодинамичной последовательности Foreman (Рис. 3А) величина выигрыша слабо изменяется с ростом номера кадра. Это объясняется тем, что утерянный на базовом кадре фрагмент используется при обработке большого количества кадров, т. е. его влияние сильно распространено во времени. С другой стороны для динамичной последовательности (Рис. 3В) после 20 кадра графики PSNR начинают сходиться. Это объясняется тем, что со временем из-за высокой динамики движения, влияние утерянного макроблока на предсказанные кадры сильно падает. Тем не менее, на этой последовательности до 20 кадра также наблюдается существенный прирост PSNR.



А

В



Б



Г

Рис. 3. А) Покадровый график PSNR для последовательности Foreman CIF.
 Б) Результат декодирования Intra кадра с внесенными потерями (Foreman CIF).
 В) Покадровый график PSNR для последовательности Football CIF.
 Г) Результат декодирования Intra кадра с внесенными потерями (Football CIF)

Заключение

В данной работе была рассмотрена задача уменьшения влияния потерь при передаче видеоданных по сети. Для решения этой задачи было предложено использовать метод, основанный на принципах восстановления изображений. Для оценки этого метода была введена в рассмотрение простая модель обработки и передачи видеоданных по сети. В рамках этой модели удалось продемонстрировать выигрыш до 13 дБ, по сравнению с декодированием без предложенной модификации. В дальнейшем планируется про-

вести сравнение с альтернативными методами борьбы с потерями, например [6], рассмотрев при этом более сложные модели ошибок. Кроме того, необходимо проанализировать способы снижения сложности используемого алгоритма восстановления, т. к. текущая реализация не подходит для применения в режиме реального времени.

Литература

1. H.264/MPEG-4 part 10 AVC ISO/IEC 14496-10, Coding of Audio-Visual Objects — Part 10. Advanced Video Coding, 2003
 2. JM H.264/AVC codec: <http://vqegstl.ugent.be/?q=node/14>
 3. *M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester*, Image inpainting, Computer Graphics, SIGGRAPH 2000, July 2000
 4. *A. Criminisi, P. Perez, K. Toyama*. Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE Transactions on Image Processing, vol.13, No9, Sep.2004
 5. *В.А.Ястребов, В.П.Реумт* «Метод пространственной интерполяции изображений», 65 международная студенческая конференция ГУАП: сб.докл.: В 2 Ч.Ч.I Технические науки // СПбГУАП. СПб., 2012, 516 с. ил.
 6. *H. Gharavi, S. Gao*, Spatial Interpolation Algorithm For Error Concealment, IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.
 7. Xiph.org Video Test Media [derf's collection], <http://media.xiph.org/video/derf/>
-

МЕТОДИКА РАСЧЕТА ПРЕДЕЛЬНЫХ ХАРАКТЕРИСТИК СИСТЕМЫ СВЯЗИ, РАБОТАЮЩЕЙ НА ПРИНЦИПАХ НЕЛИНЕЙНОГО ОТРАЖЕНИЯ

Гранкин М.А.

асп. кафедры инфокоммуникационных систем ГУАП, m.a.grankin@gmail.com

Бакин Е.А.

ассистент кафедры моделирования вычислительных и электронных систем ГУАП, jenyb@vu.spb.ru

Аннотация: В работе рассмотрена перспективная система сбора низкоскоростной телеметрической информации с маломощных сенсоров, основанная на принципах нелинейного отражения. В таких системах для передачи информации используется энергия внешнего электромагнитного поля. Предложена методика расчета таких параметров системы, как потенциальное отношение сигнал/шум, предельные скоростные характеристики и т. д. В работе приводится пример использования данной методики для типового расчета характеристик сенсорной сети.

Введение

В настоящее время актуальной задачей является разработка новых методов сбора данных с маломощных сенсорных устройств [1]. Предметом исследований в этой области является, как правило, минимизация аппаратной сложности сенсоров, их стоимости и энергопотребления. Обычно к таким системам не накладывается существенных требований по скорости передачи данных.

В данной работе рассматривается новый перспективный способ сбора низкоскоростной телеметрической информации, основанный на принципах нелинейного отражения первичного сигнала. В системах использующих данный принцип каждый сенсор оснащается так называемым нелинейным отражателем. Нелинейный отражатель представляет собой антенну, нагруженную на нелинейную нагрузку [2]. При поступлении в антенну внешнего сигнала во входной цепи появляются кратные комбинационные гармоники, которые переизлучаются в эфир через ту же антенну. Отключение нелинейной нагрузки приводит к пропаданию комбинационных гармоник из эфира. Таким образом, путем включения/выключения нелинейной нагрузки возможна передача данных при помощи, например, сигналов с пассивной паузой (ON/OFF keying) или позиционно импульсной модуляции (pulse-position modulation, PPM) [3]. Как правило, используется только вторая гармоника. Пример сенсорной сети, сбор информации с которой основан на данном принципе, приведен на рисунке 1.

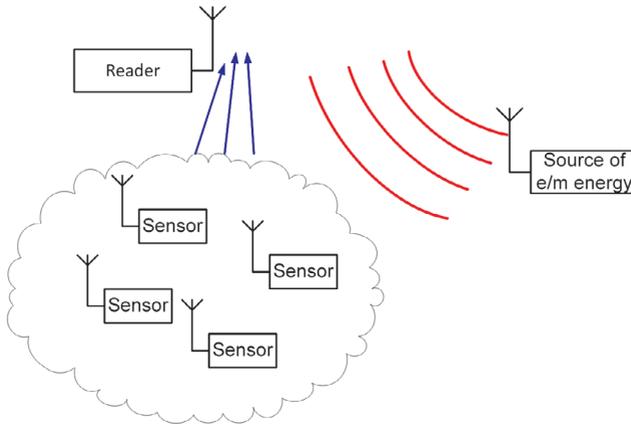


Рис. 1. Пример сенсорной сети

В предыдущих работах анализ подобных систем осуществлялся в основном экспериментально [2]. Представляет, однако, интерес методика теоретической оценки характеристик системы. Разработке такой методики посвящена данная работа.

Согласно основному уравнению дальности энергии, принятая сенсором равна [4], [5]:

$$P_{S, TX} = P_{e/m} \left(\frac{G_{e/m}}{4\pi r_0^2} \right) \left(\frac{G_S \lambda_1^2}{4\pi} \right) \left(\frac{r_0}{r_1} \right)^\beta \quad (1)$$

где: $P_{e/m}$ — максимальная излучаемая мощность источника электромагнитной энергии (Вт), $G_{e/m}$ — КНД (коэффициент направленного действия) антенны точки доступа (разы), G_S — КНД (коэффициент направленного действия) антенны сенсора (разы), r_0 — референсное расстояние, λ_1 — длина волны первичного сигнала (метры), r_1 — расстояние между источником электромагнитной энергии точкой доступа и сенсором в метрах, β — коэффициент затухания в среде, для вакуума равен 2, но для зданий или помещений с препятствиями β равна между 4 и 5 [6].

С учетом потерь в датчике, энергия, излученная сенсором равна:

$$P_{S, RX} = L \times P_{S, TX} = LP_{e/m} \left(\frac{G_{e/m}}{4\pi r_0^2} \right) \left(\frac{G_S \lambda_1^2}{4\pi} \right) \left(\frac{r_0}{r_1} \right)^\beta$$

где L — потери в сенсоре (разы).

Применяя второй раз основное уравнение дальности (1), находим мощность сигнала, поступившего в считыватель:

$$P_{R,TX} = P_{S,RX} \left(\frac{G_S}{4\pi r_0^2} \right) \left(\frac{G_R \lambda_2^2}{4\pi} \right) \left(\frac{r_0}{r_2} \right)^\beta = P_{e/m} \frac{L G_{e/m} G_R (G_S \lambda_1 \lambda_2)^2}{(4\pi r_0)^4} \left(\frac{r_0^2}{r_1 r_2} \right)^\beta$$

где: G_R — КНД (коэффициент направленного действия) антенны считывателя (разы), λ_2 — длина волны отраженного сигнала (метры).

С учетом потерь в нелинейном отражателе, излучаемая мощность сенсора равна:

$$P_{S,RX} = L \times P_{S,TX} = L P_{e/m} \left(\frac{G_{e/m}}{4\pi r_0^2} \right) \left(\frac{G_S \lambda_1^2}{4\pi} \right) \left(\frac{r_0}{r_1} \right)^\beta$$

При длительности бита, равной $\tau_b = 1/R$ энергия, приходящаяся на бит, равна:

$$E_{R,TX} = \frac{1}{2} \frac{P_{e/m} L G_{e/m} G_R (G_S \lambda_1 \lambda_2)^2}{R (4\pi r_0)^4} \left(\frac{r_0^2}{r_1 r_2} \right)^\beta$$

Здесь коэффициент $1/2$ говорит о том, что используется система с пассивной паузой. В этом типе модуляции наличие сигнала символизирует двоичную единицу, а отсутствие сигнала — двоичный ноль.

По формуле Больцмана спектральная мощность аддитивного шума в приемнике считывателя равна:

$$N_0 = k T K_{ш}$$

где k — постоянная Больцмана (1.38×10^{-23}), $K_{ш}$ — коэффициент шума приемника считывателя (разы), T — температура (Кельвены).

Тогда отношение сигнал/шум в приемнике (как отношение энергии на бит к спектральной плотности мощности шума) равно:

$$q_s = \frac{P_{e/m} L G_{e/m} G_R (G_S \lambda_1 \lambda_2)^2}{2k T K_n R (4\pi r_0)^4} \left(\frac{r_0^2}{r_1 r_2} \right)^\beta \quad (2)$$

Для численного примера возьмем исходный сигнал на частоте 2.4 ГГц так как это нелецензируемый бесплатный диапазон. Параметры для анализа будут следующие:

- $r_0 = 75$ см (т.е. 6 длин волны сигнала с несущей 2.4 ГГц);
- $r_1 = 2$ м;
- $G_{e/m} = 2.15$ dB (соответствует полуволновой дипольной антенне);
- $G_S = 2.15$ dB (соответствует полуволновой дипольной антенне);
- $G_R = 2.15$ dB (соответствует полуволновой дипольной антенне);
- $L = -60$ dB (данный замер был сделан в [2]);
- $K_n = 10$ dB (соответствует обычному приемнику);

- $P_{e/m} = 100$ мВт (стандарт OFDM передатчик);
- $f_1 = 2.4$ ГГц (частота несущего сигнала);
- $f_2 = 4.8$ ГГц (соответствует второй гармонике сигнала с несущей частотой 2.4 GHz сигнала прошедшего через нелинейный элемент).

Следует отметить, что при малых расстояниях между сенсором и считывателем, основной луч отклика канала является доминирующим. Исходя из этого были взяты $\beta = 2$ и $\beta = 2.5$.

Для системы с описанными параметрами зависимость отношения сигнал/шум от скорости передачи данных и расстояния между источником электромагнитной энергии (стандартный OFDM передатчик) и сенсором представлено на рис. 2–3.

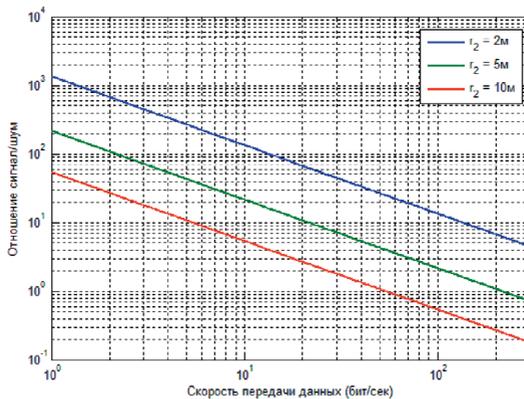


Рис. 2. Зависимость отношения сигнал/шум от скорости передачи данных при $\beta = 2$

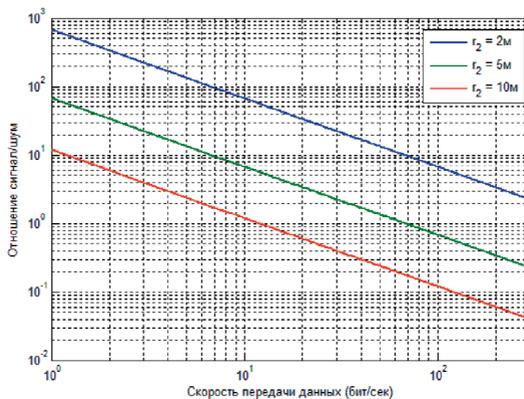


Рис. 3. Зависимость отношения сигнал/шум от скорости передачи данных при $\beta = 2.5$

Зная отношение сигнал/шум в приемнике считывателя и, учитывая, что используемая модуляционная схема является схемой с ортогональными сигналами и пассивной паузой можно оценить вероятность битовой ошибки в случае некогерентного приема:

$$P_e = \frac{1}{2} \exp\left(-\frac{q_s}{2}\right) \quad (3)$$

Согласно уравнениям (2), (3), достижимая скорость передачи данных будет составлять:

$$R = -\frac{P_{e/m} L G_{e/m} G_R (G_S \lambda_1 \lambda_2)^2}{2kTK_n (4\pi r_0)^4 2 \ln(2P_e)} \left(\frac{r_0^2}{r_1 r_2}\right)^\beta$$

Для системы с описанными параметрами зависимость вероятности ошибки на бит (BER) от используемой скорости передачи данных и расстояния между источником электромагнитной энергии (стандартный OFDM передатчик) и сенсором была представлена на рис. 4–5.

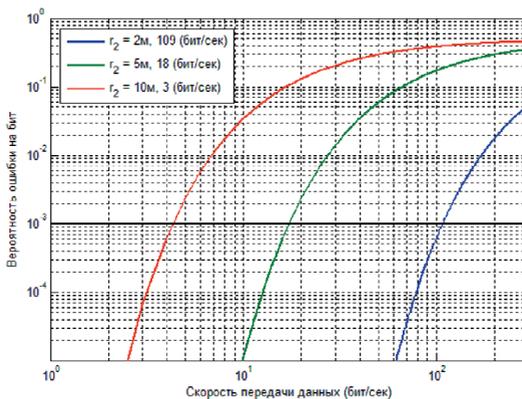


Рис. 4. Зависимость вероятности ошибки на бит от скорости передачи данных при $\beta=2$

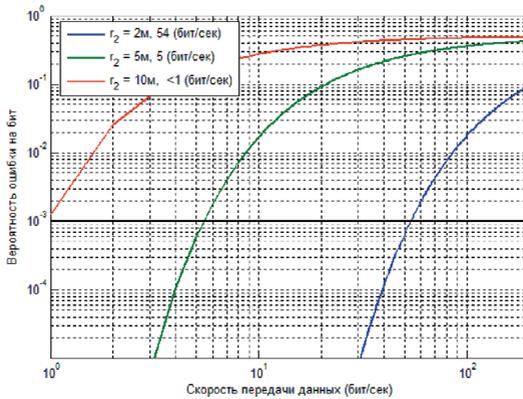


Рис. 5. Зависимость вероятности ошибки на бит от скорости передачи данных при $\beta = 2.5$

Заключение

Как видно, при требовании на вероятность ошибок по битам в 10^{-3} (вероятность ошибки на бит при которой возможно использование маломощных корректирующих кодов), может быть достигнута скорость передачи данных до нескольких десятков бит в секунду. Эта скорость передачи данных достаточна для сбора низкоскоростной телеметрической информации.

Литература

1. *Yoichi Yano*. Take the expressway to go greener. ISSCC 2012, 2012.
2. *I.S. Shishalov V.A. Yashnov A.L. Umnov, D.A. Golavachev*. The system collecting the information from sensor network nodes built on the basis of nonlinear scatterers. 5th International Conference on Antenna Theory and Techniques, 24–27 May, Kyiv, Ukraine, 2005.
3. *Kamil Sh. Zigangirov*. Theory Of Code Division Multiple Access Communication. IEEE Press, 2004.
4. *Sony Y. Tjandra Seth R. Parko Ajay Gupta Boris Kulic Arthur A. Julius Vinko Erceg, Larry J. Greenstein and Renee Bianchi*. An empirically based path loss model for wireless channels in suburban environments. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 17, NO. 7, 1999.
5. Channel models for fixed wireless applications. IEEE 802.16 Broadband Wireless Access Working Group, 2001.
6. *Atreyi Bose and Chuan Heng Foh*. A practical path loss model for indoor wifi positioning enhancement. School of Computer Engineering Nanyang Technological University, Singapore, 2007.

Распараллеливание в OPEN MP и сплайновые аппроксимации



**Бурова
Ирина Герасимовна**

д.ф.-м.н.

профессор кафедры параллельных алгоритмов СПбГУ

РАСПАРАЛЛЕЛИВАНИЕ В ОРЕНМР СРЕДНЕКВАДРАТИЧЕСКИХ ПРИБЛИЖЕНИЙ НЕПОЛИНОМИАЛЬНЫМИ СПЛАЙНАМИ МИНИМАЛЬНОГО ДЕФЕКТА

Винник М.П.

*541 группа Математико-механический факультет СПбГУ,
mvinnik92@gmail.com*

Бурова И.Г.

*профессор каф. Вычислительной математики
Математико-механический факультет СПбГУ, burovaig@mail.ru*

Аннотация: В данной работе рассматривается применение не-полиномиальных сплайнов минимального дефекта к задаче построения среднеквадратического приближения. Исследуются различные варианты оптимизации решения методом релаксации системы линейных алгебраических уравнений, возникающей в процессе построения среднеквадратического приближения. Проведен сравнительный анализ различных вариантов распараллеливания программ.

Введение

Практически всегда экспериментально полученные данные содержат различные погрешности и помехи. При решении такого рода задач появляется необходимость корректировки данных. Базисные функции второй степени минимального дефекта приводят к решению системы линейных алгебраических уравнений с ленточной матрицей с шириной ленты равной пяти и дают непрерывно дифференцируемое решение задачи среднеквадратического приближения.

1. Построение базисных сплайнов

1.1. О построении полиномиальных B -сплайнов второй степени

Пусть m — натуральное число, на промежутке $[a, b]$ задана равномерная сетка узлов

$$a = x_0 < \dots < x_i < \dots < x_m = b,$$

с шагом

$$h = (b - a) / m,$$

так что

$$x_i = a + ih, \quad i = 0, 1, \dots, m = b.$$

Базисные сплайны, $\omega_j(x)$, удовлетворяющие условиям:

$$\text{supp } \omega_j = [x_{j-1}, x_{j+2}], \quad \omega_j \in C^1[a, b],$$

будем строить решая систему уравнений с параметрами $c_{01}, c_{10}, c_{02}, c_{20}, c_{22}$ относительно $\omega_j(x)$. На промежутке $[x_j, x_{j+1}]$ система имеет вид:

$$\begin{cases} \omega_{j-1}(x) + \omega_j(x) + \omega_{j+1}(x) = 1, \\ x_{j-1}\omega_{j-1}(x) + x_j\omega_j(x) + x_{j+1}\omega_{j+1}(x) = c_{10}x + c_{01}, \\ x_{j-1}^2\omega_{j-1}(x) + x_j^2\omega_j(x) + x_{j+1}^2\omega_{j+1}(x) = c_{20}x + c_{11}x^2 + c_{02}. \end{cases}$$

Аналогичные системы уравнений имеем на соседних промежутках.

Параметры $c_{01}, c_{10}, c_{02}, c_{20}, c_{22}$ подбираем так, чтобы $\omega_j \in C^1[a, b]$.

При значениях параметров

$$c_{02} = -h, \quad c_{01} = \frac{h}{2}, \quad c_{10} = c_{20} = 1, \quad c_{11} = h, \quad c_{02} = \frac{h^2}{2}$$

находим:

$$\begin{aligned} \omega_j(x) &= -\frac{(x-jh)^2}{h^2} + \frac{1}{2h}(h(2j-1)+2x)+1-2j, \\ \omega_{j-1}(x) &= \frac{(x-jh-h)^2}{2h^2}, \quad \omega_{j+1}(x) = \frac{(x-jh)^2}{2h^2}. \end{aligned}$$

Нетрудно получить формулу базисного сплайна:

$$\omega_j(x) = \begin{cases} \frac{1}{2h^2}(x+h-jh)^2, & x \in [x_{j-1}, x_j] \\ -\frac{(x-jh)^2}{h^2} + \frac{1}{2h}(h(2j-1)+2x), & x \in [x_j, x_{j+1}] \\ \frac{1}{2h^2}(x-2h-jh)^2, & x \in [x_{j+1}, x_{j+2}] \\ 0, & x \notin [x_{j-1}, x_{j+2}] \end{cases}$$

1.2. О построении тригонометрических сплайнов минимального дефекта

Для построения тригонометрических базисных сплайнов, таких, что $\text{supp } \omega_j = [x_{j-1}, x_{j+2}]$ рассмотрим три системы уравнений с параметрами $c_{10}, c_{01}, c_{20}, c_{02}$.

Таким образом, при $x \in [x_j, x_{j+1}]$

$$\begin{aligned} \omega_{j-1}(x) + \omega_j(x) + \omega_{j+1}(x) &= 1, \\ \sin(x_{j-1})\omega_{j-1}(x) + \sin(x_j)\omega_j(x) + \sin(x_{j+1})\omega_{j+1}(x) &= c_{10} \sin(x) + c_{01} \cos(x), \\ \cos(x_{j-1})\omega_{j-1}(x) + \cos(x_j)\omega_j(x) + \cos(x_{j+1})\omega_{j+1}(x) &= c_{02} \sin(x) + c_{20} \cos(x), \end{aligned}$$

на промежутке $[x_{j-1}, x_j]$

$$\begin{aligned} \omega_{j-2}(x) + \omega_{j-1}(x) + \omega_j(x) &= 1, \\ \cos(x_{j-2})\omega_{j-2}(x) + \cos(x_{j-1})\omega_{j-1}(x) + \cos(x_j)\omega_j(x) &= c_{02} \sin(x) + c_{20} \cos(x), \end{aligned}$$

и если $x \in [x_{j+1}, x_{j+2}]$ —

$$\begin{aligned} \omega_j(x) + \omega_{j+1}(x) + \omega_{j+2}(x) &= 1, \\ \sin(x_j)\omega_j(x) + \sin(x_{j+1})\omega_{j+1}(x) + \sin(x_{j+2})\omega_{j+2}(x) &= c_{10} \sin(x) + c_{01} \cos(x), \\ \cos(x_{j+1})\omega_j(x) + \cos(x_{j+1})\omega_{j+1}(x) + \cos(x_{j+2})\omega_{j+2}(x) &= c_{02} \sin(x) + c_{20} \cos(x). \end{aligned}$$

Значения параметров $c_{01}, c_{10}, c_{02}, c_{20}$ находим из условия: $\omega_j \in C^1(R^1)$. Таким образом, получаем $c_{02} = -c_{01} = \cos(h/2) \sin(h/2)$, $c_{10} = -c_{20} = \cos^2(h/2)$, и далее на промежутке $[x_j, x_{j+1}]$ находим

$$\begin{aligned} \omega_j(x) &= - \frac{\cos(h) - \cos(x - jh - \frac{h}{2}) \cos(\frac{h}{2})}{\cos(h) - 1}, \\ \omega_{j-1}(x) &= \frac{\cos(x - jh - h) - 1}{2(\cos(h) - 1)}, \quad \omega_{j+1}(x) = \frac{\cos(x - jh) - 1}{2(\cos(h) - 1)}, \end{aligned}$$

Объединяя формулы $\omega_j(x)$, найденные на трех соседних промежутках с одинаковым номером j , получаем:

$$\omega_j(x) = \begin{cases} \frac{(\cos(x - jh + h) - 1)}{2(\cos(h) - 1)}, & x \in [x_{j-1}, x_j], \\ \frac{\cos(h) - \cos(x - jh - \frac{h}{2}) \cos(\frac{h}{2})}{\cos(h) - 1}, & x \in [x_j, x_{j+1}], \\ \frac{(\cos(x - jh - 2h) - 1)}{2(\cos(h) - 1)}, & x \in [x_{j+1}, x_{j+2}]. \end{cases}$$

1.3. Вычисление элементов матрицы

Как известно, задача среднеквадратического приближения приводит к задаче решения системы линейных уравнений с матрицей Грама.

В частном случае, при $n=4$, структура матрицы Грама M будет иметь следующий вид:

$$M = \begin{pmatrix} (\omega_1, \omega_1) & (\omega_2, \omega_1) & (\omega_3, \omega_1) & 0 & 0 & 0 \\ (\omega_1, \omega_2) & (\omega_2, \omega_2) & (\omega_3, \omega_2) & (\omega_4, \omega_2) & 0 & 0 \\ (\omega_1, \omega_3) & (\omega_2, \omega_3) & (\omega_3, \omega_3) & (\omega_4, \omega_3) & (\omega_5, \omega_3) & 0 \\ 0 & (\omega_2, \omega_4) & (\omega_3, \omega_4) & (\omega_4, \omega_4) & (\omega_5, \omega_4) & (\omega_6, \omega_4) \\ 0 & 0 & (\omega_3, \omega_5) & (\omega_4, \omega_5) & (\omega_5, \omega_5) & (\omega_6, \omega_5) \\ 0 & 0 & 0 & (\omega_4, \omega_6) & (\omega_5, \omega_6) & (\omega_6, \omega_6) \end{pmatrix}$$

Вычисляя скалярные произведения, в случае полиномиальных B -сплайнов, получаем следующие значения:

$$\begin{aligned} (\omega_1, \omega_1) &= \int_0^h \left(\frac{1}{2h^2} (x+h-jh)^2 \right)^2 dx, \\ (\omega_2, \omega_2) &= \int_0^h \left(\frac{1}{2h^2} (x+h-jh)^2 \right)^2 dx + \int_h^{2h} \left(-\frac{(x-jh)^2}{h^2} + \frac{1}{2h} (h(2j-1)+2x) \right)^2 dx, \\ (\omega_{n+2}, \omega_{n+2}) &= \int_{1-h}^1 \left(\frac{1}{2h^2} (x-2h-jh)^2 \right)^2 dx, \\ (\omega_{n+1}, \omega_{n+1}) &= \int_{1-h}^1 \left(\frac{1}{2h^2} (x-2h-jh)^2 \right)^2 dx + \\ &+ \int_{1-h}^{1-h} \left(-\frac{(x-jh)^2}{h^2} + \frac{1}{2h} (h(2j-1)+2x) \right)^2 dx, \\ (\omega_j, \omega_j) &= \int_{x_{j-1}}^{x_j} \left(\frac{1}{2h^2} (x+h-jh)^2 \right)^2 dx + \\ &+ \int_{x_j}^{x_{j+1}} \left(-\frac{(x-jh)^2}{h^2} + \frac{1}{2h} (h(2j-1)+2x) \right)^2 dx + \\ &+ \int_{x_{j+1}}^{x_{j+2}} \left(\frac{1}{2h^2} (x-2h-jh)^2 \right)^2 dx. \end{aligned}$$

$$(\omega_j, \omega_{j+1}) = \int_{x_j}^{x_{j+1}} \left(-\frac{(x-jh)^2}{h^2} + \frac{1}{2h}(h(2j-1)+2x) \right) \left(\frac{1}{2h^2}(x+h-(j+1)h)^2 \right) dx + \\ + \int_{x_{j+1}}^{x_{j+2}} \left(\frac{1}{2h^2}(x-2h-jh)^2 \right) \left(-\frac{(x-(j+1)h)^2}{h^2} + \frac{1}{2h}(h(2(j+1)-1)+2x) \right) dx,$$

$$(\omega_j, \omega_{j+2}) = \int_{x_{j+1}}^{x_{j+2}} \left(\frac{1}{2h^2}(x-2h-jh)^2 \right) \left(\frac{1}{2h^2}(x-2h-jh)^2 \right) dx,$$

$$(\omega_{n+1}, \omega_{n+2}) = \int_{1-h}^1 \left(\frac{1}{2h^2}(x-2h-(n+1)h)^2 \right) \left(\frac{1}{2h^2}(x+h-(n+2)h)^2 \right) dx,$$

$$(\omega_1, \omega_2) = \int_0^h \left(\frac{1}{2h^2}(x-3h)^2 \right) \left(-\frac{(x-2h)^2}{h^2} + \frac{1}{2h}(3h+2x) \right) dx,$$

$$(\omega_1, \omega_3) = \int_0^h \left(\frac{1}{2h^2}(x-3h)^2 \right) \left(\frac{1}{2h^2}(x-2h)^2 \right) dx,$$

$$(\omega_n, \omega_{n+2}) = \int_{1-h}^1 \left(\frac{1}{2h^2}(x-2h-nh)^2 \right) \left(\frac{1}{2h^2}(x-2h-(n+2)h)^2 \right) dx.$$

Вычислив интегралы, получим:

$$(\omega_1, \omega_1) = \frac{1}{20}h, (\omega_2, \omega_2) = \frac{1}{2}h, (\omega_{n+1}, \omega_{n+1}) = \frac{1}{2}h, (\omega_{n+2}, \omega_{n+2}) = \frac{1}{20}h,$$

$$(\omega_j, \omega_{j+1}) = \frac{13}{60}h, (\omega_1, \omega_2) = \frac{13}{120}h, (\omega_1, \omega_3) = \frac{1}{120}h,$$

$$(\omega_{n+1}, \omega_{n+2}) = \frac{13}{120}h, (\omega_j, \omega_j) = \frac{11}{20}h, (\omega_j, \omega_{j+2}) = \frac{1}{120}h.$$

Элементы вектора правой части F таковы:

$$(f, \omega_j) = \int_{x_{j-1}}^{x_{j+2}} f(x)\omega_j(x)dx.$$

Вычисляя скалярные произведения в случае тригонометрических сплайнов минимального дефекта, получаем следующие значения:

$$\begin{aligned}
(\omega_j, \omega_j) &= \frac{1}{4} \frac{(\sin(h) \cos(h) - 4\sin(h) + 3h)}{\cos^2(h) - 2\cos(h) + 1} + \\
&+ \frac{1}{4} \frac{(-7\sin(h) \cos(h) + \sin(h) - 4h \cos^2(h) + h)}{\cos^2(h) - 2\cos(h) + 1}, \\
(\omega_1, \omega_1) &= \frac{1}{8} \frac{(\sin(h) \cos(h) - 4\sin(h) + 3h)}{\cos^2(h) - 2\cos(h) + 1}, \\
(\omega_2, \omega_2) &= \frac{1}{4} \frac{(-7\sin(h) \cos(h) + \sin(h) + 4h \cos^2(h) + \cos(h)h + h)}{\cos^2(h) - 2\cos(h) + 1} + \\
&+ \frac{1}{8} \frac{(\sin(h) \cos(h) - 4\sin(h) + 3h)}{\cos^2(h) - 2\cos(h) + 1}, \\
(\omega_N, \omega_N) &= \frac{1}{4} \frac{(-7\sin(h) \cos(h) + \sin(h) + 4h \cos^2(h) + \cos(h)h + h)}{\cos^2(h) - 2\cos(h) + 1} + \\
&+ \frac{1}{8} \frac{(\sin(h) \cos(h) - 4\sin(h) + 3h)}{\cos^2(h) - 2\cos(h) + 1}, \\
(\omega_{N+1}, \omega_{N+1}) &= \frac{1}{8} \frac{(\sin(h) \cos(h) - 4\sin(h) + 3h)}{\cos^2(h) - 2\cos(h) + 1}, \\
(\omega_j, \omega_{j+1}) &= - \frac{1}{8} \frac{(-3 \sin(h) \cos(h) - \sin 3(h) + 5h \cos(h)h + h)}{\cos^2(h) - 2\cos(h) + 1} - \\
&- \frac{1}{12} \frac{(6 \cos(h)h + 4h^3 + 6h - 3 \sin(h)h^2 - 12 \sin(h))}{\frac{h^2}{\cos(h) - 1}},
\end{aligned}$$

$$(\omega_1, \omega_2) = - \frac{\frac{1}{12}(6 \cos(h)h + 4h^3 + 6h - 3 \sin(h)h^2 - 12 \sin(h))}{\frac{h^2}{\cos(h) - 1}},$$

$$(\omega_1, \omega_3) = \frac{\frac{1}{8}(\cos(h)h - 3 \sin(h) + 2h)}{\cos^2(h) - 2 \cos(h) + 1},$$

$$(\omega_{N-1}, \omega_{N+1}) = \frac{\frac{1}{8}(\cos(h)h - 3 \sin(h) + 2h)}{\cos^2(h) - 2 \cos(h) + 1},$$

$$(\omega_N, \omega_{N+1}) = \frac{\frac{1}{8}(-3 \sin(h) \cos(h) - 3 \sin(h) + 5 \cos(h)h + h)}{\cos^2(h) - 2 \cos(h) + 1}.$$

Элементы вектора в правой части F таковы: $(f, \omega_j) = \int_{x_{j-1}}^{x_{j+2}} f(x) \omega_j(x) dx.$

2. Распараллеливание вычислений и оптимизация

2.1. Хранение данных

При решении задач, требующих хранения информации большого объема, встает вопрос о способе хранения данных. В нашем случае матрица системы уравнений имеет ленточный пятидиагональный вид, и ранее было получено, что различных элементов в ней всего 6.

$$A = h \cdot \begin{pmatrix} \frac{1}{20} & \frac{13}{120} & \frac{1}{120} & 0 & 0 & \dots & 0 \\ \frac{13}{120} & \frac{1}{2} & \frac{13}{60} & \ddots & 0 & \dots & 0 \\ \frac{1}{120} & \frac{13}{60} & \frac{11}{20} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{11}{20} & \frac{13}{60} & \frac{1}{120} \\ 0 & \dots & 0 & \ddots & \frac{13}{60} & \frac{1}{2} & \frac{13}{120} \\ 0 & \dots & 0 & 0 & \frac{1}{120} & \frac{13}{120} & \frac{1}{20} \end{pmatrix}$$

В данной работе было рассмотрено два подхода к хранению данных.

При первом подходе мы не учитываем тот факт, что матрица имеет пятидиагональный вид с малым числом различных элементов. При таком подходе мы, очевидно, получим выигрыш в скорости обращения к элементам, так как системная функция обращения к элементу динамического массива работает значительно быстрее, чем функция определения элемента для такой задачи. Однако, для работы с матрицами больших размерностей такой подход не годится, так как такие матрицы будут занимать слишком много места.

Второй подход состоит в том, что реализована специально для этой задачи функция, которая по номеру элемента матрицы определяет его значение. Использование такого подхода гарантирует нам очень большой выигрыш в памяти (вместо N^2 элементов мы храним только 6 для любых размеров матрицы), но дает также сильный проигрыш в скорости.

Был проведен сравнительный анализ времени выполнения программ, иллюстрирующих оба подхода (таблица 2).

Теперь рассмотрим функцию вычисления элемента:

```
get_elem (int i, int j, int n, double h, bool zero_on_diag)
{
    int low=min(i,j);
    if ((i==j) & (i>=3) & (i<=n)) {
        if (zero_on_diag) return 0;
        return 41/20.0*h;
    }
    if ((abs(i-j)==1) & (low>=2) & (low<=n)) {
        //if (zero_on_diag) return 47/60.0*h/(41/20.0*h);
        if (zero_on_diag) return
            47/60.0*h/get_elem(i,i,n,h,false);
        return -47/60.0*h;
    }
    if ((abs(i-j)==2) & (low>=1) & (low<=n)) {
        if (zero_on_diag) return
            -31/120.0*h/get_elem(i,i,n,h,false);
        return 31/120.0*h;
    }
    if ((i==1) & (j==1)) {
        if (zero_on_diag) return 0.0;
        return 31/20.0*h;
    }
    if ((i==2) & (j==2)) {
        if (zero_on_diag) return 0.0;
        return 2*h;
    }
}
```

```

    if ((i==n+1) & (j==n+1)) {
        if (zero_on_diag) return 0;
        return h/2;
    }
    if ((i==n+2) & (j==n+2)) {
        if (zero_on_diag) return 0;
        return 1/20.0*h;
    }
    if ((i==2) & (j==1)) {
        if (zero_on_diag) return 77/120.0*h/(2.0*h);;
        return -77/120.0*h;
    }
    if((i==1) & (j==2)) {
        if (zero_on_diag) return 77/120.0*h/(31/20.0*h);
        return -77/120.0*h;
    }
    if ((i==n+2) & (j==n+1)) {
        if (zero_on_diag) return 17/120.0*h/(1/20.0*h);
        return -17/120.0*h;
    }
    if((i==n+1) & (j==n+2)) {
        if (zero_on_diag) return 17/120.0*h/(1/2.0*h);
        return -17/120.0*h;
    }
    if (zero_on_diag) return -0.0;
    return 0.0;
}

```

Это первый вариант функции вычисления элемента матрицы.

Функцию получения элемента матрицы также можно реализовать с помощью конструкции `switch`, что даст нам небольшое преимущество в скорости. К сожалению, в C++ предусмотрена подстановка в аргументы только константных выражений, поэтому функция выглядит довольно громоздко и в ней есть повторяющиеся участки кода:

```

double get_elem_case (int i, int j, int n, double h, bool
zero_on_diag)
{
    int low=min(i, j);

    switch(i)
    {
    case 1:
        if ((i==1) & (j==1)) {

```

```

        if (zero_on_diag) return 0.0;
        return 31/20.0*h;
    }
    if ((abs(i-j)==2)&(low>=1)&(low<=n)) {
        if (zero_on_diag) return
            -31/120.0*h/get_elem_case(i,i,n,h,false);
        return 31/120.0*h;
    }
    if((i==1)&(j==2)){
        if (zero_on_diag) return 77/120.0*h/(31/20.0*h);
        return -77/120.0*h;
    }
    break;
case 2:
    if ((i==2)&(j==1)){
        if (zero_on_diag) return 77/120.0*h/(2.0*h);
        return -77/120.0*h;
    }
    if ((i==2)&(j==2)){
        if (zero_on_diag) return 0.0;
        return 2*h;
    }
    if ((abs(i-j)==1)&(low>=2)&(low<=n)) {
        if (zero_on_diag)
            return 47/60.0*h/get_elem_case(i,i,n,h,false);
        return -47/60.0*h;
    }
    if ((abs(i-j)==2)&(low>=1)&(low<=n)) {
        if (zero_on_diag) return
            -31/120.0*h/get_elem_case(i,i,n,h,false);
        return 31/120.0*h;
    }
    break;
default:
    if ((i==j)&(i>=3)&(i<=n)) {
        if (zero_on_diag) return 0;
        return 41/20.0*h;
    }
    if ((abs(i-j)==1)&(low>=2)&(low<=n)) {
        if (zero_on_diag) return
            47/60.0*h/get_elem_case(i,i,n,h,false);
        return -47/60.0*h;
    }
    if ((abs(i-j)==2)&(low>=1)&(low<=n)) {

```

```

    if (zero_on_diag) return
        -31/120.0*h/get_elem_case(i,i,n,h,false);
    return 31/120.0*h;
}
if ((i==n+1)&(j==n+1)){
    if (zero_on_diag) return 0;
    return h/2;
}
if ((i==n+2)&(j==n+2)){
    if (zero_on_diag) return 0;
    return 1/20.0*h;
}
if ((i==n+2)&(j==n+1)){
    if (zero_on_diag) return 17/120.0*h/(1/20.0*h);
    return -17/120.0*h;
}
if((i==n+1)&(j==n+2)){
    if (zero_on_diag) return 17/120.0*h/(1/2.0*h);
    return -17/120.0*h;
}
break;
}
if (zero_on_diag) return -0.0;
return 0.0;
}

```

Сравнительный анализ времени выполнения приведен в таблице 3.

2.2. Поиск оптимального параметра ω

Пусть система приведена к виду $x=Hx+g$. Оптимальное значение параметра ω находится по формуле:

$$\omega = \frac{1}{1 + \sqrt{1 - \rho^2(H)}}.$$

Для поиска спектрального радиуса, в программе реализован степенной метод:

```

lambda2=lambda+1;
double temp2=0;
while (abs(lambda-lambda2)>eps)
{
    k++;

```

```

lambda2=lambda;
for (i=1;i<=n+2;i++)
{
    npH2[i]=npH[i];
    npH[i]=0;
}
for (i=1;i<=n+2;i++)
{
    for (j=1;j<=n+2;j++)
    {
        npH[i]=npH[i]+
        npH2[j]*get_elem(i,j,n,h,false);
    }
}
lambda=npH[n/2]/npH2[n/2];
}
omega=2/(1+sqrt(1-lambda*lambda));

```

При оптимизации циклов такого рода, где неизвестно количество итераций, удобно воспользоваться следующим приемом: заменим тело цикла на цикл `for`, который уже можно распараллелить и будем проверять условие завершения метода после нескольких итераций внутреннего цикла. При таком подходе возникает проблема определения оптимального числа итераций внутреннего цикла.

```

lambda2=lambda+1;
double temp2=0;
while (abs(lambda-lambda2)>eps)
{
    #pragma omp parallel for
    for (i2=0;i2<iter_size;i2++){
        k++;
        lambda2=lambda;
        for (i=1;i<=n+2;i++)
        {
            npH2[i]=npH[i];
            npH[i]=0;
        }
        for (i=1;i<=n+2;i++)
        {
            for (j=1;j<=n+2;j++)
            {
                npH[i]=npH[i]+
                npH2[j]*get_elem(i,j,n,h,false);
            }
        }
    }
}

```

```

    }
  }
  lambda=npH[n/2]/npH2[n/2];
}
}
omega=2/(1+sqrt(1-lambda*lambda));

```

Второй подход состоит в распараллеливании только внутреннего цикла, в котором матрица умножается на вектор.

Сравнительный анализ времени выполнения в таблице (5).

2.3. Решение системы методом верхней релаксации

Рассмотрим решение системы уравнений $x = Hx + g$, где x — вектор-решение, а $H = (h_{ij})_{i,j=1}^n$.

Расчетная формула имеет вид:

$$x_i^{k+1} = x_i^k + \omega \sum_{j=1}^{i-1} x_j^{k+1} h_{ij} - \omega \sum_{j=i+1}^n x_j^{k-1} h_{ij} + \omega f_i.$$

При вычислении следующего приближения x_i^{k+1} можно оптимизировать формулу: при хранении текущего и следующего приближения в одном векторе можно произвести оба суммирования в одном цикле:

```

for (i=1; i<=n+2; i++)
{
  dtmp2=0;
  for (j=1; j<=n+2; j++)
  {
    dtmp2=dtmp2+get_elem(i, j, n, h, true) * x[j]; //x_k+1[i]
  }
  x[i]=omega*F[i]+omega*dtmp2-x1[i]*(omega-1);
}

```

Первый подход к оптимизации аналогичен рассмотренному ранее подходу при вычислении параметра ω степенным методом. Внутри общего цикла помещается цикл for работающий фиксированное число итераций.

```

do{
  k2++;
  #pragma omp parallel for shared(x, omega, F, n)
  private(i, j, dtmp2, x1)
  for (int i5=0; i5<iter_size; i5++){

```

```

        //вычисления
    }
}
while (dtemp1>pogr);

```

Второй подход предполагает распараллеливание только внутреннего цикла с тем же набором общих и разделяемых переменных.

```

do {
    k2++;
    //x1=x;//запоминаем предыдущую итерацию
    for (int i4=1;i4<=n+2;i4++)
    {
        x1[i4]=x[i4];
    }
    #pragma omp parallel for shared(x,omega,F,n)
        private(i,j,dtmp2,x1)
    for (int i=1;i<=n+2;i++)
    {
        //вычисления
    }
}
while (dtemp1>pogr);

```

На матрицах размера $n \times n$, $n < 5000$, второй подход показывает даже лучшее время, чем первый.

Сравнительный анализ времени выполнения:

	Обычный цикл	Параллельный цикл (I)	Параллельный цикл (II)
100	172	187	141
500	3 338	2 512	2 324
1 000	9 937	9 375	9 220
2 000	38 517	36 535	38 652
5 000	243 298	231 962	249 818

Результаты

Целью работы было произвести оптимизацию времени расчетов. Были произведены замеры времени на различных матрицах. Результаты приведены в сводной таблице. Здесь и далее все измерения проводились при погрешности метода релаксации ε , равной 0.00001 и при априорном фиксированном количестве итераций (для построения эффективной параллельной программы) равном 30. Время в таблицах указано в миллисекундах.

1). Сравнительный анализ производительности
(римская цифра I указывает на обычный способ хранения матрицы, а II — на оптимизированный)

N (размер матрицы)	Последовательная программа (I)	Последовательная программа (II)	Оптим. программа (I)	Оптим. программа (II)
100	31	156	47	97
250	109	827	78	301
500	436	4415	218	1193
750	568	7878	364	2640
1000	1466	12386	827	4704
2000	2824	43539	2871	18160
5000	18439	264077	18081	133401

2). Сравнительный анализ способов хранения данных:
(римская цифра I указывает на обычный способ хранения матрицы, а II — на оптимизированный)

N (размер матрицы)	Последовательная программа (I)	Последовательная программа (II)	Оптим. программа (I)	Оптим. программа (II)
100	31	156	47	172
250	109	827	78	796
500	436	4415	218	2886
750	568	7878	364	6147
1000	1466	12386	827	10967
2000	2824	43539	2871	40544
5000	18439	264077	18081	252008

3). Сравнительный анализ времени доступа к элементу: (общее время доступа ко всем элементам матрицы)

N (размер матрицы)	без switch	switch	встроенное
250	6	3	0
500	17	12	0
750	51	48	0
1000	61	49	0
2000	220	196	15
5000	1404	1355	62
10000	5725	5727	234

4). Сравнительный анализ степенного метода:

	обычный	оптимизированный (1)	оптимизированный (2)
100	94	141	47
500	951	1 001	562
1 000	2 450	2 867	1 388
2 000	5 709	8 689	4 711
5 000	19 172	49 301	18 027
10 000	52 931	174 428	51 261

Целью приближения функции сплайнами была очистка функции от «помех», искажающих истинное значение.

Результат среднеквадратического приближения функции

$$f(x) = \sin(x) + \frac{1}{10} \sin(50x)$$

полиномиальными В-сплайнами второй степени при $h = 1/4$ и $n = 4$ можно видеть на рис.1

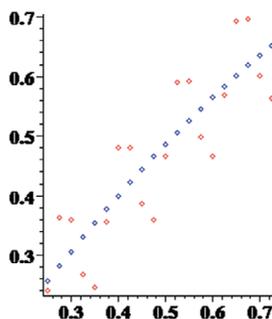


Рис. 1.

Заключение

Благодаря особенностям распараллеливания, в оптимизированных программах реализуется выигрыш в точности вычислений (а благодаря многопоточности, одновременно и выигрыш во времени). Оптимальная функция доступа к элементу матрицы обеспечит нам еще больший выигрыш в скорости. Точная оценка числа итераций в общем случае является достаточно серьезной проблемой, а выигрыш в точности обеспечен на любой матрице. Программа демонстрирует мощь Open MP как средства оптимизации и языка C++, как средства реализации объемных вычислений.

Литература

1. *Воеводин В.В., Кузнецов Ю.А.* Матрицы и вычисления. 1984 .
 2. *Рябенский В.С.* Введение в вычислительную математику. 2000.
 3. *Фаддеев Д.К. Фаддеева В.Н.*, Вычислительные методы линейной алгебры. 2009.
 4. *Бурова И.Г. Евдокимова Т.О.*, Приближения неполиномиальными сплайнами минимального дефекта. 2007.
-

О РАСПАРАЛЛЕЛИВАНИИ РЕШЕНИЯ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ ЛИНЕЙНОЙ ВЯЗКОУПРУГОСТИ

Н.И. Лещенко

*аспирантка кафедры вычислительной математики СПбГУ,
m02pni@star.math.spbu.ru*

Для ускорения вычислений на параллельной системе возникает острая необходимость иметь алгоритмы решения задач, пригодные для распараллеливания вычислений. Традиционно для решения интегральных уравнений линейной вязкоупругости применяются последовательные алгоритмы (см. ниже). В этой статье предлагается алгоритм, пригодный для одновременного вычисления решения в различных точках. Таким образом, на основе этого алгоритма в среде OPEN MP с помощью директивы `parallelsections` можно организовать вычисления одновременно на нескольких процессорах, что существенно сократит время решения задачи.

При решении численными методами задач, описываемых интегро-дифференциальными уравнениями, приближённое решение находится для возрастающих значений аргумента последовательно. И если есть необходимость в большом количестве шагов, то процесс может оказаться трудоёмким и долгим. Интегральные же преобразования позволяют эту трудность обойти, поскольку вся информация о решении заключена в его образе и приближения к решению в различных точках находятся независимо друг от друга. Правда, для этого надо уметь решать задачу обращения — поиска решения по его образу. Мы рассматриваем задачи линейной вязкоупругости, для которых характерны длительные медленно протекающие процессы, и применение интегральных преобразований для их решения дает определенные преимущества по сравнению с прямыми методами решения исходной задачи, тем более, что очевидна возможность распараллеливания задачи поиска решения в нескольких не зависящих друг от друга точках.

В задачах линейной вязкоупругости [1], описывающих напряженное состояние на основе определяющего соотношения Больцмана—Вольтерра (пространственные координаты ниже для простоты опущены)

$$\varepsilon(t) = \frac{1}{E} \left(\sigma(t) + \beta \int_0^t K(t-\tau) \sigma(\tau) d\tau \right), \quad (1)$$

сравнительно просто находятся изображения по Лапласу решений (деформаций $\varepsilon(t)$ и напряжений $\sigma(t)$) из соответствующих интегральных уравнений вида (1). Сами решения медленно изменяются на конечном по t отрезке времени и допускают хорошие приближения вида $t^{s-1}Q(t^a)$ при

$0 < a < 1$ где $Q(t)$ — некоторый многочлен. Такая ситуация характерна для длительных процессов деформирования [2]. Изображения таких функций равны $p^{-s}R(p^{-a})$, где $R(x)$ — многочлен той же степени, что и $Q(x)$.

Важнейшей задачей становится выбор подходящего ядра интегрального уравнения, описывающего поведение материала, и определение его параметров. Ядро K должно иметь интегрируемую особенность в точке $t=0$ [1]. Чаще всего в качестве такового берут дробно-экспоненциальную функцию Ю.Н. Работнова [1]

$$\mathfrak{E}_\alpha(\beta, t) = t^\alpha \sum_{k=0}^{\infty} \frac{(\beta t^{1+\alpha})^k}{\Gamma((1+\alpha)(1+k))}, \quad -1 < \alpha \leq 0. \quad (2)$$

Интеграл от этого ядра по полуоси $t \geq 0$ должен быть конечным, для чего необходимо $\beta < 0$. Не умаляя общности, далее считаем $\beta = -1$, и пусть символ $\mathfrak{E}_\alpha(t)$ означает $\mathfrak{E}_\alpha(-1, t)$.

В наследственной механике твердого тела наряду с функцией (2) широко используется и интеграл от нее с переменным верхним пределом. Для облегчения использования этих величин составлены таблицы функций [1], [3]

$$F_1(\alpha, x) = t^{-\alpha} \mathfrak{E}_\alpha(x), \quad F_2(\alpha, x) = t^{-\alpha-1} \int_0^t \mathfrak{E}_\alpha(\tau) d\tau, \quad x = t^{\alpha+1}.$$

Однако при решении конкретных задач необходимо вводить в память вычислительной машины части этих таблиц, соответствующие найденным параметрам \mathfrak{E}_α — функций, которые к тому же заранее неизвестны и определяются в процессе решения задачи (и в итоге таковых в таблице может не оказаться). При изменении параметров приходится эту работу проделывать заново, что неудобно и сопряжено с внесением ошибок.

Таким образом, на первом шаге решения уравнения (1) мы находим изображения искомых напряжений или деформаций. Затем необходимо обратит преобразование Лапласа. Аналогичная задача возникает, если ядра ползучести или релаксации задавать их изображениями (вычисление ядер по явным формулам в виде рядов (2) возможно лишь при малых значениях аргумента, с ростом которого члены ряда быстро растут по модулю, что затрудняет задачу суммирования ряда).

Один из подходов к решению задач линейной вязкоупругости был представлен в работе [4]. Здесь же мы рассмотрим задачу вычисления дробно-экспоненциальных функций и интегралов от них с помощью деформации контура интегрирования. Ниже будет дано более развёрнутое изложение материала из статьи [5].

2. ДЕФОРМАЦИЯ КОНТУРА ИНТЕГРИРОВАНИЯ

Задача обращения интегрального преобразования Лапласа состоит в нахождении решения уравнения

$$F(p) = \int_0^{\infty} e^{-px} f(x) dx,$$

в котором $F(p)$ — известное изображение, $f(x)$ — искомый оригинал. Для простоты будем считать, что функция $F(p)$ регулярна в полуплоскости $\operatorname{Re}(p) > 0$, чего всегда можно добиться домножением оригинала на соответствующую экспоненту. Как правило, точное обращение осуществить не удастся, и потому возникает необходимость разработки и применения приближенных методов. Наиболее полно возможные подходы к задаче обращения и их реализация описаны в книге [6]. В первую очередь следует назвать построение квадратурных формул для приближенного вычисления интеграла Римана—Меллина

$$f(t) = L^{-1}(F)(t) = \int_{c-i\infty}^{c+i\infty} e^{pt} F(p) dp, \quad c > 0 \quad (3)$$

задающего обращение преобразования Лапласа. Напомним, что интеграл (3) понимается в смысле главного значения, он не зависит от c и в случае разрыва оригинала в точке t мы получаем полусумму предельных значений оригинала слева и справа от точки t .

Не существует универсального метода вычисления этого интеграла, дающего удовлетворительные результаты для произвольного изображения $F(p)$. Любой конкретный метод обращения должен учитывать специфику поведения изображения (или функции-оригинала), что прежде всего находит отражение в выборе подходящих систем функций в пространствах оригиналов и изображений, с которыми легко работать и с помощью которых могут быть хорошо приближены заданные образы и оригиналы.

Изображения по Лапласу функции $\mathfrak{E}_\alpha(t)$ и интеграла от нее $\int_0^t \mathfrak{E}_\alpha(\tau) d\tau$ равны, соответственно,

$$\frac{1}{p^\alpha + 1}, \quad \frac{1}{p(p^\alpha + 1)}, \quad \alpha = 1 + \alpha.$$

По этим изображениям легко находятся образы напряжений $\sigma(t)$ и деформаций $\varepsilon(t)$ из уравнения (1).

Положим в формуле обращения (3) $p = c + i\tau$, тогда $\exp(pt) = \exp(ct) \cdot \exp(i\tau t)$. При фиксированном t первый множитель постоянен, а второй пробегает единичную окружность на комплексной плоскости бесконечное

число раз. С ростом t первый сомножитель и скорость пробегания окружности вторым сомножителем неограниченно возрастают, так что попытка приблизить интеграл в (3) римановыми суммами вряд ли приведет к цели. Например, в простейшем случае для $f(t)=1$ имеем $F(p)=1/p$, так что при любом $c>0$ сомножитель $\exp(ct)$ быстро растет с увеличением t , однако оригинал постоянен и не зависит от c .

С целью уменьшения осцилляций сомножителя $\exp(i\tau t)$ заменим линию интегрирования в (3) контуром L , предложенным в работе [7]:

$$L = \left\{ p \mid p = \frac{2\pi yi}{1 - \exp(-2\pi yi)}, \quad y \in [-1, 1] \right\}.$$

Этот контур состоит из двух симметричных относительно вещественной оси плоскости p ветвей, исходящих из точки $p(0)=1$ налево и при $y \rightarrow \pm 1$ стремящихся к асимптотам $p = \pm \pi i$. Для изображения выполнены условия леммы Жордана [8] и кривая L содержит внутри себя особые точки изображения, так что мы имеем

$$f(t) = \int_{-1}^1 \exp(p(y)t) F(p(y)) p'(y) dx.$$

Изображение Лапласа функции $\mathcal{E}_\alpha(t)$ имеет точку ветвления при $p=0$, а изображение интеграла от нее еще и простой полюс в той же точке [1]. Для устранения многозначности достаточно выбрать одну из ветвей, что делается стандартным образом: приведенные выше преобразования Лапласа функции $\mathcal{E}_\alpha(t)$ и интеграла от нее не имеют особенностей на комплексной плоскости $C \setminus R_-$ с разрезом вдоль полупрямой

$$R_- = \{ p \in C : \operatorname{Im}(p) = 0, \operatorname{Re}(p) \leq 0 \}.$$

Наши изображения $F(p)$ фактически зависят от p^a , т. е. $F(p) = \Phi_1(p^a)$. Введем в рассмотрение функции

$$F^\pm(t) = \Phi_1(t^a \exp(\pm ia\pi)), \quad t > 0.$$

Очевидно, $F^+(t) = \overline{F^-(t)}$ в силу вещественности функции-оригинала. Воспользуемся полученным в работе [9] следующим результатом:

Лемма. Пусть выполнены условия

(А) $F(p) = o(1)$ при $|p| \rightarrow \infty$, $F(p) = o(|p|^{-1})$ при $|p| \rightarrow 0$ равномерно в любом секторе $|\arg p| < \pi - \eta$, $\pi > \eta > 0$;

(Б) существует $\varepsilon > 0$ такое, что для любого φ , удовлетворяющего неравенству $\pi - \varepsilon < \varphi \leq \pi$, справедливы соотношения

$$\frac{F(r \exp(\pm i\varphi))}{1+r} \in L_1(R_+), \quad |F(r \exp(\pm i\varphi))| \leq \alpha(r),$$

где $\alpha(r)$ не зависит от φ и $\alpha(r) \exp(-\sigma r) \in L_1(R_+)$ для любого $\delta > 0$. Тогда

$$f(x) = L^{-1}(F)(x) = \frac{1}{\pi} \int_0^{\infty} e^{-xt} \operatorname{Im} F^{-}(t) dt. \quad (4)$$

З а м е ч а н и е. Утверждение леммы получается в результате замены линии интегрирования в (3) контуром, состоящим из нижнего и верхнего разрывов, соединенных окружностью сколь угодно малого радиуса с центром в точке $p=0$.

Пусть $F(p) = 1/(pa+1)$, тогда

$$F^{-}(t) = \operatorname{Im} \frac{1}{t^a \exp(-i\pi a) + 1} = \frac{t^a \sin \pi a}{1 + 2t^a \cos \pi a + t^{2a}},$$

так что выполнены все условия леммы и формула (4) дает

$$\begin{aligned} \mathfrak{A}_a(x) &= \frac{\sin \pi a}{\pi} \int_0^{\infty} e^{-xt} \frac{t^a dt}{1 + 2t^a \cos \pi a + t^{2a}} = \\ &= x^{a-1} \frac{\sin \pi a}{\pi} \int_0^{\infty} e^{-xt} \frac{z^a e^{-z} dz}{z^{2a} + 2z^a x^a \cos \pi a + x^{2a}}. \end{aligned} \quad (5)$$

При $x \rightarrow 0$ последний интеграл в представлении (5) стремится к величине $\Gamma(1-a)$, и с учетом формулы $\Gamma(a)\Gamma(1-a) = \pi/\sin \pi a$ при $x \rightarrow 0$ из представления (5) получаем $\mathfrak{A}_a(x) \approx x^{a-1}/\Gamma(a)$, что совпадает с первым членом ряда (2).

Положим

$$g_a(x) = \int_0^x \mathfrak{A}_a(t) dt.$$

Изображение этой функции равно

$$G_a(p) = \frac{1}{p(p^a+1)}. \quad (6)$$

Для нее не выполняется условие (A) теоремы 2 (при $p \rightarrow 0$ величина $|G_a(p)|$ слишком быстро возрастает).

Представим $G_a(p)$ в виде

$$G_a(p) = \frac{1}{p} - \frac{p^{a-1}}{p^a + 1} \tag{7}$$

и положим

$$Q_a(p) = \frac{p^{a-1}}{p^a + 1} \tag{8}$$

Обозначим через $q_a(x)$ функцию-оригинал с изображением (8). Формула (7) означает, что $g_a(x) = 1 - q_a(x)$.

Изображение (8) удовлетворяет условиям теоремы 2, для него находим

$$F^-(t) = \text{Im} \frac{t^{a-1} \exp(-i\pi(a-1))}{t^a \exp(-i\pi a) + 1} = \frac{t^{a-1} \sin \pi a}{1 + 2t^a \cos \pi a + t^{2a}}.$$

Следовательно,

$$\begin{aligned} q_a(x) &= \frac{\sin \pi a}{\pi} \int_0^\infty e^{-xt} \frac{t^{a-1} dt}{1 + 2t^a \cos \pi a + t^{2a}} = \\ &= x^a \frac{\sin \pi a}{\pi} \int_0^\infty e^{-xz} \frac{z^{a-1} e^{-z} dz}{z^{2a} + 2z^a x^a \cos \pi a + x^{2a}}. \end{aligned} \tag{9}$$

Из определения (6) следует равенство

$$g_a(0) = \lim_{p \rightarrow \infty} pG_a(p) = 0,$$

поэтому необходимо $q_a(0) = 1$. Подставив $x=0$ в первый интеграл в представлении (9) и сделав замену $t^a = z$, придем к табличному легко вычисляемому интегралу и таким образом убедимся в справедливости равенства $q_a(0) = 1$ при всех $a > 0$.

Итак, наши задачи обращения преобразования Лапласа свелись к вычислению интегралов

$$\mathfrak{E}_a(x) = x^{a-1} \frac{\sin \pi a}{\pi} \int_0^\infty \frac{z^a e^{-z} dz}{z^{2a} + 2z^a x^a \cos \pi a + x^{2a}}; \tag{10}$$

$$q_a(x) = x^a \frac{\sin \pi a}{\pi} \int_0^\infty \frac{z^{a-1} e^{-z} dz}{z^{2a} + 2z^a x^a \cos \pi a + x^{2a}}. \tag{11}$$

Для их приближенного вычисления можно применить квадратурные формулы типа Гаусса [10] с весом Лагерра $z^a e^{-z}$ для первого интеграла и с весом $z^{a-1} e^{-z}$ для второго интеграла. Однако с уменьшением a точность формул будет уменьшаться.

Поэтому для приближенного вычисления интегралов (10) и (11) построим обобщенные квадратурные формулы вида

$$\int_0^{\infty} z^{\beta} e^{-z} f(z) dz \approx \sum_{k=1}^n A_k f(z_k), \quad \beta > -1, \quad (12)$$

точные для функций $f(z) = z^{am}$, $m = 0, 1, \dots, 2n - 1$.

Теорема 1. Для того чтобы формула (12) была точна для функций $f(z) = z^{am}$, $m = 0, 1, \dots, 2n - 1$ необходимо и достаточно выполнение двух условий:

- 1) формула (12) интерполяционная;
- 2) построенный по узлам формулы (12) многочлен

$$\omega_n(z) = \prod_{k=1}^n (z - z_k^a) \quad (13)$$

удовлетворяет условиям

$$\int_0^{\infty} z^{\beta} e^{-z} \omega_n(z^a) z^{am} dz = 0, \quad m = 0, 1, \dots, n - 1. \quad (14)$$

Доказательство этой теоремы проводится точно так же, как в случае классических формул типа Гаусса [10], и мы не будем его здесь повторять.

Покажем, что многочлен (13), удовлетворяющий условиям (14), существует и определяется однозначно.

После замены переменной $z^a = x$ условия (14) принимают вид

$$\int_0^{\infty} x^{(\beta+1)/a-1} \exp(-x^{1/a}) \omega_n(x) x^m dx = 0, \quad m = 0, 1, \dots, n - 1.$$

Функция $\omega(x) = x^{(\beta+1)/a-1} \exp(-x^{1/a})$ обладает свойствами веса на полуоси $(0, \infty)$, поскольку $(\beta+1)/a > 0$, следовательно, искомый многочлен существует и единствен, а его корни, т. е. $z_k^a = 1, 2, \dots, n$, попарно различны и положительны. Все коэффициенты формулы положительны. Итак, квадратурная формула типа Гаусса вида (12) существует.

Опишем способ вычисления узлов и коэффициентов формулы (12).

Будем искать многочлен (13) в виде

$$\omega_n(z) = z_n + b_1 z^{n-1} + \dots + b_n$$

с неизвестными коэффициентами b^k .

Условия (14) приводят к системе линейных алгебраических уравнений

$$\sum_{j=1}^n \Gamma(\beta + (k+n-j)a+1) b_j = -\Gamma(\beta + (k+n)a+1), \quad k=0, 1, \dots, n-1.$$

Ее решение существует и единственно, как показано выше. Далее находим корни уравнения $\omega_n(z) = 0$, т. е. числа z_k^a , $k=1, 2, \dots, n$.

Коэффициенты формулы (12) определяем из системы уравнений

$$\sum_{j=1}^n A_k (z_k^a)^{j-1} = \Gamma(\beta + (j-1)a+1), \quad j=1, 2, \dots, n.$$

Заметим, что узлы и коэффициенты формулы (12) вещественны.

Литература

1. *Работнов Ю.Н.* Элементы наследственной механики твердых тел. М., 1977. 384 с.
2. *Екельчик В.С., Рябов В.М.* Об использовании одного класса наследственных ядер в линейных уравнениях вязкоупругости // *Механика композитных материалов*. 1981. №3. С. 393–404.
3. *Работнов Ю.Н., Паперник Л.Х., Звонов Е.Н.* Таблицы дробно-экспоненциальной функции отрицательных параметров и интеграла от нее. М., 1969. 132 с.
4. *Порошина Н.И.* Специальные квадратурные формулы обращения интегрального преобразования Лапласа // *Список-2012: Материалы всероссийской научной конференции по проблемам информатики*. 2012. С. 259–266.
5. *Порошина Н.И., Рябов В.М.* Об обращении преобразования Лапласа некоторых специальных функций // *Вестн. С.-Петерб. ун-та*. 2009. Сер. 1. Вып. 3. С. 50–60.
6. *Крылов В.И., Скобля Н.С.* Методы приближенного преобразования Фурье и обращения Лапласа. М., 1974. 224 с.
7. *Talbot A.* The accurate numerical inversion of Laplace transform // *J. Inst. Maths. Applics*. 1979. Vol. 23. P. 97–120.
8. *Лаврентьев М.А., Шабат Б.В.* Методы теории функций комплексного переменного. М., 2002. 688 с.
9. *Bobylev A.V., Cercignani C.* The inverse Laplace transform of some analytic functions with an application to the eternal solutions of the Boltzmann equation // *Applied Mathematics Letters*. 2002. Vol. 15. P. 807–813.
10. *Мысовских И.П.* Лекции по методам вычислений. СПб., 1998. 472 с.

РАСПАРАЛЛЕЛИВАНИЕ РЕШЕНИЯ ЗАДАЧИ КОШИ, МЕТОДАМИ С ПРИМЕНЕНИЕМ ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫХ СПЛАЙНОВ

Абдурахимова А.С.

421 г., Математико-механический факультет СПбГУ, im0to@mail.ru

Бурова И.Г.

профессор каф. Вычислительной математики.

Математико-механический факультет СПбГУ, burovaig@mail.ru

Верификация решения задачи Коши производится, например, с помощью вычислений разными численными методами. Таким образом требуется решать задачу несколько раз.

В этой работе рассмотрим параллельное вычисление в OPEN MP решения задачи Коши двумя численными методами, представленными одним из авторов на международной конференции [1]. Эти методы разработаны на основе неполиномиальных интегродифференциальных сплайнов [2].

Решаем задачу Коши

$$y' = f(x, y(x)), \quad y(x_0) = y_0, \quad x \in [x_0, X].$$

Пусть n — целое число, на промежутке $[x_0, X]$ построена сетка узлов

$$\{x_j\}: x_0 = \dots < x_j < x_{j+1} < \dots < x_n = X.$$

1) На промежутке $[x_j, x_{j+1}]$ функцию $u(x)$ приближаем выражением

$$\begin{aligned} \tilde{u}(x) = & u(x_j)\omega_{j,0}(x) + u(x_{j+1})\omega_{j+1,0}(x) + \\ & + \left(\int_{x_{j-1}}^{x_j} u(x)dx \right) \omega_{j-1}^{<-1>}(x). \end{aligned}$$

Здесь $u(x_j)$ — значения функции u в узлах сетки x_j , а базисные сплайны $\omega_{j,0}(x)$, $\omega_{j+1,0}(x)$, $\omega_{j-1}^{<-1>}(x)$ находим из условий

$$\tilde{u}(x) = u(x), \quad u(x) = \varphi_1(x), \varphi_2(x), \varphi_2(x).$$

Заменяя в тождестве Ньютона-Лейбница

$$y(x_{j+1}) = y(x_j) + \int_{x_j}^{x_{j+1}} y'(x)dx$$

под знаком интеграла производную на $\tilde{u}(x)$

$$y(x_{j+1}) = y(x_j) + \int_{x_j}^{x_{j+1}} \tilde{u}(x) dx + R,$$

получаем

$$y(x_{j+1}) = y(x_j) + u(x_j) \int_{x_j}^{x_{j+1}} \omega_{j,0}(x) dx + u(x_{j+1}) \int_{x_j}^{x_{j+1}} \omega_{j+1,0}(x) dx + \\ + \int_{x_{j-1}}^{x_j} u(x) dx \int_{x_j}^{x_{j+1}} \omega_{j-1}^{<-1>}(x) dx + R.$$

Отсюда имеем соотношение для неявного интерполяционного метода решения задачи Коши

$$y_{j+1} = y_j(1 + I^{<-1>}) - y_{j-1}(I^{<-1>}) + f(x_j, y_j)I_0 + f(x_{j+1}, y_{j+1})I_1,$$

где

$$I^{<-1>} = \int_{x_j}^{x_{j+1}} \omega_{j-1}^{<-1>}(x) dx, \quad I_0 = \int_{x_j}^{x_{j+1}} \omega_{j,0}(x) dx, \quad I_1 = \int_{x_j}^{x_{j+1}} \omega_{j+1,0}(x) dx.$$

Вычисляя интегралы, получаем

$$I_0 = -A_0/B_0,$$

$$A_0 = \left(-\frac{h}{2} - 1\right)e^{(-h)} + \left(1 - \frac{h}{2}\right)e^{(h)} + (2+h)e^{(-h/2)} + (h-2)e^{(h/2)} - he^{(0)},$$

$$B_0 = \left(-1 + \frac{h}{4}\right)e^{(-h/2)} + \left(-1 - \frac{h}{4}\right)e^{(h/2)} + \frac{1}{2}e^{(h)} + e^{(0)} + \frac{1}{2}e^{(-h)},$$

$$I_1 = -A_1/B_1,$$

$$A_1 = (2-h)e^{(-h/2)} + (-2-h)e^{(h/2)} - e^{(-h)} + 2he^{(0)} + e^{(h)},$$

$$B_1 = \left(-1 + \frac{h}{4}\right)e^{(-h/2)} + \left(-1 - \frac{h}{4}\right)e^{(h/2)} + \frac{1}{2}e^{(h)} + e^{(0)} + \frac{1}{2}e^{(-h)},$$

причем

$$|R| \leq Kh^4 \|4y^{IV} - y^{II}\|_{[x_{j-1}, x_{j+1}]}, \quad K > 0.$$

В случае полиномиальной системы базисных функций

$$\varphi_1(x)=1, \quad \varphi_2(x)=x, \quad \varphi_3(x)=x^2,$$

имеем соотношение для неявного метода решения задачи Коши

$$y_{j+1}=y_j\left(\frac{4}{5}\right)-y_{j-1}\left(\frac{-1}{5}\right)+f(x_j, y_j)\frac{4h}{5}+f(x_{j+1}, y_{j+1})\frac{2h}{5},$$

$$|R|\leq Kh^4\|y^{IV}\|_{[x_{j-1}, x_{j+1}]}, \quad K>0.$$

Пример.

Задача Коши

$$y'=f(x, y), \quad f(x, y)=-50y+50\cos(x), \quad y(0)=0,$$

имеет точное решение $y=\frac{2500}{2501}\cos(x)+\frac{50}{2501}\sin(x)-\frac{2500}{2501}e^{(-50x)}$.

В случае тригонометрической системы базисных функций

$$\varphi_1(x)=1, \quad \varphi_2(x)=\sin(x), \quad \varphi_3(x)=\cos(x),$$

$$I_0=2\frac{\cos^2(h)+h\sin(h)\cos(h)-1}{2\sin(h)\cos(h)-h\cos(h)-h}, \quad I_1=-2\frac{\cos^2(h)+h\sin(h)-1}{2\sin(h)\cos(h)-h\cos(h)-h},$$

$$I^{<-1>}=\frac{-h\cos(h)+2\sin(h)-h}{2\sin(h)\cos(h)-h\cos(h)-h}.$$

2) Аналогично можно получить неявный метод решения задачи Коши четвертого порядка:

$$\begin{aligned} \tilde{u}(x) &= u(x_j)\omega_{j,0}(x)+u(x_{j+1})\omega_{j+1,0}(x)+ \\ &+ \left(\int_{x_{j-1}}^{x_j} u(x)dx\right)\omega_{j-1}^{<-1>}(x)+\int_{x_{j-2}}^{x_j} u(x)dx\left)\omega_{j-2}^{<-2>}(x). \end{aligned}$$

Здесь базисные сплайны $\omega_{j,0}(x)$, $\omega_{j+1,0}(x)$, $\omega_{j-1}^{<-1>}$, $\omega_{j-2}^{<-2>}$ находим изусловий

$$\tilde{u}(x)=u(x), \quad u(x)=\varphi_1(x), \varphi_2(x), \varphi_3(x), \varphi_4(x).$$

Заменяя в тождестве Ньютона-Лейбница

$$y(x_{j+1}) = y(x_j) + \int_{x_j}^{x_{j+1}} y'(x) dx$$

под знаком интеграла производную на $\tilde{u}(x)$

$$y(x_{j+1}) = y(x_j) + \int_{x_j}^{x_{j+1}} \tilde{u}(x) dx + R,$$

получаем

$$y(x_{j+1}) = y(x_j) + u(x_j) \int_{x_j}^{x_{j+1}} \omega_{j,0}(x) dx + u(x_{j+1}) \int_{x_j}^{x_{j+1}} \omega_{j+1,0}(x) dx + \\ + \int_{x_{j-1}}^{x_j} u(x) dx \int_{x_j}^{x_{j+1}} \omega_{j-1}^{<-1>}(x) dx + \int_{x_{j-2}}^{x_j} u(x) dx \int_{x_j}^{x_{j+1}} \omega_{j-2}^{<-2>}(x) dx + R.$$

Отсюда имеем соотношение для неявного интерполяционного метода решения задачи Коши

$$y_{j+1} = y_j(1 + I^{<-1>} + I^{<-2>}) - y_{j-1}(I^{<-1>}) - y_{j-2}(I^{<-2>}) + \\ + f(x_j, y_j)I_0 + f(x_{j+1}, y_{j+1})I_1,$$

где значения интегралов в случае полиномиальной системы

$$I^{<-2>} = \int_{x_j}^{x_{j+1}} \omega_{j-2}^{<-2>}(x) dx = \frac{1}{17},$$

$$I^{<-1>} = \int_{x_j}^{x_{j+1}} \omega_{j-1}^{<-1>}(x) dx = \frac{-9}{17},$$

$$I_0 = \int_{x_j}^{x_{j+1}} \omega_{j,0}(x) dx = \frac{18h}{17},$$

$$I_1 = \int_{x_j}^{x_{j+1}} \omega_{j+1,0}(x) dx = \frac{6h}{17}.$$

Таким образом, в случае полиномиальной системы базисных функций

$$\varphi_1(x) = 1, \quad \varphi_2(x) = x, \quad \varphi_3(x) = x^2, \quad \varphi_4(x) = x^3,$$

получаем

$$y_{j+1} = y_j \left(\frac{9}{17} \right) - y_{j-1} \left(\frac{-9}{17} \right) - y_{j-2} \left(\frac{1}{17} \right) + f(x_j, y_j) \left(\frac{18h}{17} \right) + f(x_{j+1}, y_{j+1}) \left(\frac{6h}{17} \right).$$

Нетрудно получить оценку

$$|R| \leq Kh^5 \|y^V\|_{[x_{j-2}, x_{j+1}]}, K > 0.$$

Пример. Решаем задачу Коши

$$y' = f(x, y), \quad f(x, y) = -50y + 50 \cos(x), \quad y(0) = 0,$$

Нетрудно видеть, что эта задача имеет точное решение

$$y = \frac{2500}{2501} \cos(x) + \frac{50}{2501} \sin(x) - \frac{2500}{2501} e^{(-50x)}.$$

Конструкция распределения заданий sections предоставляет возможность выполнять каждому потоку свой структурированный блок.

В результате вычислений в параллельном режиме предложенными методами

```
#pragma omp parallel sections
{
    #pragma omp section
    {
        \\первый метод решения
    }
    #pragma omp section
    {
        \\ второй метод решения
    }
}
```

экономим время выполнения и проверяем результаты вычислений.

Литература

1. Бурова И.Г. Materialy IX mezinarodni vedecko-prakticka konference «Moderni vymozenosti vedy-2013». 27.01.13-05.02.2013. Praha. 2013. С. 3–6.
2. Бурова И.Г. О моделировании неполиномиальных интегро-дифференциальных приближений // Труды СПИИРАН. Вып. 4 (19). 2010. с.176–202.

О РАСПАРАЛЛЕЛИВАНИИ РЕШЕНИЯ ЗАДАЧИ ЭРМИТА-БИРКГОФА ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫМИ СПЛАЙНАМИ

Бурова И.Г.

*д. ф.-м. н., профессор; профессор кафедры вычислительной математики, профессор кафедры параллельных алгоритмов
математико-механического факультета СПбГУ,
BurovaIG@mail.ru.*

Родникова О.В.

*аспирантка кафедры вычислительной математики СПбГУ,
ilmarik.spb@gmail.com.*

Аннотация: Изучается решение обобщенной задачи Эрмита-Биркгофа с помощью интегродифференциальных сплайнов и исследуются свойства непрерывности построенных сплайнов. Обсуждается распараллеливание решения в OPEN MP.

Традиционно задача Эрмита-Биркгофа решается с помощью интерполяционных полиномов (см., напр. [1]), хотя в некоторых случаях получить решение с помощью интерполяционных полиномов невозможно. В работе [2] рассматривается решение задачи Эрмита-Биркгофа с помощью неполиномиальных минимальных сплайнов. В книге [3] и статье [6] предложены полиномиальные интегро-дифференциальные сплайны. Неполиномиальные интегро-дифференциальные сплайны рассмотрены в статье [4]. Здесь рассмотрим решение одной обобщенной задачи Эрмита-Биркгофа с помощью интегро-дифференциальных сплайнов. Исследуем полученное решение на непрерывность и предложим обобщение задачи, позволяющее построить решение более высокого класса гладкости.

Рассматриваем функцию $U(x)$, достаточное количество раз дифференцируемую на конечном отрезке $[a, b]$. Строим на отрезке $[a, b]$ упорядоченную по возрастанию сетку несовпадающих узлов с переменным шагом h_k :

$$\Omega: a = x_0 < x_1 < \dots < x_k < x_{k+1} < \dots < x_n = b.$$

Предполагаем, что определены значения производной $U'(x_k)$ в узлах x_k неравномерной сетки Ω и известны значения интегралов $\int_{x_k}^{x_{k+1}} U(x) dx$ ($k=0, \dots, n-1$). Требуется восстановить функцию $U(x)$ сплайном $\tilde{U}(x) = \bigcup_{k=0}^{n-1} \tilde{U}_k(x)$, составленным из звеньев вида

$$\tilde{U}_k(x) = U'(x_k)\omega_{k,1}(x) + U'(x_{k+1})\omega_{k+1,1}(x) + \int_{x_k}^{x_{k+1}} U(x)dx \omega_k^{<1>}(x), \quad (1)$$

$$x \in [x_k, x_{k+1}],$$

удовлетворяющих дифференциальным и интегральным условиям согласования:

$$1. \tilde{U}'_k(x_k) = U'(x_k), \tilde{U}'_k(x_{k+1}) = U'(x_{k+1}).$$

$$2. \int_{x_k}^{x_{k+1}} \tilde{U}_k(x)dx = \int_{x_k}^{x_{k+1}} U(x)dx.$$

Базисные функции $\omega_{k,1}(x)$, $\omega_{k+1,1}(x)$, $\omega_k^{<1>}(x)$ находим из условий:

$$1. U(x) - \tilde{U}_k(x) = 0, \text{ если } U(x) = 1, x, x^2 \text{ (случай 1).}$$

$$2. U(x) - \tilde{U}_k(x) = 0, \text{ если } U(x) = 1, \sin(x), \cos(x) \text{ (случай 2).}$$

Введем замену переменных $x = x_k + th_k$, $t \in [0, 1]$. После несложных преобразований для базисных функций получаем следующие выражения:

в первом случае:

$$\omega_{k,1}(t) = -\frac{h_k}{2} \left(t^2 - 2t + \frac{2}{3} \right), \quad \omega_{k+1,1}(t) = \frac{h_k}{2} \left(t^2 - \frac{1}{3} \right), \quad \omega_k^{<1>}(t) = \frac{1}{h_k};$$

во втором случае:

$$\omega_{k,1}(t) = \frac{1}{\Delta} (\sin(h_k) - \cos(h_k - th_k)), \quad \omega_{k+1,1}(t) = \frac{1}{\Delta} (-\sin(h_k) + h_k \cos(th_k)),$$

$$\omega_k^{<1>}(t) = \frac{1}{h_k}.$$

Справедливо следующее утверждение:

Лемма 1. Звенья сплайна $\tilde{U}_k(x)$ ($k=0, \dots, n-1$), полученные подстановкой базисных функций в выражение (1), удовлетворяют требуемым дифференциальным и интегральным условиям согласования:

$$1. \tilde{U}'_k(x_k) = U'(x_k), \tilde{U}'_k(x_{k+1}) = U'(x_{k+1}).$$

$$2. \int_{x_k}^{x_{k+1}} \tilde{U}_k(x)dx = \int_{x_k}^{x_{k+1}} U(x)dx.$$

Объединяем звенья $\tilde{U}_k(x)$ на всех отрезках $[x_k, x_{k+1}]$ ($k=0, \dots, n-1$). Получаем локальный интегро-дифференциальный сплайн на отрезке $[a, b]$:

$$\tilde{U}(x) = \bigcup_{k=0}^{n-1} \tilde{U}_k(x).$$

Построенный локальный сплайн $\tilde{U}(x)$ является разрывным в узлах, но значения его производной в узлах справа и слева совпадают.

Справедлива следующая теорема:

Теорема 1. Если функция $U(x) \in C^3[a, b]$, то справедлива оценка:

$$\|\tilde{U}(x) - U(x)\|_{L^2_{[a,b]}} \leq h_k^3 \|LU\|_{L^2_{[a,b]}}$$

где LU имеет вид

1. $LU = U'''(x)$ в случае 1.
2. $LU = U'''(x) + U'(x)$ в случае 2.

На основе полученных звеньев $\tilde{U}_k(x)$ строим глобальный сплайн класса $C^1[a, b]$.

Предполагаем, что заданы значения интегралов $\int_{x_k}^{x_{k+1}} U(x) dx$ ($k=0, \dots, n-1$) и значения производной в граничных точках $U'(x_0), U'(x_n)$. Требуется восстановить функцию $U(x)$ интегро-дифференциальным сплайном $\hat{U}(x) = \bigcup_{k=0}^{n-1} \hat{U}_k(x)$ класса $C^1[a, b]$, составленным из звеньев вида

$$\hat{U}_k(x) = \hat{U}'(x_k) \omega_{k,1}(x) + \hat{U}'(x_{k+1}) \omega_{k+1,1}(x) + \int_{x_k}^{x_{k+1}} U(x) dx \omega_k^{<1>}(x), \tag{2}$$

$$x \in [x_k, x_{k+1}],$$

удовлетворяющих интегральным условиям согласования

$$\int_{x_k}^{x_{k+1}} \hat{U}_k(x) dx = \int_{x_k}^{x_{k+1}} U(x) dx.$$

Для получения звеньев $\hat{U}_k(x)$ вычисляем неизвестные значения производной $U'(x)$ во внутренних узлах сетки Ω следующим образом.

Записываем условия стыковки звеньев $\hat{U}_k(x)$ в узлах сетки Ω :

$$\hat{U}_{k-1}(x_k) = \hat{U}_k(x_k), \text{ где } k = 1, \dots, n-1.$$

После несложных преобразований получаем в случае 1:

$$\begin{aligned} h_{k-1} \hat{U}'(x_{k-1}) + 2(h_{k-1} + h_k) \hat{U}'(x_k) + h_k \hat{U}'(x_{k+1}) = \\ = 6 \left(\frac{1}{h_k} \int_{x_k}^{x_{k+1}} U(x) dx - \frac{1}{h_{k-1}} \int_{x_{k-1}}^{x_k} U(x) dx \right). \end{aligned}$$

В случае 2:

$$\left(-\frac{1}{h_{k-1}} + \frac{1}{\sin(h_{k-1})}\right)\widehat{U}'(x_{k-1}) + \left(\frac{1}{h_{k-1}} + \frac{1}{h_k} - \frac{\cos(h_{k-1})}{\sin(h_{k-1})} - \frac{\cos(h_k)}{\sin(h_k)}\right)\widehat{U}'(x_k) - \left(\frac{1}{h_k} - \frac{1}{\sin(h_k)}\right)\widehat{U}'(x_{k+1}) = \frac{1}{h_k} \int_{x_k}^{x_{k+1}} U(x) dx - \frac{1}{h_{k-1}} \int_{x_{k-1}}^{x_k} U(x) dx.$$

Полученные трехдиагональные системы уравнений устанавливают связь между заданными значениями интегралов $\int_{x_k}^{x_{k+1}} U(x) dx$ ($k=0, \dots, n-1$) и известными значениями производной $U'(x_k)$ ($k=1, \dots, n-1$). Для замыкания систем используем известные значения производной в граничных точках $U'(x_0)$, $U'(x_n)$. Системы имеют диагональное преобладание. Для их решения применяем метод прогонки.

Справедливо следующее утверждение:

Лемма 2. Звенья сплайна $\widehat{U}_k(x)$ ($k=0, \dots, n-1$), полученные подстановкой значений производной $U'(x_k)$ ($k=1, \dots, n-1$) в выражение (2), удовлетворяют интегральным условиям согласования:

$$\int_{x_k}^{x_{k+1}} \widehat{U}_k(x) dx = \int_{x_k}^{x_{k+1}} U(x) dx.$$

Объединяем звенья $\widehat{U}_k(x)$ на всех отрезках $[x_k, x_{k+1}]$ ($k=0, \dots, n-1$). Получаем глобальный интегро-дифференциальный сплайн класса $C^1[a, b]$:

$$\widehat{U}(x) = \bigcup_{k=0}^{n-1} \widehat{U}_k(x).$$

Справедлива следующая теорема:

Теорема 2. Если функция $U(x) \in C^3[a, b]$, то справедлива оценка:

$$\max_{x \in [a, b]} |\widehat{U}(x) - U(x)| \leq Kh_k^3 \max_{x \in [a, b]} |LU|, \text{ где } K \text{ и } LU \text{ имеют вид}$$

1. $K = \frac{9}{24}$ и $LU = U'''(x)$ в случае 1.
2. $K = 0,414$ и $LU = U'''(x) + U'(x)$ в случае 2.

В системе Марле теоретические погрешности подтверждены вычислениями приближений для различных функций.

Рассмотрим теперь распараллеливание решения системы уравнений с трехдиагональной матрицей

$$a_k y_{k-1} + b_k y_k + c_k y_{k+1} = f_k, \quad k=1, 2, \dots, \quad N=2n, \quad y_0=0, \quad y_{2n}=0$$

на два процессора при $N=2n$ методом встречной прогонки.

Верхние индексы будем относить к номеру процессора.

На первом процессоре имеем

$$K_2^{(1)} = -\frac{c_1}{b_1}, \quad L_2^{(1)} = \frac{f_1}{b_1};$$

$$K_{i+1}^{(1)} = -\frac{c_i}{b_i + a_i K_i^{(1)}}, \quad L_{i+1}^{(1)} = -\frac{f_i - a_i L_i^{(1)}}{b_i + a_i K_i^{(1)}};$$

$$i=2, \dots, n.$$

На втором процессоре получаем

$$K_{N-1}^{(2)} = \frac{-a_N}{b_N}, \quad L_{N-1}^{(2)} = \frac{f_N}{b_N};$$

$$K_{N-1-i}^{(2)} = -\frac{-a_{N-i}}{c_{N-i} + a_i K_{N-i}^{(2)} + b_{N-i}}, \quad L_{N-i}^{(2)} = -\frac{f_{N-i} - c_{N-i} a_i L_{N-i}^{(1)}}{c_{N-i} + a_i K_{N-i}^{(2)} + b_{N-i}};$$

$$i=1, \dots, n-1.$$

Находим $y_n = (L_{N+1}^{(1)} + L_N^{(2)} K_{N+1}^{(1)}) / (1 - K_N^{(2)} K_{N+1}^{(1)})$.

Далее вычисляем y_i по формулам

$$y_{i-1} = K_i^{(1)} y_i + L_i^{(1)}, \quad i=n, \dots, 2,$$

$$y_{n+i+1} = K_{n+i}^{(2)} y_{n+i} + L_{n+i}^{(2)}, \quad i=0, \dots, n-1.$$

Применим полученные результаты для решения задачи Коши.

Найдем предварительно базисные функции $\omega_{k,1}(x)$, $\omega_{k+1,1}(x)$, $\omega_k^{<-1>}(x)$ для аппроксимации $\tilde{U}(x) = U'(x_k) \omega_{k,1}(x) + U'(x_{k+1}) \omega_{k+1,1}(x) + \int_{x_{k-1}}^{x_k} U(x) dx \omega_k^{<-1>}(x)$ из условия $\tilde{U}(x) = U(x)$, $U = 1, x, x^2$. Переходя на промежутке $[x_k, x_{k+1}]$ к переменной t по правилу $x = x_k + th$, $h = x_{k+1} - x_k$, получим

$$\omega_{k,1}(t) = -\frac{1}{6}h(-4 - 6t + 3t^2), \quad \omega_{k+1,1}(t) = \frac{1}{2}ht^2 - \frac{1}{6}h, \quad \omega_k^{<-1>}(x) = \frac{1}{h}.$$

Решаем задачу Коши

$$y' = f(x, y(x)), \quad y(x_0) = y_0, \quad x \in [x_0, X].$$

Пусть n — целое число. На промежутке $[x_0, X]$ построена сетка узлов

$$\{x_j\}: x_0 < \dots < x_j < x_{j+1} \leq \dots \leq x_n = X.$$

Заменяя в тождестве Ньютона-Лейбница

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} y'(x) dx$$

под знаком интеграла производную на $\tilde{U}(x) = U'(x_k)\omega_{k,1}(x) + U'(x_{k+1}) \cdot \omega_{k+1,1}(x) + \int_{x_{k-1}}^{x_k} U(x) dx \omega_k^{<1>}(x)$, и полагая $y'(x) = \tilde{U}(x)$, получаем аналогично [7] после вычисления интегралов и отбрасывания погрешности явный двухшаговый метод решения задачи Коши

$$y_{i+1} = y_i + (f'_x(x_i, y_i) + f'_y(x_i, y_i)f(x_i, y_i))h^2 + (y_i - y_{i-1}).$$

Пример. Задача Коши $y' = 2(y(x) - \sin(x)) + \cos(x)$, $y_0 = 0$ имеет решение $y = \sin(x)$. Решая эту задачу предложенным методом на промежутке $[0, 2]$ с шагом $h = 0,01$, получаем решение с абсолютной погрешностью 10^{-5} . Заметим, что построенный численный метод обладает неустойчивостью по отношению к ошибкам округления.

Литература

1. *Мысовских И.П.* Лекции по методам вычислений. — СПб.: Изд-во СПбГУ, 1998. — 472 с.
2. *Бурова И. Г, Тимофеев В.А.* Решение задачи Эрмита-Биркгофа с помощью минимальных неполиномиальных сплайнов. — СПб.: Вестник СПбГУ, сер 1, вып 3, 2006. — с. 50–51.
3. *Киреев В.И., Пантелеев А.В.* Численные методы в примерах и задачах. — М.: Высшая школа, 2008. — 480 с.
4. *Бурова И.Г.* О моделировании неполиномиальных интегро-дифференциальных приближений. — СПб.: Труды СПИИРАН, вып 4(19), 2011. — с. 176–208.
5. *Бурова И.Г., Демьянович Ю.К.* Минимальные сплайны и их приложения. — СПб.: Изд-во СПбГУ, 2010. — 364 с.
6. *Киреев В.И., Бирюкова Т.К.* Полиномиальные интегро-дифференциальные одномерные и двумерные сплайны. — Новосибирск.: Вычислительные технологии, 1998. — т.3, № 3, с. 19–34.
7. *Materialy IX mezinarodni vedecko-prakticka konference "Moderni vymozenosti vedy-2013"*. 27.01.13-05.02.2013. — Прага, 2013. — с. 3–6.

О ПОСТРОЕНИИ СРЕДНЕКВАДРАТИЧЕСКОГО ПРИБЛИЖЕНИЯ ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫМИ СПЛАЙНАМИ ПЯТОГО ПОРЯДКА ПЕРВОЙ ВЫСОТЫ

Бурова И.Г.

*профессор кафедры вычислительной математики СПбГУ,
BurovaIG@mail.ru*

Полуянов С.В.

*аспирант кафедры вычислительной математики СПбГУ,
sergeypoluyanov@gmail.com*

Аннотация: Рассматривается построение среднеквадратического приближения с помощью базисных сплайнов пятого порядка аппроксимации первой высоты. Приводятся результаты численных экспериментов по ускорению вычислений при распараллеливании алгоритма.

1. Построение базисных сплайнов

Рассмотрим промежуток $[a, b]$, где a и b — вещественные числа. Возьмём целое число n и построим равномерную сетку узлов $\{x_j\}$ с шагом $a = (b-a)/n$:

$$a = x_1 < \dots < x_{j-1} < x_j < x_{j+1} < \dots < x_n = b.$$

Пусть в узлах сетки $\{x_j\} \subset [a, b]$ заданы значения функции u , $u \in C^5[a, b]$, и ее первой производной, а также известны значения $\int_{x_k}^{x_{k+1}} u(t) dt$. Рассмотрим на каждом $[x_k, x_{k+1}]$ приближение для $u(x)$ в виде

$$\begin{aligned} \tilde{u}(x) = & u(x_k)\omega_{k,0}(x) + u(x_{k+1})\omega_{k+1,0}(x) + \\ & + u'(x_k)\omega_{k,1}(x) + u'(x_{k+1})\omega_{k+1,1}(x) + \int_{x_k}^{x_{k+1}} u(t) dt \omega_k^{<1>}(x), \end{aligned}$$

где $\omega_{k,0}(x)$, $\omega_{k+1,0}(x)$, $\omega_{k,1}(x)$, $\omega_{k+1,1}(x)$, $\omega_k^{<1>}$ определяем из условий

$$\tilde{u}(x) = u(x), \text{ при } u = 1, \varphi_i(x), i = 2, \dots, 5.$$

Предполагаем, что $1, \varphi_i(x)$, $i = 2, \dots, 5$ — чебышевская система на $[x_k, x_{k+1}]$, $\varphi_i \in C^5[x_k, x_{k+1}]$.

Пусть $h = x_{k+1} - x_k$. Для случая $\varphi_i = x^{i-1}$, $i=2, \dots, 5$ переходя к переменной t по правилу $x = kh + th$ получаем следующие формулы базисных сплайнов:

$$\omega_{k,0}(t) = \begin{cases} -(5t+1)(3t-1)(t-1)^2, & t \in [0, 1], \\ -(3t+1)(5t-1)(1+t)^2, & t \in [-1, 0], \\ 0, & t \notin [-1, 1], \end{cases}$$

$$\omega_{k,1}(t) = \begin{cases} -\frac{1}{2}th(5t-2)(t-1)^2, & t \in [0, 1], \\ \frac{1}{2}th(2+5t)(1+t)^2, & t \in [-1, 0], \\ 0, & t \notin [-1, 1], \end{cases}$$

$$\omega_k^{<1>}(t) = \begin{cases} \frac{30t^2}{h}(t-1)^2, & t \in [0, 1], \\ 0, & t \notin [-1, 1], \end{cases}$$

заметим, что

$$\omega_{k+1,0}(t) = -(5t-6)(3t-2)t^2, \quad t \in [0, 1],$$

$$\omega_{k+1,1}(t) = \frac{1}{2}(5t-3)(t-1)t^2.$$

Нетрудно видеть, что $\omega_{k,0}, \omega_{k,1}, \omega_k^{<1>} \in C^1(R^1)$.

2. Формирование и хранение элементов матрицы Грама

Элементы матрицы Грама имеют вид

$$(\omega_{i,k}, \omega_{j,s}) = \int_{x_{j-1}}^{x_{j+1}} \omega_{i,k}(x) \omega_{j,s}(x) dx, \quad (\omega_{i,k}, \omega_j^{<1>}) = \int_{x_{j-1}}^{x_{j+1}} \omega_{i,k}(x) \omega_j^{<1>}(x) dx,$$

$$k, s = 0, 1, i, j = 1, \dots, n.$$

Матрица Грама M системы уравнений $MC = F$ может быть представлена в блочном виде

$$M = \begin{pmatrix} M_1 & M_2 & M_5 \\ M_3 & M_4 & M_6 \\ M_7 & M_8 & M_9 \end{pmatrix},$$

причем $M_3 = (M_2)^T$, $M_7 = (M_5)^T$, $M_8 = (M_6)^T$, а $M_1, M_2, M_5, M_4, M_6, M_9$ — ленточные матрицы. Носители базисных функций $\omega_{j,0}, \omega_{j,1}$, содержат два сеточных интервала, а $\omega_j^{<1>}$ — один сеточный интервал, поэтому в матрице M мало отличных от нуля элементов. Результаты вычислений показывают, что различных между собой элементов 12. Таким образом нет необходимости хранить всю матрицу, а ограничиваемся хранением 12 элементов.

Приведем вид матриц M_i при $n = 3$

$$M_1 = \begin{pmatrix} (8/35)h & -(1/70)h & 0 & 0 \\ -(1/70)h & (16/35)h & -(1/70)h & 0 \\ 0 & -(1/70)h & (16/35)h & -(1/70)h \\ 0 & 0 & -(1/70)h & (8/35)h \end{pmatrix},$$

$$M_2 = \begin{pmatrix} (1/60)h^2 & (1/210)h^2 & 0 & 0 \\ -(1/210)h^2 & 0 & (1/210)h^2 & 0 \\ 0 & -(1/210)h^2 & 0 & (1/210)h^2 \\ 0 & 0 & -(1/210)h^2 & -(1/60)h^2 \end{pmatrix},$$

$$M_4 = \begin{pmatrix} (1/630)h^3 & (1/1260)h^3 & 0 & 0 \\ (1/1260)h^3 & (1/315)h^3 & (1/1260)h^3 & 0 \\ 0 & (1/1260)h^3 & (1/315)h^3 & (1/1260)h^3 \\ 0 & 0 & (1/1260)h^3 & (1/630)h^3 \end{pmatrix},$$

$$M_5 = \begin{pmatrix} -3/14 & 0 & 0 \\ -3/14 & -3/14 & 0 \\ 0 & -3/14 & -3/14 \\ 0 & 0 & -3/14 \end{pmatrix},$$

$$M_6 = \begin{pmatrix} -(1/84)h & 0 & 0 \\ (1/84)h & -(1/84)h & 0 \\ 0 & (1/84)h & -(1/84)h \\ 0 & 0 & (1/84)h \end{pmatrix},$$

$$M_9 = \begin{pmatrix} (10/7)h & 0 & 0 \\ 0 & (10/7)h & 0 \\ 0 & 0 & (10/7)h \end{pmatrix}.$$

2. Решение системы линейных алгебраических уравнений

Разработан алгоритм для преобразования матрицы M системы уравнений к ленточному виду с помощью перестановки строки столбцов.

Вычисление элементов вектора $F_i, i=0, 1, 2$ производится параллельно динамическим распределением блоками по процессорам.

Для решения системы уравнений применяется параллельный вариант встречной прогонки. Среднеквадратическое приближение функции $u(x)$

$$\tilde{u}(x) = c_{k,0}\omega_{k,0}(x) + c_{k+1}\omega_{k+1,0}(x) + c_{k,1}\omega_{k,1}(x) + c_{k+1}\omega_{k+1,1}(x) + c_{k,2}\omega_k^{<1>}(x),$$

и среднеквадратическое приближение производной функции $u(x)$

$$\tilde{u}'(x) = c_{k,0}\omega'_{k,0}(x) + c_{k+1}\omega'_{k+1,0}(x) + c_{k,1}\omega'_{k,1}(x) + c_{k+1}\omega'_{k+1,1}(x) + c_{k,2}\omega_k^{<1>'}(x),$$

строим параллельно на двух процессорах при $x \in [x_1, x_{N-1}]$ и $x \in [x_N, x_n]$ (предполагаем, что $2N = n - 1$).

Литература

1. Бурова И.Г., Демьянович Ю.К. Теория минимальных сплайнов. 357 с., Изд-во СПбГУ, 1999.
2. Бурова И.Г. Приближения неполиномиальными сплайнами максимального дефекта, 37 с., Изд-во СПбГУ, 2007.
3. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры, 736 с., Изд-во «Лань» 1960.
4. Самарский А.А. Введение в численные методы, 271 с., Изд-во «Наука» 1982.
5. Канг Су Гэтлин, Пит Айсенсие OpenMPиC++, MSDN Magazine, Русская Редакция, Октябрь 2005г. <http://www.microsoft.com/Rus/Msdn/Magazine/2005/10/OpenMP.mspх>
6. Barbara Chapman, Gabriel Jost, Ruud van der Pas Using OpenMP. Portable Shared Memory Parallel Programming, 353 с., The MIT Press, 2008.
7. Березин И.С., Жидков Н.П. Методы вычислений. Т.1. М. 1962.

Нелинейные динамические системы: модели, управление, численный анализ



**Клейменов
Анатолий Федорович**

д.ф.-м.н., в.н.с. отдела динамических систем ИММ им. Н.Н. Красовского УрО РАН
профессор кафедры механики и мат. моделирования ИМКН УрФУ



**Кандоба
Игорь Николаевич**

к.ф.-м.н., с.н.с. отдела прикладных проблем управления ИММ им. Н.Н. Красовского УрО РАН
доцент кафедры вычислительной математики ИМКН УрФУ

О ЗАДАЧЕ ОПТИМАЛЬНОГО ВЫВЕДЕНИЯ РАКЕТЫ-НОСИТЕЛЯ НА ОКОЛОЗЕМНУЮ ОРБИТУ¹

Думшева Т.Д.

*с. н. с. отдела прикладных проблем управления ИММ УрО РАН,
did@imm.uran.ru,*

Кандоба И.Н.

*с. н. с. отдела прикладных проблем управления ИММ УрО РАН,
kandoba@imm.uran.ru,*

Костоусова Е.К.

в. н. с. отдела оптимального управления ИММ УрО РАН, kek@imm.uran.ru,

Ложников А.Б.

*н. с. отдела дифференциальных уравнений ИММ УрО РАН,
ablozhnikov@yandex.ru,*

Починский В.И.

в. н. с. ФГУП НПО автоматики им. акад. Н.А. Семихатова

Аннотация: Рассматриваются задачи управления, возникающие при выведении ракеты-носителя на заданную околоземную эллиптическую орбиту. Движение ракеты-носителя описывается нелинейной динамической системой. В ряде практически важных задач к искомому управлению предъявляются дополнительные требования. Первое заключается в максимизации выводимой на орбиту массы полезной нагрузки, второе заключается в обеспечении требования падения отделяемых частей РН в заданные районы, третье диктуется необходимостью обеспечения успешного возвращения космического аппарата на Землю с любой точки траектории движения ракеты-носителя. Второе и третье условия приводят к возникновению ограничений на текущее фазовое состояние динамической системы.

Введение

На протяжении последних примерно десяти лет в Институте математики и механики им. Н.Н.Красовского УрО РАН (ИММ) совместно с НПО автоматики им. акад. Н.А.Семихатова (г. Екатеринбург) (НПОА) активно иссле-

¹ Работа выполнена в рамках программ фундаментальных исследований Президиума РАН «Динамические системы и теория управления» при поддержке УрО РАН (проект 12-П-1-1022), «Информационные, управляющие и интеллектуальные технологии и системы» (проект 12-П-1-1023) и интеграционного проекта УрО и СО РАН (проект 12-С-1-1017).

дуются задачи оптимального выведения ракеты-носителя (РН) на заданную околоземную эллиптическую орбиту. Математическая модель управляемого движения РН описывается нелинейной динамической системой, где в качестве управления используются угловые скорости изменения углов тангажа и рысканья, определяющих угловую ориентацию строительной оси РН. Основное внимание в этих исследованиях уделяется разработке методов построения программного управления, обеспечивающего вывод РН на заданную орбиту и удовлетворяющего ряду дополнительных требований. В настоящее время у разработчиков ракетно-космической техники наибольший интерес вызывают проблемы разработки такого управления, которое позволяет вывести на орбиту полезную нагрузку максимальной массы и обеспечивает выполнение некоторых фазовых ограничений на текущее состояние нелинейной динамической системы. Рассматриваемые фазовые ограничения диктуются необходимостью обеспечения падения отделяемых частей РН в заданные районы, а также учета возможности успешного возвращения космического аппарата на Землю с любой точки траектории движения РН. Особую актуальность последнее условие приобретает при подготовке РН к пилотируемому пуску.

1. Математические модели управляемого движения ракеты-носителя и баллистического спуска его отделяемых частей

Задача вывода РН на заданную орбиту решается на базе математических моделей управляемого движения РН на всей траектории полета [1] и движения его отделяемых частей (ОЧ) при баллистическом спуске [2].

Уравнения движения центра масс РН в инерциальной стартовой системе координат от момента старта t_s до момента t_f выхода на заданную орбиту могут быть записаны в следующем компактном виде:

$$\dot{x} = v, \dot{v} = W(t, x, v, m, \vartheta, \psi), \dot{m} = -\mu t, \dot{\vartheta} = u_1, \dot{\psi} = u_2, t \in [t_s, t_f] \quad (1)$$

где $x \in R^3$, $v \in R^3$ — координаты и компоненты скорости центра масс РН; m — масса РН; $\mu t > 0$ — кусочно-постоянная функция расхода массы с переключениями в моменты сброса отделяемых частей РН; $W(t, x, v, m, \vartheta, \psi)$ — ускорение, задаваемое суммой составляющих, определяемых реактивными, аэродинамическими и гравитационными силами:

$$W(t, x, v, m, \vartheta, \psi) = W_R(t, x, m, \vartheta, \psi) + W_A(t, x, v, m, \vartheta, \psi) + g_x,$$

$$W_R(t, x, m, \vartheta, \psi) = \frac{P(t, x)}{m(t)} (\cos(\vartheta) \cos(\psi), \sin(\vartheta) \cos(\psi), -\sin(\psi))^T;$$

$P(t, x)$ — сила тяги двигательной установки РН; ϑ, ψ — углы тангажа и рысканья РН, определяющие угловую ориентацию строительной оси РН;

t_s — момент начала движения РН; t_f — момент выхода РН на заданную орбиту.

Орбита, на которую выводится РН, задается следующими пятью параметрами (параметры оскулирующей орбиты [1]): наклонение плоскости орбиты i , долгота восходящего узла Ω , минимальная высота орбиты h_{\min} , максимальная высота орбиты h_{\max} , аргумент перигея ω .

Для системы (1) задаются начальные условия:

$$x(t_s) = x_s, \quad v(t_s) = v_s, \quad m(t_s) = m_s, \quad \vartheta(t_s) = \vartheta_s, \quad \psi(t_s) = \psi_s. \quad (2)$$

Момент времени выхода РН на орбиту не фиксирован и полагается равным моменту времени выключения двигателя последней ступени РН.

Скорости и изменения углов и рассматриваются как управляющие воздействия, ограниченные по величине неравенствами

$$|u_1(t)| \leq u_1^{\max}, \quad |u_2(t)| \leq u_2^{\max}, \quad t \in [t_s, t_f]. \quad (1)$$

Математическая модель баллистического спуска отделяемой части РН (первая, вторая ступени для РН «Союз-2»), хвостовой отсек и головной обтекатель последней ступени, космический аппарат в случае его аварийного спуска) описывает траекторию движения центра масс ОЧ в неоднородном центральном поле тяготения от момента $\hat{t}_s \in [t_s, t_f]$ отделения ОЧ от РН до момента \hat{t}_f соприкосновения ОЧ с земной поверхностью. Параметрами модели являются масса, коэффициент лобового сопротивления и площадь мидела ОЧ, а также — начальные условия в момент отделения ОЧ от РН: пространственные координаты центра масс ОЧ $\hat{x}(\hat{t}_s)$, направление и величина вектора скорости $\hat{v}(\hat{t}_s)$ ОЧ. Здесь в обозначениях величин, относящихся к ОЧ, символ «крышка» использован для того, чтобы различать соотношения, описывающие движение РН на всей траектории и движение ОЧ при ее баллистическом спуске. Движение центра масс ОЧ [2] описывается системой дифференциальных уравнений

$$\dot{\hat{x}} = \hat{v}, \quad \dot{\hat{v}} = W_A(t, \hat{x}, \hat{v}) + g(\hat{x}), \quad t \in [\hat{t}_s, \hat{t}_f], \quad (4)$$

где $W_A(t, \hat{x}, \hat{v})$ — аэродинамическое ускорение.

Начальные условия для системы (4) определяются значениями векторов x и v для РН в момент времени \hat{t}_s :

$$\hat{x}(\hat{t}_s) = x(\hat{t}_s), \quad \hat{v}(\hat{t}_s) = v(\hat{t}_s). \quad (5)$$

Разрешенный район падения ОЧ считается круговым и задается тремя параметрами: φ_f , λ_f , ε_{LB} — широта и долгота центра района и его радиус, т. е. предельно допустимое отклонение точки $\hat{x}(\hat{t}_f)$ от центра, заданное

в метрах. В терминах φ_f , λ_f , и $\hat{x}(t_f)$ можно вычислить «радиус промаха» r для точки $\hat{x}(t_f)$. Тогда ограничение на точку падения ОЧ может быть записано в виде:

$$r(\hat{t}_f, \hat{x}(t_f)) \leq \varepsilon_{\text{ЛВ}}. \quad (6)$$

В случае возникновения непредвиденной аварийной ситуации космический аппарат (КА) должен быть благополучно возвращен на Землю. При движении КА в атмосфере (которое описывается уравнениями типа (4)) возникает перегрузка $n = n(\hat{x}, \hat{v}) = \|W_A\|/g_0$, которая показывает во сколько раз аэродинамическое ускорение больше ускорения силы тяжести g_0 на поверхности Земли.

Для перегрузки задается ее предельное значение \bar{n}_{max} , превышение которого недопустимо. Для обеспечения успешного возвращения КА на Землю в случае возникновения непредвиденной аварийной ситуации требуется, чтобы максимально возможные перегрузки не превышали значения \bar{n}_{max} . С учетом того, что траектория аварийного спуска КА $\hat{x}(t) = \hat{x}(t; \hat{t}_s)$, $\hat{v}(t) = \hat{v}(t; \hat{t}_s)$ зависит от момента $\hat{t}_s \in [t_s, t_f]$ начала спуска, ограничение на перегрузки может быть записано в виде

$$\max_{\hat{t}_s \in [t_s, t_f]} n_{\text{max}}(\hat{x}(\hat{t}_s), \hat{v}(\hat{t}_s)) \leq \bar{n}_{\text{max}}, \quad (7)$$

$$n_{\text{max}}(\hat{x}(\hat{t}_s), \hat{v}(\hat{t}_s)) = \max_{t \in [t_s, t_f]} n(\hat{x}(t; \hat{t}_s), \hat{v}(t; \hat{t}_s)). \quad (8)$$

2. Постановки задач оптимального выведения ракеты-носителя. Методы решения.

Задача максимизации выводимой ракетой-носителем на заданную орбиту массы полезной нагрузки исследовалась, в частности, в работах [3–6]. Известно, что при заданной положительной функции $\mu(t) > 0$ эта задача равносильна задаче минимизации значения момента времени t_f выхода РН на орбиту — задаче оптимального быстрого действия с терминальными ограничениями:

Задача 1. Для управляемой системы (1) с заданными начальными условиями (2) найти программное управление $u(\cdot) = (u_1(\cdot), u_2(\cdot))$, минимизирующее значение функционала $J[u(\cdot)] = t_f$. При этом должны быть выполнены ограничения (3) и

$$\begin{aligned} |i - \bar{i}| &\leq \bar{\Delta}_i, \quad |\Omega - \bar{\Omega}| \leq \bar{\Delta}_\Omega, \quad |h_{\text{min}} - \bar{h}_{\text{min}}| \leq \bar{\Delta}_{h_{\text{min}}}, \\ |h_{\text{max}} - \bar{h}_{\text{max}}| &\leq \bar{\Delta}_{h_{\text{max}}}, \quad |\omega - \bar{\omega}| \leq \bar{\Delta}_\omega, \end{aligned} \quad (9)$$

где \bar{i} , $\bar{\Omega}$, \bar{h}_{\min} , \bar{h}_{\max} , $\bar{\omega}$ — заданные значения параметров эллиптической орбиты выведения, а $\bar{\Delta}_i$, Δ_Ω , $\Delta_{h_{\min}}$, $\Delta_{h_{\max}}$, Δ_ω — допустимые отклонения от этих параметров.

Здесь следует отметить, что в задаче 1 рассматривается безатмосферный участок траектории последней ступени РН после сброса всех его отделяемых частей (t_s — момент времени начала работы последней ступени РН после сброса всех ОЧ).

В [3] предложен конструктивный метод численного решения задачи 1. Основная идея этого метода заключается в декомпозиции движения РН на этом участке на три составляющих — боковое, вертикальное и горизонтальное движения РН. Для каждого из этих типов движения РН решается соответствующая специальная задача оптимального управления. Синтез решений этих задач позволяет построить квазиоптимальное по быстродействию управление (базовое управление u^{base}), обеспечивающее вывод РН на заданную орбиту. Такой метод во многом базируется на инженерном представлении о структуре оптимального в задаче 1 управления. Тем не менее, несмотря на отсутствие полного строгого с математической точки зрения обоснования этого алгоритма, результаты численного моделирования с использованием реальных данных подтверждают его эффективность.

В [4] предложен алгоритм построения допустимого управления на атмосферном участке полета с учетом ограничений (6).

В [5,6] для задачи 1 разработаны другие подходы к построению допустимого управления, обеспечивающего выведение РН на заданную орбиту к фиксированному моменту времени t_f . В частности, допустимое управление в задаче 1 может быть построено как решение некоторой вспомогательной нелинейной экстремальной задачи [5,6]. Для решения такой задачи используется итерационная процедура, основанная на методе сопряженных градиентов. Попытки улучшить базовое управление с помощью этого подхода, а также с помощью другого подхода, описанного в [5], дали несущественный выигрыш по времени выведения РН на орбиту. При этом время вычислений оказывается существенно больше, чем при реализации метода из [5].

С учетом фазовых ограничений (6) и (7) на траекторию движения РН особую актуальность приобретают следующие задачи.

Задача 2. Для управляемой системы (1) с заданными начальными условиями (2) найти программное управление $u(\cdot) = (u_1(\cdot), u_2(\cdot))$, минимизирующее значение функционала $J[u(\cdot)] = t_f$. При этом должны быть выполнены ограничения (3), (6) и (9).

Задача 3. Для управляемой системы (1) с заданными начальными условиями (2) найти программное управление $u(\cdot) = (u_1(\cdot), u_2(\cdot))$, минимизиру-

ющее значение функционала $J[u(\cdot)] = t_f$. При этом должны быть выполнены ограничения (3), (6) (7) и (9).

При этом в задачах 2, 3 (в отличие от задачи 1) рассматривается весь промежуток управляемого движения РН, начиная от момента его старта.

Для построения допустимого в задаче 2 управления разработано несколько методов, основанных на использовании базового управления u^{base} . Для этого разработана модификация алгоритма [5] построения управления u^{base} на промежутке траектории движения РН, включающем атмосферный участок полета РН и моменты времени сброса всех его ОЧ.

Допустимое в задаче 2 управление, выводящее РН на орбиту и обеспечивающее падение его ОЧ в заданные районы, предлагается строить, используя несколько управлений u^{base} . Эти базовые управления последовательно строятся на нескольких интервалах времени. Начальные точки этих интервалов соответствуют либо моменту времени достижения РН априорно заданной высоты (высота конца ветра), либо моментам времени сброса отделяемых частей РН. Для каждого из таких моментов времени строится выводящее РН на орбиту управление u^{base} , определяемое состоянием системы (1) в соответствующий момент времени. Затем на этих интервалах времени последовательно (от интервала к интервалу) корректируются соответствующие управления u^{base} так, чтобы действующее на всем промежутке управление, образованное такими «скорректированными» базовыми управлениями и «неприкосновенным» базовым управлением на последнем интервале, являлось допустимым в задаче 2. Предложено несколько процедур «коррекции» базовых управлений.

В задаче 3 проведено исследование свойств функции максимальной перегрузки n_{max} . Были определены более конструктивные, чем координаты векторов $\hat{x}(\hat{t}_s)$, $\hat{v}(\hat{t}_s)$ (аргументы функции (8)), основные факторы, оказывающие доминирующее влияние на значения функции максимальной перегрузки, которые могут быть использованы в качестве ресурсов, за счет которых возможно удовлетворить ограничениям (7). Такими факторами являются: $H(\hat{x}(\hat{t}_s))$ — высота, на которой находится КА в момент начала аварийного спуска; $\|\hat{v}(\hat{t}_s)\|$ — модуль скорости КА в этот момент; α — угол между векторами $\hat{x}(\hat{t}_s)$ и $\hat{v}(\hat{t}_s)$. Предложены подходы к численному решению задачи 3, основанные на решении вспомогательных задач оптимального управления на фиксированном промежутке времени и использовании идеи метода штрафов. При этом минимизируемый функционал во вспомогательной задаче отвечает за выполнение ограничений на параметры орбиты, а фазовые ограничения, связанные с перегрузками, учитываются путем введения тех или иных слагаемых со штрафами. В частности, для задачи, где возможность аварийного спуска рассматривается только для не очень больших высот, предложено произвести неявный учет фазовых ограничений (7) путем введения в минимизируемый функционал слагаемого, зависящего от угла α , — интегрального слагаемого с подынтегральной функцией $\|x\|^{-2}\|v\|^{-2}(x^T v)^2$.

Заключение

В ИММ и НПОА усовершенствованы существующие и разработаны новые эффективные алгоритмы численного решения задач нелинейной динамики. Эти алгоритмы были апробированы в рамках широкомасштабного вычислительного эксперимента на многопроцессорной вычислительной системе с использованием реальных данных. Результаты численного моделирования свидетельствуют о достаточной эффективности разработанных методов.

Литература

1. *Охоцимский Д. Е., Сихарулидзе Ю.Г.* Основы механики космического полета. М.: Наука, 1990. 448 с.
2. *Охоцимский Д.Е., Голубев Ю.Ф., Сихарулидзе Ю.Г.* Алгоритмы управления космическим аппаратом при входе в атмосферу. М.: Наука, 1975. 399 с.
3. *Мазгалин Д.В.* Построение способа управления ракетой-носителем при использовании в качестве управления программных угловых скоростей разворотов // Информационно-управляющие системы. 2010. №3 (46). С.21–29.
4. *Мазгалин Д.В., Починский В.И.* Метод определения азимута пуска и программы угла тангажа на атмосферном активном участке полета РН // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление и радиоэлектроника». 2010. Вып.12. №22 (198). С. 47–50.
5. *Думиева Т.Д., Костоусов В.Б., Костоусова Е.К., Починский В.И.* Исследование задачи оптимального вывода полезной нагрузки на заданную эллиптическую орбиту// Труды Института математики и механики. УрО РАН. 2010. Т.16. №5. С.57–65.
6. *Костоусова Е.К., Починский В.И.* О задачах выведения ракеты-носителя на заданные эллиптические орбиты// Труды Института математики и механики. УрО РАН. 2011. Т.17. №3. С.201–216.

АЛГОРИТМЫ ПОСТРОЕНИЯ ОПТИМАЛЬНЫХ ТРАЕКТОРИЙ ПО АППРОКСИМИРУЮЩИМ УПРАВЛЕНИЯМ В ЗАДАЧАХ НА БЕСКОНЕЧНОМ ПРОМЕЖУТКЕ ВРЕМЕНИ¹

Тарасьев А.М.

*д. ф.-м. н., зав. сектором отдела динамических систем
ИММ УрО РАН им. Н.Н. Красовского, tam@imm.uran.ru*

Усова А.А.

*к. ф.-м. н., м. н. с. отдела динамических систем
ИММ УрО РАН им. Н.Н. Красовского, anastasy.ousova@gmail.com*

Аннотация: Статья посвящена задаче оптимального управления, построенной по модели оптимизации продуктивности природных ресурсов. Анализ проводится в рамках принципа максимума Понтрягина для задач с бесконечным горизонтом. Исследуется поведение динамической системы при различных режимах управления, осуществляется поиск стационарных уровней гамильтоновой динамики. Предлагается конструкция аппроксимирующего управления ступенчатого вида, построенного на основе условия ограниченности природных ресурсов.

Введение

Статья посвящена задаче увеличения продуктивности природных ресурсов за счет сбалансированной инвестиционной политики, направленной на дематериализацию экономики. Постановка задачи осуществлена на основе классических подходов, используемых в моделях экономического роста ([2], [6]), а также заимствует методы и подходы, предложенные в работе [7].

В модель включен механизм ценообразования, который обеспечивает рост стоимости материалов при уменьшении их запасов, что может трактоваться как естественный инструмент штрафов за нарушение фазовых ограничений в задачах оптимального управления. Дорожающие ресурсы отрицательно влияют на потребительский индекс, по которому оценивается качество процесса управления [8].

Актуальность подобных моделей значительна, поскольку опираясь на статистику [4] можно отметить стремительно растущие темпы потребления природных ресурсов.

¹ Работа поддержана грантами РФФИ (11-01-00427-а, 12-01-00024-а, 12-01-31300), Программой ведущих научных школ (НШ-64508.2010.1), Программой президиума РАН (12-П-1-1002, 12-П-1-1012, 12-П-1-1038, 12-П-7-1001), Проект 13-1-НП-253 УрО РАН и Международным Институтом Прикладного Системного Анализа, ИААСА (NSFC-ИАСА).

Задача состоит в нахождении таких оптимальных инвестиционных стратегий в увеличение продуктивности ресурсов, при которых интегральный индекс потребления логарифмического типа принимает максимальное значение вдоль траекторий динамической системы. Исследование задачи осуществляется в рамках принципа максимума Понтрягина ([1], [5]). Формулируются условия, при которых существует стационарная точка гамильтоновой динамики. Опираясь на свойства системы при постоянных управлениях, строится аппроксимирующее решение и сравнивается со стабилизированными траекториями, построенными по алгоритму, описанному в работе [7].

Описание модели и постановка задачи

Основными переменными модели служат внутренний валовой продукт (ВВП) y , объемы потребления ресурсов m и общий расход ресурсов к моменту времени t

$$M = M(t) = \int_0^t m(\tau) d\tau, \quad M_0 = \int_0^{+\infty} m(t) dt. \quad (1), (2)$$

Начальный уровень потребления m_0 и общий запас ресурсов M_0 известны. Продуктивность ресурсов $z(t)$ в момент времени t задается равенством

$$z(t) = z(t)/m(t) \quad (3)$$

В силу ограниченности природных ресурсов логично предполагать, что при уменьшении запасов материалов цены $p(t)$ на них будут расти. Указанная взаимосвязь описывается пропорцией:

$$p(t)/p_0 = M_0/(M_0 - M(t)) \quad (4)$$

В силу замкнутости экономической системы можно выписать следующее **балансовое соотношение** в относительных величинах

$$c(t)/y(t) = 1 - p(t)m(t)/y(t) - u(t), \quad (5)$$

которое говорит о том, что текущий объем ВВП $y(t)$ расходуется на потребление $c(t)$, закупку материалов по цене $p(t)$ в объеме $m(t)$ и инвестиции $u(t)$ в увеличение продуктивности $z(t)$ природных ресурсов. В виду соотношения (5) можно указать такую положительную величину \bar{u} , которая ограничивает уровни инвестиций сверху: $0 \leq u(t) \leq \bar{u} < 1$.

В модели используется **производственная функция** Кобба-Дугласа

$$y(t) = ae^{bt}m^{\alpha}(t), \quad a > 0, \quad b \geq 0, \quad 0 \leq \alpha < 1, \quad (6)$$

где параметр a — коэффициент масштаба, символ b определяет процесс роста выпуска $y(t)$ за счет развития производственных факторов (здесь потребление ресурсов $m(t)$), величина α — коэффициент эластичности.

Замечание. *Параметры модели удовлетворяют неравенству $m_0/M_0 < b/\alpha$, которое выполняется в статистических данных для большинства стран.*

Задача оптимального управления

Предполагается, что относительный темп изменения продуктивности $z(t)$ ресурсов пропорционален доле ВВП $u(t)$, инвестируемой в технологические разработки и направленной на повышение продуктивности ресурсов (см. [2]):

$$\dot{z}(t) = \beta u(t) z(t), \quad \beta \geq 0, \quad (7)$$

где параметр β определяет эффективность инвестиционного процесса $u(t)$. Из равенства (7), в силу соотношений (3) и (6), определяется закон изменения потребления природных ресурсов

$$\dot{m}(t) = (b - \beta u(t)) m(t) / (1 - \alpha). \quad (8)$$

Для приведения задачи оптимального управления к классическому виду вводятся фазовые переменные

$$x_1(t) = m(t) / (M_0 - M(t)), \quad x_2(t) = p(t) m(t) / y(t). \quad (9)$$

Отметим, что введение таких переменных необходимо в силу метода разрешения особенностей для величин, принимающих бесконечно большие (малые) значения в системах дифференциальных уравнений [3]. Для получения динамики введенных переменных $x(\cdot) = (x_1(\cdot), x_2(\cdot))$ используются соотношения (1), (3), (6) – (8).

$$\begin{aligned} \dot{x}_1(t) &= f_1(x(t), u(t)) = x_1(t) \left(x_1(t) + \frac{b - \beta u(t)}{1 - \alpha} \right), \quad x_1^0 = \frac{m_0}{M_0} \\ \dot{x}_2(t) &= f_2(x(t), u(t)) = x_2(t) (x_1(t) - \beta u(t)), \quad x_2^0 = \frac{p_0 m_0^\alpha}{a} \end{aligned} \quad (10)$$

Управление $u(t)$ в уравнениях (10) удовлетворяет ограничениям

$$0 \leq u(t) \leq \bar{u} < 1. \quad (7)$$

В переменных $x(t) = (x_1(t), x_2(t))$ логарифмический индекс потребления (6) записывается следующим образом:

$$\ln c(t) = \ln \frac{x_1(t)}{x_2(t)} \ln(1 - u(t) - x_2(t)) + \ln(p_0 M_0) = \omega(x, y) + \ln(p_0 M_0).$$

Функционал качества процесса управления определяется как интегральный логарифмический индекс потребления, дисконтированный на бесконечном промежутке времени (ρ — дисконтирующий множитель).

$$J(x(\cdot), u(\cdot)) = \int_0^{+\infty} e^{-\rho t} \omega(x_t, u_t) dt. \tag{12}$$

Задача оптимального управления состоит в максимизации функционала качества (12) вдоль траекторий (x_t, u_t) динамической системы (10), удовлетворяющей начальным условиям $x^0 = (x_1^0, x_2^0)$ и ограничениям (11) по переменной управления $u(t)$.

Исследование задачи

Анализ задачи оптимального управления проводится в рамках принципа максимума Понтрягина [5] при наличии особенностей бесконечного промежутка времени [1].

Гамильтониан и области его определения

Стационарная гамильтонова функция задачи управления имеет вид

$$\hat{H}(x, \psi, u) = \omega(x(t), u(t)) + \langle \psi, f(x, u) \rangle, \quad f(x, u) = \begin{pmatrix} f_1(x, u) \\ f_2(x, u) \end{pmatrix}, \tag{13}$$

где $\psi = (\psi_1, \psi_2)$ — вектор сопряженных переменных, $f(x, u)$ — вектор-функция, определяющая правые части динамической системы (10).

Вычисляя максимум функции $\hat{H}(x, \psi, u)$ (14) по переменной u , находим структуру управления u^0 (см. таблицу 1).

Управление, u^0	Ограничение
0	$x \in D_1 = \{x : v(x_3, x_4) \geq 1 - x_2\}$
$1 - x_2 - v(x_3, x_4)$	$x \in D_2 = \{x : 1 - \bar{u} - x_2 \leq v(x_3, x_4) \leq 1 - x_2\}$
\bar{v}	$x \in D_3 = \{x : v(x_3, x_4) \leq 1 - \bar{u} - x_2\}$

Таблица 1. Структура управления

Здесь введены сопряженные переменные $x_3 = \psi_1 x_1$, $x_4 = \psi_2 x_2$, функция $v(x_3, x_4) = -\frac{(1-\alpha)}{\beta} \frac{1}{x_3 + (1-\alpha)x_4}$, а вектор (x_1, x_2, x_3, x_4) обозначен символом x .

Стационарная точка и область ее существования

Анализируя гамильтонову динамику, полученную из принципа максимума Понтрягина [1], [5], исследуется вопрос существования стационарных точек гамильтоновой.

Утверждение. В области D_2 переменного управления существует стационарная точка $x^* = (x_1^*, x_2^*, x_3^*, x_4^*)$, если параметры модели лежат внутри эллиптической области Ω , изображенной на рисунке 1. При этом координаты стационарной точки находятся в явном виде

$$\begin{aligned} x_1^* &= b/\alpha, & x_2^* &= (\alpha\rho - b)((1-\alpha)\rho + \alpha\beta - b), \\ x_3^* &= (1-\alpha)\left(1 - \frac{\alpha\rho}{\alpha\beta - b}\right)x_4^*, & x_4^* &= \frac{\alpha\beta - b}{\alpha\rho(1-\alpha)(\beta - \rho) + b(\alpha\beta - b)}. \end{aligned} \quad (14)$$

Управление в стационарной точке вычисляется по формуле:

$$u^* = b/(\alpha\beta). \quad (15)$$

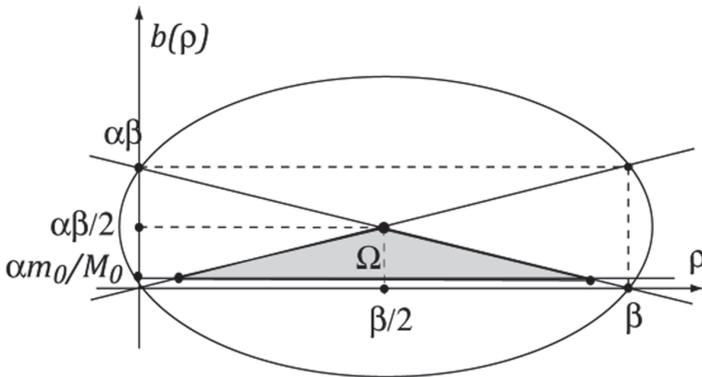


Рис. 1. Область существования стационарной точки

Замечание. Стационарный режим управления лежит в области D_2 переменного управления, если выполнено неравенство $u^* \leq \bar{u} < 1$.

Аппроксимирующее управление

Для построения аппроксимирующего управления мы исходим из того, что изначально запасы ресурсов ограничены величиной $M_0(2)$. Значит можно ожидать, что $M(t) \xrightarrow{t \rightarrow \infty} M_0$.

Момент смены управления

Возьмем любое постоянное начальное управление $u_1 \geq 0$. Зная закон изменения потребляемых в период t ресурсов (8), найдем текущий уровень использования материалов $m_1(t, u_1)$.

Предположим, что найденная функция описывает потребление ресурсов вплоть до некоторого момента времени $T = T(u_1)$, после которого происходит смена управления на стационарный режим u^* (15), и потребление ресурсов подчиняется закону $m_2(t, u^*)$. Следовательно, текущее потребление материалов описывается функцией

$$m(t, u_1) = \begin{cases} m_0 e^{\frac{b - \beta u_1}{1 - \alpha} t}, & t \leq T = t(u_1) \\ m_0 e^{\frac{b - \beta u_1}{1 - \alpha} T} e^{-\frac{b}{\alpha}(t - T)}, & t \geq T = t(u_1) \end{cases} \quad (16)$$

Полученные соотношения (16) подставим в условие ограниченности ресурсов (2), откуда найдем момент времени $T = t(u_1)$ смены управления

$$T(u_1) = \begin{cases} \frac{1 - \alpha}{b - \beta u_1} \ln \left(1 - \frac{b M_0 (\hat{u} - u_1)}{m_0 \alpha (u^* - u_1)} \right), & \beta u_1 \neq b \\ \frac{M_0}{m_0} - \frac{\alpha}{b}, & \beta u_1 = b \end{cases} \quad (17)$$

Замечание. Существует такой уровень управления $\hat{u} = \frac{1}{\beta} (b + (1 - \alpha) \frac{M_0}{m_0})$, что $\lim_{u_1 \rightarrow \hat{u}} T = +\infty$, а при $u_1 > \hat{u}$ указанного момента времени $T(u_1)$ не существует, т. е. условие ограниченности ресурсов будет выполнено, если только начальное управление u_1 не превышает уровня \hat{u} .

Выбор начального управления

Начальное управление u_1 выбирается из того условия, что при достижении фазовыми траекториями $(x_1(t), x_2(t))$ момента времени $T = T(u_1)$ (17), они оказываются на стационарном уровне (x_1^*, x_2^*) .

Используя соотношения (4), (6), (9), найдем $x_i(T(u_1), u_1) = x_i(u_1)$, $i = 1, 2$.

$$x_1(u_1) = \frac{b}{\alpha} \equiv x_1^*, \quad x_2(u_1) = \frac{p_0 M_0}{\alpha \alpha} m_0^\alpha \left(\frac{b M_0}{\alpha m_0} \frac{\hat{u} - u_1}{u^* - u_1} \right)^{-\alpha \beta \frac{u^* - u_1}{b - u_1}} = x_2^*$$

Первое равенство выполнено при всех u_1 . Таким образом, для нахождения управления u_1 используется второе соотношение.

Замечание. Численные эксперименты показывают, что то начальное управление u_1^0 , которое находится из второго равенства $x_2(u_1) = x_2^*$, удовлетворяет ограничениям $0 \leq u_1^0 < \hat{u}$.

Сравнение аппроксимирующего и стабилизированного решений

Стабилизированное решение гамильтоновой системы строится в соответствии с алгоритмом, предложенным в работе [7]. Для этого требуется, в частности, чтобы стационарная точка была седлового типа.

По данным экономики Китая эконометрическая калибровка параметров модели дает следующие результаты: $M_0 = 1.81 \cdot 10^6$, $\bar{u} = 0.12$, $\rho = 0.18$, $a = 64.34$, $b = 0.069$, $\alpha = 0.4091$, $\beta = 1.523$, $p_0 = 100$. Стационарная точка имеет координаты $x^* = (0.17, 0.03, 6.21, -2.22)$, $u^* = 0.11$. Собственные числа якобиана, вычисленного в стационарной точке, равны: $\lambda_1 = -1.832$, $\lambda_2 = -0.011$, $\lambda_3 = 0.203$, $\lambda_4 = 2.000$.

Известно [7], что вблизи установившегося состояния x^* стабилизированные траектории достаточно точно описывают поведение оптимальных решений. Этот факт позволяет оценить качество аппроксимационной схемы в окрестности стационарной точки.

Сравнивая результаты, полученные (см. рисунок 2) при помощи стабилизатора и аппроксимирующего управления, можно отметить, что в целом поведение траекторий схоже, но качество приближения не очень высокое. Поэтому в дальнейшем планируется строить аппроксимирующее управление ступенчатого вида, с большим количеством промежуточных ступеней, роль которых в данной схеме играло начальное управление u_1 .

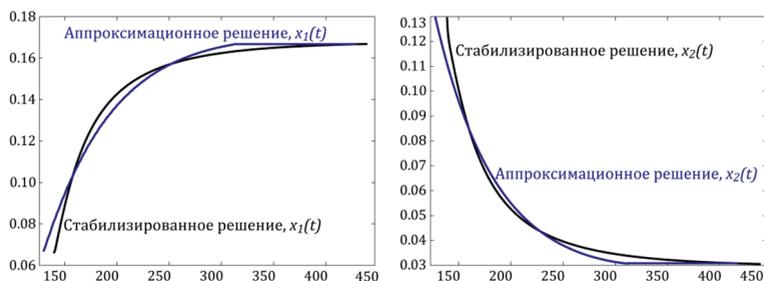


Рис. 2. Сравнение стабилизированных (черная линия) и аппроксимирующих (синяя линия) решений

Литература

1. *Aseev, S.M., Kryazhinskiy, A.V.* The Pontryagin Maximum Principle and Optimal Economic Growth Problems // Proceedings of the Steklov Institute of Mathematics, Pleiades Publishing, 2007, Vol. 257.
 2. *Grossman, G.M., Helpman, E.* Innovation and Growth in the Global Economy. MIT Press, Cambridge, Massachusetts, 1991.
 3. *Hartman, Ph.* Ordinary Differential Equations. J. Wiley & Sons, NY, 1964. OECD Synthesis Report: Measuring Material Flows and Resource Productivity // OECD Publishing, 2008.
 4. *Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F.* The Mathematical Theory of Optimal Processes. Interscience, NY, 1962.
 5. *Solow, R.M.* Growth Theory: An Exposition. NY, Oxford Univ.Press, 1970.
 6. *Tarashev, A.M., Usova, A.A.* Stabilizing the Hamiltonian System for Constructing Optimal Trajectories // Proceedings of the Steklov Institute of Mathematics, Vol. 277, pp. 248–265, 2012.
 7. *Tarashev, A., Zhu, B.* Optimal Proportions in Growth Trends of Resource Productivity // Proceedings of the 15th IFAC Workshop «Control Applications of Optimization» CAO'12, 2012.
-

ПРОГРАММА ПРИЗМА ДЛЯ РАСЧЕТОВ ЗАДАЧ ПЕРЕНОСА ЧАСТИЦ МЕТОДОМ МОНТЕ-КАРЛО

**Ядгар З. КАНДИЕВ, Олег В. ЗАЦЕПИН*,
Геннадий Н. МАЛЫШКИН, Светлана Г. СПИРИНА,
Тамара В. ВАХОНИНА, Нина А. ВОРОНИНА,
Елена А. КАШАЕВА, Лариса В. ЛОБАНОВА,
Геннадий В. ЛУКИН, Алексей А. МАЛАХОВ,
Дмитрий Г. МОДЕСТОВ, Рим Ф. МУХАМАДИЕВ,
Виктор Г. ОРЛОВ, Сергей И. САМАРИН, Елена В. СЕРОВА,
Кирилл Е. ХАТУНЦЕВ**

*Российский Федеральный Ядерный Центр
Всероссийский Научно-Исследовательский Институт Технической Физики
имени академика Е.И. Забабахина (РФЯЦ-ВНИИТФ),
Снежинск, 456770, Россия*

В РФЯЦ-ВНИИТФ для решения задач переноса ионизирующего излучения методом Монте-Карло в течение более тридцати лет развивается программа ПРИЗМА. Программа позволяет моделировать раздельный и совместный перенос нейтронов, фотонов, электронов, позитронов, ионов в одномерной, двухмерной, трехмерной геометрии. Система константного обеспечения использует оцененные данные библиотек БАС-78, ENDF/V V,VI,VII, EPDL 97. Реализована модель термализации нейтронов с учетом теплового движения вещества и химической связи. Для описания периодических структур применяются двух и трехмерные решетки произвольной глубины вложенности. Для учета случайной структуры многокомпонентных полидисперсных сред реализовано моделирование переноса частиц в стохастической геометрии. В программе заложены широкие возможности по описанию источников и вычисляемых функционалов. Для оценки влияния на функционалы малых возмущений геометрии или состава веществ возможно получение в одном расчете коррелированных результатов для заданных вариантов задачи. Для повышения эффективности расчетов разработан единый подход моделирования по ценности для различных классов задач. В рамках этого подхода развиты методы неаналогового моделирования для решения задач прохождения через оптически плотные и прозрачные среды, задач детектирования, включая случаи с расположением детекторов в рассеивающей среде. Для расчета задач на собственное значение для систем с делящимися материалами применяется метод поколений с постоянным числом точек деления в поколении. Процесс генерации следующего поколения и оценка функционалов для текущего поколения разделены на две стадии расчета, что позволяет применять методы неаналогового моделирования на второй стадии.

* Автор, ведущий переписку, E-mail: o.v.zatsepin@vniitf.ru

Возможности программы демонстрируются на примере расчетов двух задач. Первая задача состоит в расчете радиационной защиты медицинской установки для нейтронозахватной терапии. Вторая задача состоит в вычислении нейтронно-физических характеристик для полномасштабной трехмерной модели активной зоны реактора ВВЭР-1000.

КЛЮЧЕВЫЕ СЛОВА: перенос частиц, метод Монте-Карло, неаналоговое моделирование, программа ПРИЗМА, нейтронно-физические характеристики реактора, реактор ВВЭР-1000, нейтронозахватная терапия.

I. Введение

В РФЯЦ-ВНИИТФ в течение более 30 лет развивается универсальная программа ПРИЗМА [1], предназначенная для решения линейных задач переноса методом Монте-Карло. Программа позволяет моделировать отдельный и совместный перенос нейтронов, фотонов, электронов, позитронов и ионов на основе спектральных библиотек оцененных ядерных данных. Программа применяется как для расчета задач с источником, так и для расчета критичности систем с делящимися материалами. Одно из направлений развития программы в ближайшие годы связано с полномасштабным моделированием физических процессов в активных зонах энергетических ядерных реакторов.

Программа ПРИЗМА написана на языке C++, функционирует под управлением операционных систем Linux и Windows, при работе в параллельном режиме использует библиотеку MPI.

В настоящем докладе дан краткий обзор возможностей программы, приводятся примеры расчетов сложных физических систем.

II. Обзор возможностей программы

1. Описание взаимодействия частиц с веществом

Моделирование переноса частиц в программе ПРИЗМА возможно в следующих диапазонах энергии:

- нейтроны — (10–5 эВ — 150 МэВ),
- фотоны — (100 эВ — 100 МэВ),
- электроны и позитроны — (1 кэВ — 100 МэВ),
- ионы — (0.1 кэВ — 300 МэВ).

Система константного обеспечения расчетов [2] содержит текстовые и числовые библиотеки спектральных нейтронных констант с данными о гамма-образовании (БАС [3], ENDF/B-V, -VI, -VII и др.), спектральных фотонных констант с образованием заряженных частиц (EPDL-92, EPDL-97 и др.) и констант взаимодействия заряженных частиц с веществом.

При моделировании взаимодействия нейтронов с ядрами вещества рассматриваются следующие типы реакций: упругое рассеяние, неупругое рассеяние, деление, поглощение, неупругое столкновение с испусканием заряженных частиц, реакции $(n, 2n)$ и $(n, 3n)$. Для учета температуры среды реализована модель максвелловского одноатомного газа. В области термализации нейтронов может применяться модель учета межатомных связей, основанная на использовании библиотек оцененных ядерных данных формата ENDF [4]. При взаимодействии нейтрона с ядрами, продукты реакции, в том числе ядра отдачи, могут рассматриваться как вторичные частицы.

При моделировании фотонов предусмотрен учет следующих реакций: когерентное рассеяние, некогерентное рассеяние, образование пар, фотопоглощение, фотоядерная реакция. При моделировании электронов учитываются три реакции: упругое рассеяние, неупругое рассеяние, тормозное излучение, а при моделировании позитронов – к ним еще добавляется реакция аннигиляции.

При моделировании ионов учитываются упругое кулоновское рассеяние, неупругие процессы возбуждения и ионизации атомов в приближении непрерывного замедления, ядерные упругие и неупругие процессы. Продукты ядерных реакций, в том числе ядра и ядра отдачи, могут рассматриваться как вторичные частицы.

2. Описание геометрии

При описании геометрии задачи используется двухэтапное разбиение рассчитываемой системы на составные части. На первом этапе система разбивается на блоки — части системы, геометрия которых описывается в собственной (относительной) системе координат. Блоки, в свою очередь, состоят из областей, внутри которых химический состав постоянен, а плотность может быть как постоянной, так и переменной. Взаимное расположение блоков описывается с помощью операций сдвига и поворота, определяющих связь относительной системы координат блока с абсолютной системой координат. Возможно вложение одних блоков в другие без ограничений глубины вложенности, при этом не допускается касание или пересечение границ блоков.

Предусмотрены одномерные, двумерные и трехмерные блоки. Одномерные блоки (плоский, сферический, цилиндрический) характеризуются изменением одной координаты. Двумерные блоки позволяют описывать тела, обладающие осевой или цилиндрической симметрией. Полное описание области двумерного блока дает описание сечения области в плоскости симметрии; внешний контур сечения может состоять из произвольного набора отрезков прямых, окружностей и эллипсов. Для описания областей трехмерных блоков используется две возможности. Первая из них — это традиционный комбинаторный подход, в котором в качестве примитивов

рассматриваются различные поверхности (плоскости, сферы, цилиндры, конусы и другие поверхности второго порядка). Вторая возможность – это описание области путем перечисления ограничивающих ее кусков поверхностей. В этом случае усложняется задание исходных данных за счет дополнительного описания кусков поверхностей, вырезаемых из каждой поверхности другими поверхностями, однако обеспечивается более высокий уровень контроля ошибок в исходных данных и более высокая эффективность геометрических алгоритмов.

Для решения задач переноса излучения в реакторной геометрии, содержащей периодические структуры в виде решеток, существуют дополнительные возможности по описанию двух- и трехмерных решеток с произвольным числом ячеек: прямоугольной, косоугольной, гексагональной. Отдельная ячейка решетки может иметь достаточно сложную трехмерную геометрию, в том числе представлять собой решетку следующего уровня вложенности. Заполнение ячеек геометрической области происходит либо автоматически, когда заполняются все внутренние ячейки, либо выборочно — по номерам ячеек.

Для учета случайной структуры дисперсных сред (например, дисперсного топлива) возможно моделирование в стохастической геометрии, описывающей многокомпонентные полидисперсные среды со случайно ориентированными включениями, имеющими форму выпуклых тел вращения с произвольным разбиением на области и заданным распределением по размерам [5]. В общем случае многокомпонентная среда состоит из нескольких сортов включений, статистически равномерно распределенных в трехмерной области без пересечения друг с другом, и связующего, заполняющего остальную часть области. Каждая компонента среды характеризуется составом материала и объемной концентрацией.

3. Описание источников

Пространственно источники могут задаваться равномерно по объему произвольной геометрической области и равномерно по площади геометрических поверхностей. Также предусмотрен широкий набор примитивов для автономного описания геометрии источника, в том числе с неравномерным распределением по пространству. Для описания распределения по энергии, углу и времени существует набор одномерных аналитических распределений, а также возможность задания произвольных одномерных аналитических распределений и 1-, 2- и 3-мерных табличных распределений.

4. Описание результатов

Линейные функционалы от решения уравнения переноса могут оцениваться на любых геометрических поверхностях и внутри областей системы. По-

верхностные результаты — это токи и потоки частиц, токи и потоки энергии. Областные результаты — это плотность и поток частиц, плотность столкновений, поглощенная энергия, число заданных реакций, число вторичных частиц определенного типа и ряд других. Областные и поверхностные результаты можно получать с весом заданной функции (табличной или аналитической) от параметров частицы. Любые функционалы могут быть распределенными по пространству, направлению полета частиц, их энергии и времени жизни.

Для получения более детальной информации о процессе распространения частиц используется возможность выделения вклада частиц, траектории которых удовлетворяют определенным условиям [6]. В общем случае условие представляет собой логическую комбинацию событий, описывающих диапазоны изменения фазовых координат частицы и ряда дополнительных параметров, к числу которых относятся: номер источника, тип частицы, номер блока и номер области, номер соударения, тип взаимодействия и др.

Для оценки влияния на функционалы малых возмущений геометрии, состава или плотности веществ в локальной области системы существует возможность получать в одном расчете коррелированные результаты нескольких вариантов задачи.

5. Неаналоговое моделирование

Для решения задач, в которых необходимо вычислять функционалы, связанные с малыми вероятностями, разработан метод моделирования по ценности [7], позволяющий адаптировать алгоритмы моделирования к особенностям конкретной задачи. Данный метод включает в себя предварительную построение приближенной функции ценности частиц и вычисление на ее основе весовой функции и параметров неаналоговых распределений.

При разработке метода учитывался тот факт, что в программе ПРИЗМА используются оценки по посещениям, то есть оценки, вклады в которые отличны от нуля только для траекторий частиц, имеющих общие точки с областью или поверхностью интегрирования (детектором). Поэтому для вычисления любого функционала моделирование траекторий необходимо строить таким образом, чтобы увеличивать число частиц, попадающих в детектор. А это означает, что необходимо искусственно увеличивать плотность столкновений в окрестности детектора и в нем самом, а также плотность частиц, летящих в детектор и его окрестность.

Выделены четыре класса элементарных задач, при решении которых требуется применение моделирования по ценности:

- прохождение через большие оптические толщины;
- моделирование процессов взаимодействия, имеющих малые вероятности;
- прохождение в детекторы, расположенные в вакууме или чистом поглотителе;

- прохождение в детектор, расположенный в излучающей и рассеивающей среде.

К комбинации элементарных задач можно свести большинство задач линейного переноса излучения. Для каждого класса элементарных задач развиты методы неаналогового моделирования, которые включают в себя:

- моделирование траекторий частиц с использованием неаналоговых распределений (экспоненциальное преобразование, метод вынужденных столкновений, смещение углового распределения, смещение вероятностей процессов взаимодействия);
- приведение статистического веса частицы на различных этапах моделирования траектории в соответствии с заданной весовой функцией фазовых координат частицы.

Программа ПРИЗМА позволяет задавать параметры неаналоговых распределений и приближенные функции ценности для элементарных задач. В программе реализованы методы расчетов сложных задач путем выделения элементарных подзадач различных классов. Расчет исходной задачи производится путем построения траекторий частиц согласно схеме моделирования по ценности той элементарной подзадачи, в рамках которой находятся частицы.

6. Расчеты критичности

Для расчета нейтронно-физических характеристик систем с делящимися материалами применяется метод поколений с постоянным числом точек деления в поколении [8]. Процесс генерации следующего поколения и оценка функционалов для текущего поколения разделены на две стадии расчета, что позволяет применять методы неаналогового моделирования на второй стадии расчета. Благодаря этому возможно решение задач радиационной защиты непосредственно в ходе расчета критичности.

III. Расчеты радиационной защиты установки для нейтронозахватной терапии

В ИЯФ им. Будкера (Новосибирск, Россия) для применения в нейтронозахватной терапии сооружен пилотный вариант источника эпитепловых нейтронов на основе ускорителя-тандема [9]. Он предназначен для генерации нейтронов в пороговой реакции ${}^7\text{Li}(p,n){}^6\text{He}$ при облучении литиевой мишени пучком протонов 10мА; энергия протонного пучка — 1.915 или 2.5 МэВ.

1. Постановка задачи

Для оценки радиационной безопасности источника нейтронов, находящегося внутри толстостенного бункера, по программе ПРИЗМА выполнены

расчеты двумерного пространственного распределения мощности эквивалентной дозы от нейтронов и вторичных гамма-квантов за внешней стеной бункера, выходящей на улицу.

На основе реальной схемы бункера, установки и мишени разработана расчетная модель, геометрия которой представлена на Рисунках 1–3.

Модель мишени (см. Рисунок 1) представляет собой тело вращения и включает тонкий слой лития на медной подложке, ниже которой задан слой воды. Остальная часть мишени выполнена из стали. В качестве источника рассматривается пучок протонов с энергией 1.915 МэВ и интенсивностью 6×10^{16} протонов/с.

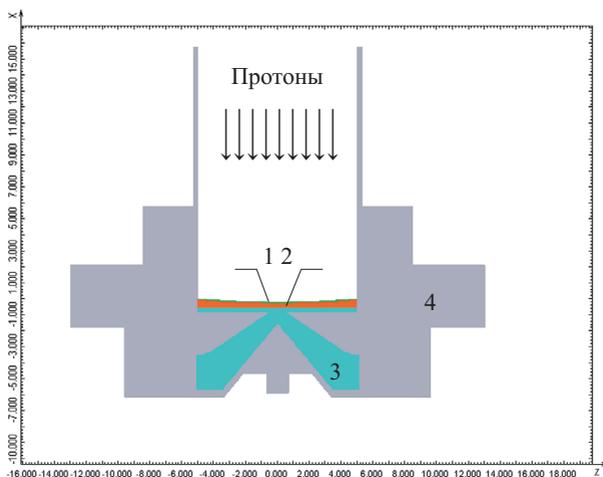


Рис. 1. Расчетная модель мишени:

1 — слой лития, 2 — медная подложка, 3 — вода, 4 — сталь

Мишень находится ниже нулевого уровня бункера (см. Рисунок 2). На уровне 0 (земля) и -2630 мм задан сплошной пол по всей площади бункера толщиной 24 мм и 16 мм, соответственно. Установка описана в виде ряда полых цилиндров из смеси углеродистой стали, нержавеющей стали, меди, алюминия или индия. Вокруг бункера ниже уровня 0 был задан грунт, а выше — воздух.

Стены бункера (см. Рисунок 3) состоят из бетона толщиной 1.2–1.3 м, с внешней керамзитобетонной панелью. В стене, граничащей с коридором, расположена стальная дверь.

Распределение дозы оценивалось на экранах-детекторах, расположенных на внешней стене бункера и параллельно ей на расстоянии 1 м от стены. Размеры экранов — 1080×550 см², шаг сетки — 5×5 см².

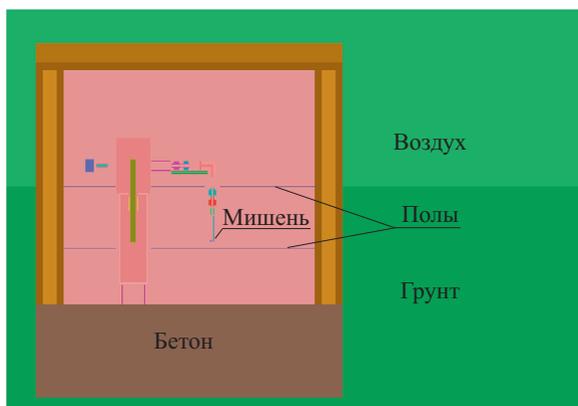


Рис. 2. Геометрия модели бункера в плоскости ZX (вид сбоку)

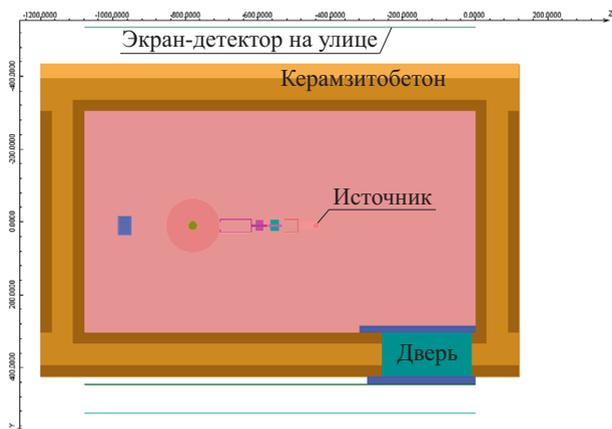


Рис. 3. Геометрия модели бункера в плоскости ZY (вид сверху)

2. Результаты расчетов

В расчетах по программе ПРИЗМА моделировался перенос протонов в литиевой мишени, рождение в ней нейтронов и гамма-квантов, их дальнейшее распространение внутри бункера и выход за его пределы. Учитывая большую оптическую толщину стен бункера, для получения результатов с приемлемой статистической погрешностью на очень подробной пространственной сетке применялся метод экспоненциального преобразования.

Результаты расчетов приведены на Рисунках 4, 5.

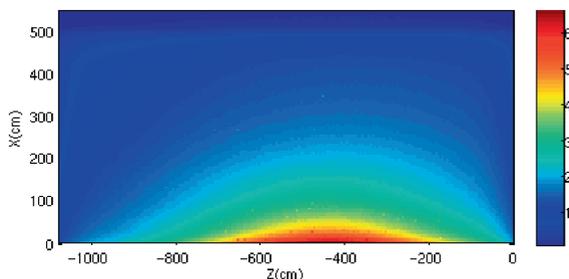


Рис. 4. Распределение мощности эквивалентной дозы (мкЗив/ч) на внешней стене бункера

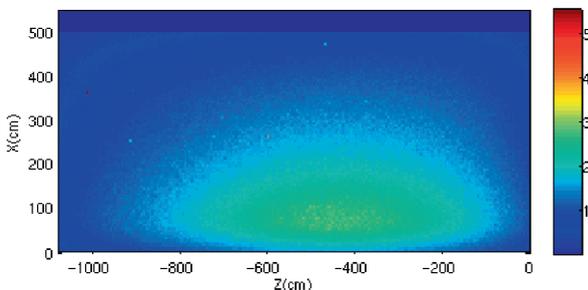


Рис. 5. Распределение мощности эквивалентной дозы (мкЗив/ч) на расстоянии 1 м от внешней стены бункера

Анализ полученных результатов показал, что максимальная мощность дозы за внешней стеной бункера на улице достигается на вертикали напротив места расположения мишени, что вполне объяснимо, так как в плоскости экрана стена однородна по составу. Максимальная мощность дозы на внешней поверхности стены бункера составляет 6.6 мкЗв/ч при относительной погрешности 0.3% (одно среднеквадратичное отклонение). Максимальная мощность дозы на расстоянии 1 м от стены составляет 4.0 мкЗв/ч при относительной погрешности 2.3%. Основной вклад в дозу дают фотоны с высокой энергией (порядка 1 МэВ), рожденные в слоях защиты, сечение взаимодействия которых меньше, чем у нейтронов, имеющих существенно меньшую энергию.

IV. Расчеты нейтронно-физических характеристик реактора ВВЭР-1000

По программе ПРИЗМА выполнены прецизионные расчеты нейтронно-физических характеристик полномасштабной трехмерной модели активной

зоны реактора ВВЭР-1000 [10] 1-го блока Волгодонской АЭС с начальной топливной загрузкой. Технические характеристики модели основаны на проектных данных и данных по пусковым экспериментам этого блока. Явно описывались такие детали, как полости в топливных таблетках, каналы в выгородке, высотная структура поглощающих стержней системы управления и защиты.

Активная зона (АЗ) высотой 353 см состоит из гексагональных тепло-выделяющих сборок (ТВС) различного типа, отличающихся материальным составом топлива (см. Рисунок 6). АЗ окружена отражателем, в состав которого входят, начиная от центра, водяной зазор, стальная выгородка с каналами, заполненными водой, шахта реактора, вода, корпус реактора.

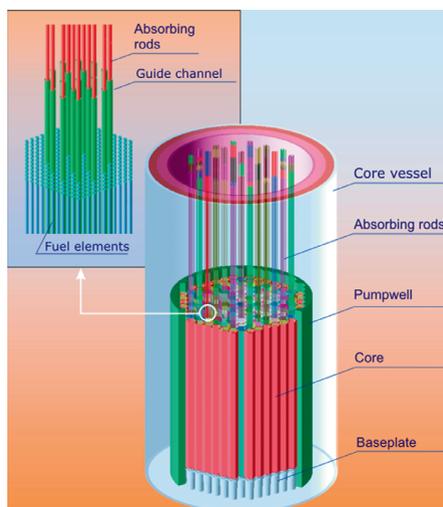


Рис. 6. Общий вид модели реактора

ТВС состоит из 331-й гексагональной ячейки. По геометрии и конфигурации материалов различают четыре типа ячеек: ячейка с тепловыделяющим элементом (твэлом) — окисным топливом различного обогащения по изотопу ^{235}U , центральная трубка или направляющий канал, канал с выгорающим поглотителем, канал с поглощающим элементом (ПЭЛ). Каналы с поглощающими элементами в определенных ТВС образуют группы, общее число групп — 10. Положение группы задается процентом ее извлечения (от 0 до 100 %) из активной зоны.

Расчеты проведены с системой констант ENDF/B VI, данные о взаимодействии нейтронов с учетом химической связи брались из библиотеки ENDF/B VII.

Для оценки влияния модели термализации нейтронов на эффективный коэффициент размножения нейтронов ($K_{эфф}$) проведено три расчета для критического состояния реактора [11]. В первом расчете для всех веществ использовалась модель одноатомного идеального газа, Во втором расчете эта модель использовалась для всех веществ, исключая воду, в которой учитывалась химическая связь. В третьем расчете химическая связь учитывалась в воде и двуокиси урана. Расчеты проводились методом поколений, размер поколения — 128 000 частиц. Оценка $K_{эфф}$ со статистической погрешностью 0.003% представлена во втором столбце Таблицы 1. В третьем столбце таблицы приведена эффективность расчетов, пропорциональная числу промоделированных траекторий в каждом расчете при одинаковом времени счета.

Модель термализации нейтронов	$K_{эфф}$	Эффективность
идеальный газ	1.00210	1
химическая связь в H_2O	1.00165	0.88
химическая связь в H_2O и UO_2	1.00149	0.85

Таблица 1. Зависимость $K_{эфф}$ и эффективности расчета от модели термализации нейтронов

График сходимости $K_{эфф}$ с ростом числа поколений представлен на Рисунке 7.

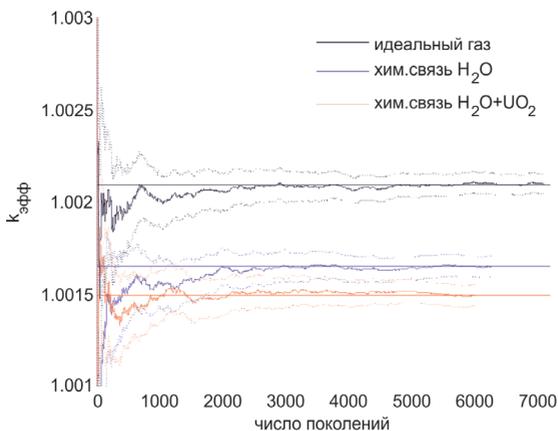


Рис. 7. Зависимость $K_{эфф}$ от числа поколений

На Рисунке 7 пунктирной линией обозначены доверительные интервалы в два среднеквадратичных отклонения, прямая линия — конечная оценка

Кэфф. Как видно, учет химической связи в топливе мало влияет на Кэфф, поэтому в дальнейших расчетах химическая связь учитывалась только в воде.

Результаты расчета покассетного распределения энерговыделения для состояния реактора, обладающего 60-градусной симметрией, представлены на Рисунке 8. Значения на цветовой шкале приведены в условных единицах.

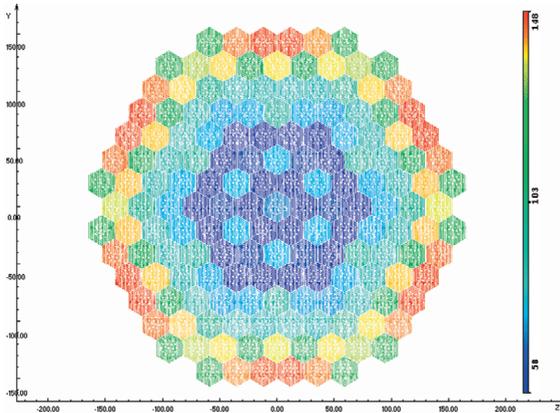


Рис. 8. Распределение энерговыделения по ТВС

Отметим, что результаты расчетов $K_{эфф}$ и покассетного распределения энерговыделения, полученные по программе ПРИЗМА, согласуются с результатами, полученными по программе MCU. [11]

На Рисунке 9 представлены результаты расчетов распределения энерговыделения по высоте активной зоны в восьми ТВС от центра, ТВС № 82 (нижняя кривая), к периферии АЗ, ТВС № 138 (верхняя кривая) для состояния реактора, в котором ПЭЛ 9-ой группы находятся на нижнем концевики, а ПЭЛ 10-ой группы извлечены из АЗ на 64%.

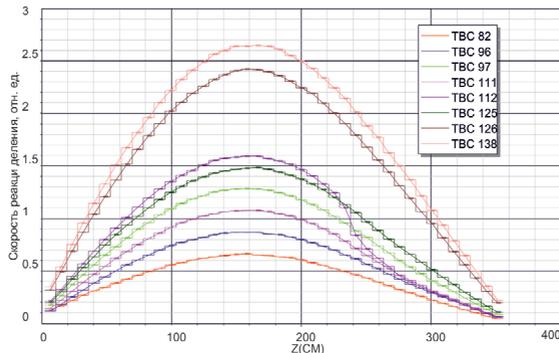


Рис. 9. Распределение энерговыделения в восьми ТВС по высоте

График энерговыделения в ТВС № 112 имеет перегиб, что связано с расположением ПЭЛ 10-ой группы в этой ТВС.

Результаты расчетов потвального распределения энерговыделения по высоте активной зоне представлены на Рисунке 10 для ТВС, соседней с центральной, а на Рисунке 11 — для периферийной ТВС.

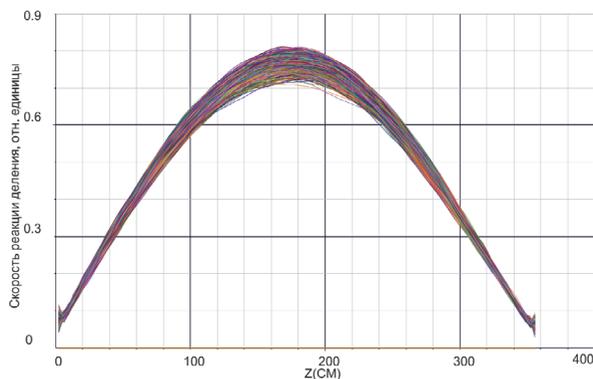


Рис. 10. Потвальное распределение энерговыделения по высоте в ТВС, соседней с центральной

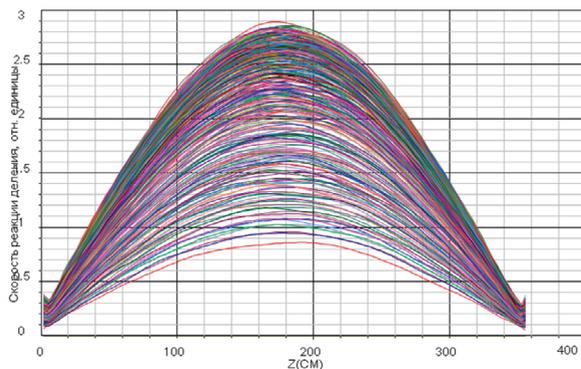


Рис. 11. Потвальное распределение энерговыделения по высоте в периферийной ТВС

Как видно, распределение по высоте в отдельных твэлах имеет достаточно плавный вид, что получено благодаря представлению результатов в виде полиномиальной зависимости.

Для оценки влияния уровня извлечения поглощающих элементов (ПЭЛ) на величину Кэфф был проведен расчет для десяти вариантов уровня извлечения ПЭЛ 10-ой группы. Расчет проведен по специальной методике оценки

малых возмущений в многовариантных расчетах. [6,7] В расчете оценивались значения $K_{эфф}$, а также разности значений $K_{эфф}$ базового варианта (100% извлечение ПЭЛ) и остальных вариантов ($\Delta K_{эфф}$). Отметим, что применение методики оценки малых возмущений позволило вычислить разности значений $K_{эфф}$ меньшие, чем значения статистической погрешности. При этом эффективность многовариантного расчета для данной задачи приблизительно в 1 000 раз выше, чем эффективность независимых расчетов для всех вариантов. Результаты расчета и погрешность (одно среднеквадратичное отклонение) приведены в Таблице 2.

Уровень извлечения ПЭЛ, %	$K_{эфф}$	$\Delta K_{эфф}$
100	1.001861±0.000012	
90.4	1.001848±0.000012	0.000013±0.000001
82	1.001770±0.000012	0.000092±0.000003
80	1.001737±0.000012	0.000125±0.000003
68.7	1.001426±0.000013	0.000435±0.000006
57.3	1.000933±0.000013	0.000928±0.000008
46	1.000272±0.000015	0.001589±0.000011
23.3	0.998960±0.000017	0.002901±0.000015
12	0.998624±0.000017	0.003237±0.000016
0	0.998531±0.000018	0.003330±0.000016

Таблица 2. Зависимость $K_{эфф}$ от уровня извлечения ПЭЛ

В. Заключение

Универсальная программа ПРИЗМА, разработанная в РФЯЦ-ВНИИТФ, обладает широкими возможностями по решению разнообразных задач переноса нейтронов, фотонов и заряженных частиц методом Монте-Карло. Одним из направлений развития программы на ближайшее будущее является моделирование физических процессов в активных зонах энергетических ядерных реакторов.

Литература

1. *M.A. Arnautova, Ya. Z. Kandiev, B.E. Lukhminsky, G.N. Malyshkin*, «Monte-Carlo simulation in nuclear geophysics. In comparison of the PRIZMA Monte-Carlo program and benchmark experiments», Nuclear Geophysics, 7[13], 407 (1993).
2. *V.M. Shmakov, V.D. Lyutov, E.I. Cherepanova et al.*, «Use of ENDF-format Libraries for Criticality Calculations at VNIITF» — The Sixth International Conference on Nuclear Criticality Safety, ICNC'99, Versailles, France, v.I, pp. 353–357, Sept. 20–24, 1999.
3. *A.P. Vasilyev, E.S. Kuropatenko, V.D. Lyutov, A.I. Orlov, V.M. Shmakov*. Nuclear Data Library — BAS. The history of development and validation for criticality

- safety calculations. ICNC'95. Proceedings of the international conference of nuclear criticality safety, Albuquerque, New Mexico, USA, September 17–21, 1995, pp 2.56–2.60.
4. *Модестов Д.Г.* К вопросу об описании рассеяния медленных нейтронов в формате ENDF. — ВАНТ, Сер. Математическое моделирование физических процессов, 2008, вып.3, с.47–54.
 5. *G.N. Malyshkin, E.A. Kashaeva and R.F. Mukhamadiev*, «Stochastic Geometry in PRIZMA Code», The Joint International Topical Meeting on Mathematics and Computation and Supercomputing in Nuclear Applications, Monterey, California, USA, April 15–19, 2007.
 6. *Я.З. Кандиев, Е.В. Серова*, «Меченые частицы в расчетах задач переноса излучения методом Монте-Карло по программе ПРИЗМА», Атомная энергия, 98[6], 386 (2005).
 7. *Ya. Z. Kandiev, G.N. Malyshkin*. Modeling by Value Implemented in PRIZMA Code. V Joint Russian-American Computational Mathematics Conference // Sandia Report. SAN98-1591, 1998, pp.149–158.
 8. *Я.З. Кандиев, А.А. Малахов, Е.В. Серова, С.Г. Спирина*, «Оценка эффекта малых возмущений в многовариантных расчетах по программе ПРИЗМА-Д», Атомная энергия, , 2005, т.99, вып.3, с.203–210.
 9. *Vayanov V., Belov V., Bender E.* et al., 1998. Accelerator based neutron source for the neutron-capture and fast neutron therapy at hospital. Nucl. Instrum. and Methods. in Phys. Res. A., 413, 397–426.
 10. *А.К. Горохов, Ю.Г. Драгунов, Г.Л. Лунин, А.Н. Новиков, В.И. Цофин, Ю. А. Анянцев*, Обоснование нейтронно-физической и радиационной частей проектов ВВЭР, Академкнига, Москва, 496 (2004).
 11. *Д.С. Олейник, М.А. Калугин, Е.А. Сухино-Хоменко, Г.Н. Малышкин, О.В. Зацепин, Я.З. Кандиев, Е.А. Кашаева, Д.Г. Модестов*, «Полномасштабная математическая модель переноса нейтронов в активной зоне реактора ВВЭР-1000, основанная на методе Монте-Карло и реализованная на многопроцессорных ЭВМ», Обеспечение безопасности АЭС с ВВЭР, 26–29 мая 2009, Подольск, Россия (2009).
-

ЗАДАЧА ВЫВОДА РАКЕТЫ-НОСИТЕЛЯ НА ОРБИТУ В УСЛОВИЯХ СЛУЧАЙНЫХ ВОЗМУЩЕНИЙ ПАРАМЕТРОВ¹

Козьмин И.В.

*м. н. с. отдела прикладных проблем управления ИММ УрО РАН,
ikozmin@imm.uran.ru*

Аннотация: Рассматривается задача максимизации массы полезной нагрузки, выводимой ракетой-носителем на заданную орбиту.

Исследуется ситуация, когда параметры нелинейной математической модели управляемого движения носителя задаются с помощью нормально распределенных случайных величин. В ходе решения задачи был проведен объемный численный эксперимент на многопроцессорном вычислителе кластерного типа «Уран». На основе данных полученных в результате проведения эксперимента была получена оценка максимальной массы полезной нагрузки, которая может быть выведена носителем на заданную орбиту с вероятностью не меньшей заданного порога.

Введение

С 2003 года Институт математики и механики им. Н.Н. Красовского УрО РАН (ИММ) активно сотрудничает с НПО автоматики им. акад. Н. А. Семихатова (г. Екатеринбург) (НПОА). В рамках этого сотрудничества решается задача оптимального выведения ракеты-носителя (РН) на заданную околоземную эллиптическую орбиту в условиях случайных возмущений параметров. Такая задача исследуется на базе математической модели управляемого движения РН, которая описывается нелинейной динамической системой. В качестве управления используются угловые скорости изменения углов тангажа и рысканья. Эти углы определяют угловую ориентацию строительной оси РН.

В работе рассматривается ситуация, когда параметры РН и термодинамические характеристики атмосферы математически описываются нормально-распределенными случайными величинами с известными средними E и среднеквадратическими отклонениями σ . Задача заключается в построении оценки максимальной массы полезной нагрузки, которая может быть выведена носителем на заданную орбиту с вероятностью не меньшей заданного порога P^* .

¹ Работа выполнена в рамках программ фундаментальных исследований Президиума РАН «Динамические системы и теория управления» при поддержке УрО РАН (проект 12-П-1-1022), «Информационные, управляющие и интеллектуальные технологии и системы» (проект 12-П-1-1023) и интеграционного проекта УрО и СО РАН (проект 12-С-1-1017).

1. Математическая модель управляемого движения ракеты-носителя

Уравнения движения центра масс РН в инерциальной стартовой системе координат от момента старта t_s до момента t_f выхода на заданную орбиту [1] могут быть записаны в следующем компактном виде:

$$\dot{x} = v, \quad \dot{v} = W(t, x, v, m, \vartheta, \psi), \quad \dot{m} = -\mu t, \quad \dot{\vartheta} = u_1, \quad \dot{\psi} = u_2,$$

$$x(t_s) = x_s, \quad v(t_s) = v_s, \quad m(t_s) = m_s, \quad \vartheta(t_s) = \vartheta_s, \quad \psi(t_s) = \psi_s$$

$$t \in [t_s, t_f]$$

где $x \in R^3$, $v \in R^3$ — координаты и скорости центра масс РН; m — масса РН; $\mu t > 0$ — расхода массы; $W(t, x, v, m, \vartheta, \psi)$ — ускорение (задается суммой составляющих, определяемых реактивными, аэродинамическими и гравитационными силами); ϑ , ψ — углы тангажа и рысканья; t_s — момент начала движения РН; t_f — момент выхода РН на заданную орбиту.

В качестве управляющих параметров выступают скорости u_1 и u_2 изменения углов ϑ и ψ . Удовлетворяют ограничениям

$$|u_1(t)| \leq u_1^{\max}, \quad |u_2(t)| \leq u_2^{\max}, \quad t \in [t_s, t_f].$$

Орбита, на которую выводится РН, определяется следующими параметрами (параметры оскулирующей орбиты [1]): наклонение плоскости орбиты i , долгота восходящего узла Ω , минимальная высота орбиты h_{\min} , максимальная высота орбиты h_{\max} , аргумент перигея ω .

В рассматриваемой задаче момент времени t_f выхода РН на заданную орбиту не фиксирован.

Вывод максимальной массы полезной нагрузки на заданную орбиту

Известно [2], что задача максимизации выводимой на заданную орбиту массы M полезной нагрузки может быть сведена к задаче минимизации момента выхода РН на заданную орбиту с фазовыми ограничениями, то есть задаче оптимального быстродействия с терминальными ограничениями:

$$\min \{J[(\cdot)] \mid u \in U\}$$

при ограничениях

$$|O(t_f) - \bar{O}| \leq \bar{\Delta}_0.$$

Здесь

$$J[u(\cdot)] = t_f; \quad U = \{u \in R^2 \mid |u_1| \leq u_1^{\max}, |u_2| \leq u_2^{\max}\};$$

$$u = (u_1, u_2)^T \in R^2, \quad u_1 = \dot{\vartheta}, \quad u_2 = \dot{\psi};$$

$$O(t_f) = (i(t_f), \Omega(t_f), h_{\max}(t_f), h_{\min}(t_f), \omega(t_f))^T \in R^5;$$

$$\bar{O} = (\bar{i}, \bar{\Omega}, \bar{h}_{\max}, \bar{h}_{\min}, \bar{\omega})^T; \quad \bar{\Delta}_0 = (\bar{\Delta}_i, \bar{\Delta}_\Omega, \bar{\Delta}_{h_{\max}}, \bar{\Delta}_{h_{\min}}, \bar{\Delta}_\omega)^T;$$

где $\bar{i}, \bar{\Omega}, \bar{h}_{\min}, \bar{h}_{\max}, \bar{\omega}$ — заданные значения параметров орбиты, $|i - \bar{i}| \leq \bar{\Delta}_i$, $|\Omega - \bar{\Omega}| \leq \bar{\Delta}_\Omega$, $|h_{\min} - \bar{h}_{\min}| \leq \bar{\Delta}_{h_{\min}}$, $|h_{\max} - \bar{h}_{\max}| \leq \bar{\Delta}_{h_{\max}}$, $|\omega - \bar{\omega}| \leq \bar{\Delta}_\omega$ — допустимые отклонения от этих параметров.

В исследуемой задаче могут быть рассмотрены две группы параметров. Параметры атмосферы (термодинамические характеристики атмосферы) — температура, плотность, давление воздуха; составляющие скорости систематического ветра. И параметры РН: массы его основных конструктивных блоков (сухие массы трёх ступеней); массы запасов топлива на каждой ступени; расход топлива на каждой ступени. В данной работе исследовались только случайные возмущения параметров РН.

Содержательно постановка задачи может быть сформулирована следующим образом. На заданную орбиту в момент t_f выводится только последняя ступень РН, на которой находится полезная нагрузка и остатки топлива этой ступени. За счёт уменьшения запасов топлива на последней ступени можно увеличить массу выводимой на орбиту полезной нагрузки. Задача заключается в том, что для заданного порога вероятности P^* выхода РН на орбиту требуется определить верхнюю оценку дополнительной массы, на которую может быть увеличена масса выводимой полезной нагрузки.

Оценка максимального приращения к массе полезной нагрузки, выводимой на орбиту с заданной вероятностью

Обозначим через M математическое ожидание массы полезной нагрузки и через ΔM — искомую добавку к M . Остальные параметры РН, перечисленные выше, будем рассматривать как нормально распределенные случайные величины с известными математическими ожиданиями и СКО. Обозначим сухую массу третьей ступени через M_3 . Тогда выводимая на орбиту за время t_f масса РН $m_{\text{РН}} = m_{\text{РН}}(t_f)$ также является случайной величиной, которая не может быть меньше суммы сухой массы третьей ступени M_3 и массы полезной нагрузки $M + \Delta M$:

$$m_{\text{РН}} \geq M_3 + M + \Delta M.$$

Отсюда для приращения ΔM к массе M имеем оценку сверху:

$$\Delta M \leq m_{\text{РН}} - M_3 - M.$$

Пусть $P(m) = P\{m_{\text{РН}} \leq m\}$ — функция распределения случайной величины $m_{\text{РН}}$. Поскольку, чем больше масса РН, тем меньше вероятность P_v вывода её на орбиту, то разумно положить $P_v = 1 - P(m)$. Случайная величина M_3 распределена по известному нормальному закону, поэтому задача определения максимальной полезной нагрузки при заданном уровне P^* вероятности вывода P_v её на орбиту сводится к вычислению λ^* -квантили $m_{\lambda^*} = \min\{m \mid P(m) \geq \lambda^*\}$ распределения $P(m)$ для $\lambda^* = 1 - P^*$. Искомая оценка максимальной добавки ΔM^* будет равна:

$$\Delta M^* \leq m_{\lambda^*} - (M_3)_{P^*} - M.$$

Здесь $(M_3)_{P^*}$ — P^* -квантиль нормального распределения случайной величины M_3 , вычисляемый известным образом [3].

Рассмотрим задачу об оценке λ^* -квантиля $m_{\lambda^*}(\lambda^* = 1 - P^*)$ неизвестного распределения $P(m)$. В данной работе эта задача решается с помощью статистической оценки m_{λ^*} , полученной на основе объёмного вычислительного эксперимента с использованием реальных данных. Эксперимент проводился на многопроцессорном вычислителе кластерного типа «Уран». Оценка m_{λ^*} для $\lambda^* = 0,003$ проводилась на выборке объёмом около 440 000 с использованием алгоритма стохастической аппроксимации [3].

На основе проведенного эксперимента было получено приближенное значение ΔM^* величины максимального приращения к массе $M = 5881$ кг полезной нагрузки, выводимой на заданную орбиту с вероятностью не ниже $P^* = 0,997$:

$$\Delta M^* = 318,54 \text{ кг.}$$

Заключение

Рассмотренная в работе методика была реализована в рамках крупномасштабного численного эксперимента с использованием реальных данных. Результаты численного моделирования показали, что учёт случайных возмущений параметров РН позволяет получить значимую оценку для максимального приращения к массе полезной нагрузки, выводимой на орбиту с вероятностью не ниже заданной. Планируется продолжить исследование влияния параметров атмосферы на максимальное приращение к массе полезной нагрузки, выводимой на орбиту с вероятностью не ниже заданной.

Литература

1. *Охоцимский Д. Е., Сихарулидзе Ю.Г.* Основы механики космического полета. М.: Наука, 1990. 448 с.
 2. *Думиева Т.Д., Костоусов В.Б., Костоусова Е.К., Починский В.И.* Исследование задачи оптимального вывода полезной нагрузки на заданную эллиптическую орбиту// Труды Института математики и механики. УрО РАН. 2010. Т.16. №5. С.57–65.
 3. *Кибзун А.И., Кан Ю.С.* Задачи стохастического программирования с вероятностными критериями. М.: Физматлит, 2009.
-

О ЧИСЛЕННОМ РЕШЕНИИ ОДНОЙ ДИФФЕРЕНЦИАЛЬНОЙ ИГРЫ С ЗАПАЗДЫВАНИЕМ¹

Гомоюнов М.И.

м. н. с., ИММ УрО РАН, gomojunov@mail.ru

Аннотация: Для динамической системы, управляемой в условиях помех и содержащей последствие по состоянию, рассматривается задача о вычислении оптимального гарантированного результата и построении закона управления, обеспечивающего этот результат. Оптимизируемый показатель состоит из двух слагаемых. Первое оценивает историю движения системы, сформировавшуюся к терминальному моменту времени, второе представляет собой интегральную оценку реализаций управления и помехи. Сравниваются два подхода для решения рассматриваемой задачи. Первый основан на дискретизации показателя качества и сведении задачи к нахождению цены и седловой точки в дифференциальной игре без последствия и с терминальной оценкой движения. Второй подход базируется на использовании в качестве модели-поводыря конечномерной аппроксимации исходной системы с запаздыванием.

Постановка задачи

В рамках теоретико-игрового подхода [1] рассматривается задача об управлении по принципу обратной связи для системы

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + A_h(t)x(t-h) + B(t)u(t) + C(t)v(t), \\ t_0 < t < \vartheta, \quad x &\in \mathbb{R}^n, \quad u \in \mathbb{R}^r, \quad v \in \mathbb{R}^s, \quad h = \text{const} > 0, \end{aligned} \quad (1)$$

с начальным условием

$$x(t) = x_0(t), \quad t \in [t_0 - h, t_0], \quad (2)$$

и показателем качества

$$\gamma = \left(\int_{v-h}^v \|x(t)\|^2 dt \right)^{1/2} + \int_{t_0}^v [\langle u(t), \Phi(t)u(t) \rangle - \langle v(t), \Psi(t)v(t) \rangle] dt. \quad (3)$$

¹ Работа поддержана грантом РФФИ №12-01-31300-мол_а, а также программой АБЦП 1.994.2011 «Устойчивые вычислительные методы анализа динамики сложных систем».

Здесь t — время, x — фазовый вектор, u — вектор управления, v — вектор помехи; t_0 и ϑ — начальный и терминальный моменты времени; $A(t)$, $A_h(t)$, $B(t)$ и $C(t)$ — непрерывные матрицы-функции; $x_0(t)$ — непрерывная вектор-функция; $\Phi(t)$ и $\Psi(t)$ — симметричные непрерывные матрицы-функции, такие, что соответствующие им квадратичные формы положительно определены.

Цель управления — доставить показателю γ как можно меньшее значение.

Первый подход

Первый подход к приближенному решению поставленной задачи заключается в ее сведении к задаче управления для системы (1), (2), но уже с дискретизированным показателем качества

$$\gamma_1 = \left(\sum_{i=1}^m \|x(\vartheta_i)\|^2 \Delta_h \right)^{1/2} + \int_{t_0}^{\vartheta} [\langle u(t), \Phi(t)u(t) \rangle - \langle v(t), \Psi(t)v(t) \rangle] dt,$$

где $m \in \mathbb{N}$, $\Delta_h = h/m$ и $\vartheta_i = \vartheta - h + i\Delta_h$, $i = \overline{1, m}$. Решение такой задачи подробно описано в [2] и основано на решении вспомогательной дифференциальной игры для системы уже без запаздывания

$$\dot{z}(t) = B(t)u(t) + C(t)v(t), \quad t_0 < t < \vartheta,$$

$$z \in \mathbb{R}^{nm}, \quad u \in \mathbb{R}^r, \quad v \in \mathbb{R}^s,$$

и терминальным показателем качества

$$\gamma_z = \|z(\vartheta)\| + \int_{t_0}^{\vartheta} [\langle u(t), \Phi(t)u(t) \rangle - \langle v(t), \Psi(t)v(t) \rangle] dt,$$

где $B(t) = (B_1(t), B_2(t), \dots, B_m(t))^T$, $B_i(t) = F(\vartheta_i, t)B(t)$, $i = \overline{1, m}$;

$C(t) = (C_1(t), C_2(t), \dots, C_m(t))^T$, $C_i(t) = F(\vartheta_i, t)C(t)$, $i = \overline{1, m}$; $F(\xi, t)$ — матрица Коши уравнения (1).

Второй подход

Второй подход, опирающийся на результаты работы [3], заключается в дискретизации не только показателя качества (3), но и исходной системы (1). Рассмотрим дифференциальную игру для системы

$$\begin{cases} \dot{y}^{[0]}(t) = A(t)y^{[0]}(t) + A_h(t)y^{[m]}(t) + B(t)\tilde{u}(t) + C(t)\tilde{v}(t), \\ \dot{y}^{[i]}(t) = \frac{(y^{[i-1]}(t) - y^{[i]}(t))}{\Delta_h}, \quad i = \overline{1, m}, \end{cases} \quad (4)$$

$$t_0 < t < \vartheta, \quad y^{[i]} \in \mathbb{R}^n, \quad i = \overline{0, m}, \quad \tilde{u} \in \mathbb{R}^r, \quad \tilde{v} \in \mathbb{R}^s,$$

с согласованным с (2) начальным условием

$$y^{[i]}(t_0) = x_0(t_0 - h + i\Delta_h), \quad i = \overline{0, m}, \quad (5)$$

и соответствующим (3) показателем качества

$$\gamma_2 = \left(\sum_{i=1}^m \|y^{[i]}(\vartheta_i)\|^2 \Delta_h \right)^{\frac{1}{2}} + \int_{t_0}^{\vartheta} [\langle \tilde{u}(t), \Phi(t)\tilde{u}(t) \rangle - \langle \tilde{v}(t), \Psi(t)\tilde{v}(t) \rangle] dt. \quad (6)$$

Методы для эффективного решения этой игры даны, например, в [1]. Для решения исходной задачи управления систему (4) будем использовать в качестве поводья для исходной системы (1): управление $u(t)$ и помеху $\tilde{v}(t)$ будем формировать в дискретной по времени схеме на базе некоторого разбиения $\Delta_k = \{t_j | t_1 = t_0, t_j < t_{j+1}, j = \overline{1, k}, t_{k+1} = \vartheta\}$ промежутка времени управления $[t_0, \vartheta]$ из условий экстремального сдвига:

$$u(t) = u_j \in \operatorname{argmin}_{\|u\| \leq R} [\langle B(t_j)u, x(t_j) - y^{[0]}(t_j) \rangle - \langle u, \Phi(t_j)u \rangle w_1(t_j)],$$

$$\tilde{v}(t) = \tilde{v}_j \in \operatorname{argmax}_{\|\tilde{v}\| \leq R} [\langle C(t_j)\tilde{v}, x(t_j) - y^{[0]}(t_j) \rangle - \langle \tilde{v}, \Psi(t_j)\tilde{v} \rangle w_2(t_j)],$$

$$w_1(t_j) = \int_{t_0}^{t_j} \langle u(t) - \tilde{u}(t), \Phi(t)(u(t) + \tilde{u}(t)) \rangle dt,$$

$$w_2(t_j) = \int_{t_0}^{t_j} \langle \tilde{v}(t) - v(t), \Psi(t)(\tilde{v}(t) + v(t)) \rangle dt,$$

$$R > 0, \quad t \in [t_j, t_{j+1}), \quad j = \overline{1, k}.$$

Согласно [1] и [3], при достаточно больших значениях $R > 0$ и m , такое взаимное прицеливание обеспечит нужную близость движений системы (1) и (4), а также близость интегральных оценок из показателей (3) и (6). Управление $\tilde{u}(t)$ будем выбирать оптимальным в смысле дифференциальной игры (4)–(6) образом.

Заключение

Оба подхода оказываются работоспособными и позволяют решать задачу (1)–(3) для достаточно больших значений размерности фазового вектора n и параметра аппроксимации m . Однако первый подход, судя по численным экспериментам, оказывается более ресурсоемким. Таким образом, в работе приведен пример задачи управления, для которой использование аппроксимаций дифференциальных систем с запаздыванием при помощи систем обыкновенных дифференциальных уравнений высокого порядка в качестве поводурей приводит к эффективному решению.

Литература

1. Красовский Н.Н. Управление динамической системой. М.: Наука, 1985. 516 с.
 2. Лукоянов Н.Ю., Решетова Т.Н. Задачи конфликтного управления функциональными системами высокой размерности // ПИММ. 1998. Т.62, Вып.4. С.586–597.
 3. Плаксин А.Р. Конечномерные поводурей в задачах управления системами с запаздыванием // «Современные проблемы математики»: тезисы Международной (43-й Всероссийской) молодежной школы-конференции. С.163–165. Екатеринбург: ИММ УрО РАН, 2012.
-

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ЗАДАЧИ РЕКОНСТРУКЦИИ ГРАНИЧНЫХ УПРАВЛЕНИЙ В ДИНАМИЧЕСКИХ СИСТЕМАХ ГИПЕРБОЛИЧЕСКОГО ТИПА¹

Грибанова Е.И.

*магистрантка кафедры вычислительной математики УрФУ,
egribanova@list.ru*

Аннотация: Рассматривается обратная задача динамики, состоящая в восстановлении неизвестных граничных управлений, порождающих наблюдаемое движение динамической системы. Динамическая система описывается краевой задачей для уравнения с частными производными гиперболического типа. Исходной информацией для решения обратной задачи служат результаты приближенных измерений текущих фазовых положений наблюдаемого движения системы. Рассматриваемая задача некорректна и для ее решения предлагается воспользоваться методом Тихонова. В работе обоснована не только сходимость регуляризованных приближений в пространствах Лебега, но и кусочно-равномерная сходимость. Приводятся результаты численного моделирования.

Введение

В работе рассматривается задача о восстановлении априори неизвестных граничных управлений, функционирующих в управляемой динамической системе, описываемой краевой задачей для уравнения с частными производными гиперболического типа. Управляющие воздействия заранее неизвестны и должны быть определены в результате наблюдений за системой по приближенным измерениям текущих фазовых положений системы. Известно, что рассматриваемая задача некорректна и ее решение требует привлечения методов регуляризации [1–2].

Для решения задачи предлагается воспользоваться вариационным методом Тихонова, состоящем в минимизации некоторого подходящего функционала невязки на множестве допустимых управлений. Задача будет решаться в статическом варианте, когда для решения задачи будет использоваться вся совокупность результатов измерений, накопленная в течение какого-либо заданного промежутка времени наблюдения за системой.

¹ Работа выполнена по программе Президиума РАН «Фундаментальные проблемы нелинейной динамики в математических и физических науках» при поддержке УрО РАН (проект 12-П-1-1009), поддержана молодежным грантом ИММ УрО РАН «Восстановление управлений в динамических системах» и поддержана РФФИ (проект 11-01-00073).

Стабилизирующие функционалы, содержащие нормы пространств Лебега приводят к довольно грубой аппроксимации. В данной работе показано, что при использовании стабилизатора в виде суммы классической вариации и среднеквадратичной нормы в задачах восстановления управлений можно получить поточечную сходимость, сходимость в пространствах Лебега, сходимость классических вариаций и кусочно-равномерную сходимость. В работе выполнена двухэтапная конечномерная аппроксимация задачи. Проведены вычислительные эксперименты, которые показали, что разработанные методы и алгоритмы могут применяться для восстановления искомого управления и его тонкой структуры. Работа продолжает исследования [3–5].

Постановка задачи

Рассматривается управляемая динамическая система

$$y_u = Ly + f(t, x), \quad (t, x) \in Q = T \times \Omega, \quad (1)$$

$$y(t_0, x) = y_0(x), \quad y_t(t_0, x) = y_1(x), \quad x \in \Omega, \quad (2)$$

$$\partial y / \partial N + \sigma y = g(x)u(t), \quad t \in T, \quad x \in \partial\Omega, \quad (3)$$

где $\sigma = \text{const} \geq 0$; y_0 — начальное положение системы; y_1 — начальная скорость системы; f, g — заданные функции; $\partial y / \partial N$ — внешняя кономальная производная, соответствующая оператору L , $u = (u_1, \dots, u_m)$ — векторная функция управляющего воздействия;

$$Ly = \sum_{i,j=1}^n \frac{\partial}{\partial x_i} \left(a_{ij}(x) \frac{\partial y}{\partial x_j} \right) - a(x)y, \quad a_{ij} = a_{ji}, \quad a \geq a_0 = \text{const} > 0.$$

Состояние системы в момент времени t из $T = [t, \vartheta]$ характеризуется парой функций $z[t] = (y[t], y_t[t]) = (y(t, \cdot), y_t(t, \cdot))$, определенных в некоторой ограниченной в R^n области Ω с кусочно-гладкой границей.

Пусть $U = \{u \in L_2(T; R^m) : u(t) \in PC R^m \text{ n.в. } t \in T\}$ — множество всех допустимых управлений в задаче.

За движением системы ведется наблюдение в течение промежутка времени T и в текущие моменты времени $t \in T$ приближенно измеряются состояния системы, причем результаты этих измерений $z^\delta[t] = (y^\delta[t], y_t^\delta[t])$ удовлетворяют условию:

$$H(z_\delta, z) = \int_T \|y^\delta[t] - y(t, \cdot; y)\|_{L_2(\Omega)}^2 + \|y_t^\delta[t] - y_t(t, \cdot; y)\|_{W_2^1(\Omega)}^2 dt \leq \delta^2,$$

где δ — числовой параметр, характеризующий точность измерений, $0 \leq \delta \leq \delta_0$.

Задача восстановления состоит в том, чтобы по результатам приближенных измерений наблюдаемого движения системы приближенно определить ту реализацию управления на динамическую систему, которая соответствует результатам наблюдений. При этом результат восстановления $u_\delta = u_\delta(t), t \in T$, должен быть тем точнее, чем меньше ошибки измерений $\|u_\delta(t) - u(t)\|_{L_2(T; R^m)} \rightarrow 0, \delta \rightarrow 0$.

Пусть

$f \in L_2(Q), g \in L_2^m(\partial\Omega), y_0 \in L_2(\Omega), y_1 \in W_2^1(\Omega)^*$; $Z = \{z[\cdot, u]: u \in U\}$ — множество всех возможных движений системы (1) – (3), отвечающих всем возможным управлениям $u \in U$; P — выпуклый компакт в R^m ;

$U(z) = \{u \in U: z[\cdot, u] = z\}$ — множество всех допустимых управлений, порождающих движение $z, Y_\delta(z) = \{z_\delta \in Y: H(z_\delta, z) \leq \delta^2\}$ — множество всех возможных измерений этого движения, $Y = L_2(T; L_2(\Omega)) \times L_2(T; W_2^1(\Omega)^*)$.

Искомый алгоритм отождествим с семейством отображений (методов): $D = \{D_\delta: 0 \leq \delta \leq \delta_0\}, D_\delta: Y \rightarrow E = L_2(T; R^m)$.

Требуется построить алгоритм D , который на любом наблюдаемом движении системы $z \in Z$ обладает регуляризующим свойством $r_\delta(z) \rightarrow 0, \delta \rightarrow 0$, где $r_\delta(z) = \sup\{\rho[D_\delta(z_\delta), U(z)]: z_\delta \in Y_\delta(z)\}, \rho[D_\delta(z_\delta), U(z)] = \min\{\|D_\delta(z_\delta) - v\|_E: v \in U(z)\}$.

Все рассматриваемые числовые величины и пространства считаются вещественными, измеримость и интегрируемость понимаются по Лебегу, определения используемых пространств имеются, например, в [6].

Решение задачи

Задача восстановления управлений некорректна. Для её решения воспользуемся методом регуляризации Тихонова.

Пусть W есть банахово пространство функций $T \rightarrow R^m$ ограниченной полной вариации на T . Введем следующие обозначения: $\alpha = \text{const} > 0$,

$$F_\alpha = F_\alpha(z, u) = H(y[\cdot; u], y[t]) + \alpha \Lambda(u), \quad \Lambda(u) = \|u\|_E^2 + V[u], \quad (4)$$

$$F_\alpha^*(z) = \min\{F_\alpha(z, u): u \in U_W\}, \quad U_W = U \cap W, \quad (5)$$

$$U_\alpha^*(z) = \{u \in U_W: F_\alpha(z, u) = F_\alpha^*(z)\}, \quad (6)$$

$V[u]$ — полная вариация функции $u \rightarrow R^m$.

Построим алгоритм, решающий задачу восстановления. Для $\delta \in [0, \delta_0]$ и $z = (y, y_t) \in Y$ определим реализацию (значение) метода $D_\delta(z)$ по правилу

$$D_\delta(z) = v \in U: F_\alpha^*(z) \leq F_\alpha(z) \leq F_\alpha^*(z) + \varepsilon, \quad (7)$$

где $\varepsilon \geq 0$ — параметр, характеризующий точность по функционалу решения экстремальной задачи (5). Величины α и ε будут являться параметрами метода, они будут выбираться в зависимости от величины погрешности измерений δ . Доказана теорема [4, 5]: построенный алгоритм при соответствующих условиях согласования параметров регуляризации $\alpha(\delta)$ и $\varepsilon(\delta)$ решает задачу восстановления, причем результаты восстановления сходятся при $\delta \rightarrow 0$ к искомому управлению кусочно-равномерно на T .

Аппроксимация задачи

Для аппроксимации задачи воспользуемся методом разделения переменных. Решение краевой задачи (1) – (3) представим в виде

$$y = y(t, x; u) = \sum_{i=1}^{\infty} y_i(t) \omega_i(x), \quad t \in T, \quad x \in \Omega; \quad \ddot{y}_i = -\lambda_i y_i(t) + f_i(t) + g^i u(t),$$

$$y_i(t_0) = y_{0i} = \langle y_0, \omega_i \rangle_{L_2(\Omega)}, \quad \dot{y}_i(t_0) = y_{1i} = \langle y_1, \omega_i \rangle, \quad i \in N; \quad \{\lambda_i, \omega_i; i \in N\}$$

— решение в $W_2^1(\Omega)$ спектральной задачи $L\omega = -\lambda\omega$ в Ω , $\frac{\partial \omega}{\partial N} + \sigma\omega = 0$ на $\partial\Omega$. Фиксируем какое-нибудь $p \in N$ и рассмотрим экстремальную задачу

$$\Phi_\alpha^{(p)}(z) = \min \{F_\alpha^{(p)}(z, u) : u \in U_W\}, \quad F_\alpha^{(p)}(z, u) = J_1^{(p)}(z, u) + \alpha \Lambda(u), \quad (8)$$

$$J_1^{(p)}(z, u) = \sum_{i=1}^p \int_T (y_i[t] - y_i(t; u))^2 dt + \sum_{i=1}^p (\lambda_i)^{-1} \int_T (\dot{y}_i[t] - \dot{y}_i(t; u))^2 dt$$

$$y_i[t] = \langle y[t], \omega_i \rangle_{L_2(\Omega)}, \quad \dot{y}_i[t] = \langle \dot{y}[t], \omega_i \rangle, \quad z = (y[\cdot], y_1[\cdot]) \in Y.$$

Экстремальная задача (8) имеет единственное решение $u_\alpha^{(p)}(z) \in U_W$, всякая минимизирующая последовательность в задаче (8) сходится сильно в E к элементу $u_\alpha^{(p)}(z)$.

Рассмотрим следующий алгоритм для решения исходной задачи реконструкции: $D = \{D_\delta^{(p)} : 0 \leq \delta \leq \delta_0, p \in N\}$, $D_\delta^{(p)} : Y \rightarrow E$. Для любых $\delta \in [0, \delta_0]$, $p \in N$, $z \in Y$ определим реализацию метода

$$D_\delta^{(p)}(z) = v \in U_W : \Phi_\delta^{(p)}(z) \leq P_\delta^{(p)}(z, v) \leq \Phi_\delta^{(p)}(z) + \varepsilon, \quad (9)$$

где ε — неотрицательный параметр, характеризующий точность по функционалу решения экстремальной задачи (8).

Доказано утверждение [4, 5]: семейство алгоритмов D при соответствующем выборе параметров регуляризации $\alpha(\delta)$, $\varepsilon(\delta)$, $p(\delta)$ является регуляризирующим, причем результаты восстановления сходятся при $\delta \rightarrow 0$ к искомому управлению кусочно-равномерно на T .

Численное моделирование

Проведем численное моделирование задачи восстановления управления в системе

$$\begin{aligned} y_u &= a^2 y_{xx}, \quad (t, x) \in T \times \Omega, \\ y(0, x) &= y_0(x), \quad y_t(0, x) = y_1(x), \quad x \in \Omega = (0, l), \\ y_x(t, 0) &= u(t), \quad y_x(t, l) = 0, \quad t \in T = [0, \vartheta]. \end{aligned}$$

Пусть множество геометрических ограничений на управление $P = [\mu_1, \mu_2] \subset R$, приближенное измерение положений динамической системы моделируется соотношением $y^\delta(t, x) = y(t, x; u) + \delta \xi(t, x)$, $y_i^\delta(t, x) = y_i(t, x; u) + \delta \eta(t, x)$, причем $\int_T \|\xi(t, \cdot)\|_{L_2(\Omega)}^2 dt \leq 0$, $\int_T \|\eta(t, \cdot)\|_{W_1^1(\Omega)}^2 dt \leq 1$.

В данном случае $\lambda_0 = 0$, $\omega_0(x) = \sqrt{1/l}$, $g^0 = \sqrt{1/l}$, $\lambda_i = (\pi i/l)^2$, $\omega_i(x) = \sqrt{2/l} \cdot \cos(\sqrt{\lambda_i} \cdot x)$, $g^i = \sqrt{2/l}$, $i = 1, 2, 3, \dots$. Погрешность измерений для дискретной задачи будем моделировать соотношениями

$$\begin{aligned} \xi_{ih} &= (\xi_{ih}(t_0), \dots, \xi_{ih}(t_m)), \quad \eta_{ih} = (\eta_{ih}(t_0), \dots, \eta_{ih}(t_m)), \quad i = \overline{0, p}; \\ \xi_{ih}(t_k) &= \chi_i \sin(\beta_i t_k), \quad \eta_{ih}(t_k) = \varphi_i \sin(\tau_i t_k), \quad k = \overline{0, m}, \quad i = \overline{0, p}, \end{aligned}$$

где $\chi_i, \beta_i, \varphi_i, \tau_i$ — вещественные числа; $\chi_0^2 + \dots + \chi_p^2 + \varphi_0^2 + \dots + \varphi_p^2 \leq 1$.

Численные эксперименты проводились при следующих параметрах задачи: $a = 1$, $\vartheta = 1$, $l = 1$, $\mu_1 = -2$, $\mu_2 = 2$, $y_0 = 0$, $y_1 = 0$, $\chi_i = \varphi_i = 1/\sqrt{2p}$, $\beta_i = \tau_i = 2$, $i = 0, \dots, p$, $p = 10$. В качестве модельного восстанавливаемого управления была выбрана функция: $u = u(t) = -1$ при $t \in [0, 0.25]$, $u(t) = 8t - 3$ при $t \in (0.25, 0.5)$, $u(t) = -1.5$ при $t \in [0.5, 0.75]$, $u(t) = 2$ при $t \in [0.75, 1]$, (разрывное управление).

Величины в методе проекции субградиента γ_k выбирались по формулам: $\gamma_0 = 1$, $\gamma_k = 1/\sqrt{k}$, $k = 1, 2, \dots$. Начальной функцией в методе проекции субградиента служила нулевая сеточная функция. Зависимость параметра $\varepsilon = \varepsilon(\delta)$ точности минимизации от погрешности δ в измерении состояния динамической системы напрямую не контролировалась, точность решения задачи определялась выбором количества итераций M в методе проекции

субградиента и величиной шага h , характеризующего степень дискретизации задачи. В расчетах полагалось: $h=0.004$, $\alpha=10^{-4}$, $M=500$.

На рис. 1 сплошной линией показано модельное восстанавливаемое управление, пунктирная линия — результат восстановления при $\delta=0.1$, линия с точками — результат восстановления при $\delta=0.005$. На рис. 2 показана погрешность восстановления при $\delta=0.005$ (приводится график разности модельного управления и его реконструкции). На рисунках горизонтальная ось — ось времени, вертикальная ось — ось значений управления и погрешности восстановления соответственно. В табл. 1 приводятся результаты восстановления управления при варьировании параметра δ , относительная погрешность указана в процентах.

Приложение

Погрешность δ	Невязка	Относительная погрешность
0.5	0.4886	35.5082
0.1	0.3404	24.7384
0.05	0.3061	22.2442
0.01	0.3051	22.1708
0.005	0.3050	22.1660

Таблица 1. Результаты восстановления разрывного управления

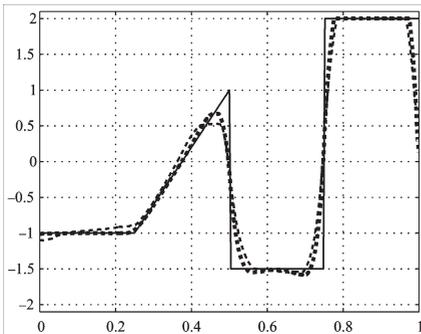


Рис. 1. Восстановление управления

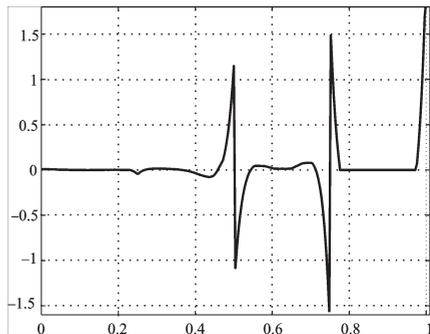


Рис. 2. Погрешность восстановления

Заключение

В данной работе рассмотрена задача реконструкции граничных управлений, функционирующих в гиперболических системах. Исходной информацией служили приближенные измерения состояний системы. Задача решена методом регуляризации Тихонова. В работе рассмотрен подходящий функционал Тихонова F_{α} , изучена задача минимизации этого функционала.

Задача минимизации разрешима и имеет единственное решение. Выполнена двухэтапная конечномерная аппроксимация задачи. Проведены вычислительные эксперименты.

Литература

1. *Тихонов А.Н.* Методы решения некорректных задач /Тихонов А.Н., Арсенин В.Я./ М.: Наука. — 1979. — С. 288.
 2. *Лаврентьев М.М.* Некорректные задачи математической физики и анализа / Лаврентьев М.М., Романов В.Г., Шишатский С.П./ М.: Наука. — 1980. — С. 288.
 3. *Короткий М.А.* Восстановление управлений и параметров методом Тихонова с негладкими стабилизаторами / Короткий М. А / Изв. вузов. Математика. — 2009. — №2. — С. 76–82.
 4. *Короткий А.И.* Восстановление управлений в гиперболических системах методом Тихонова с негладкими стабилизаторами / Короткий А.И., Грибанова Е.И./ Тр. Ин-та математики и механики УрО РАН. — 2010. — Т. 17. — №1. — С. 99–108.
 5. *Короткий А.И.* Восстановление граничных управлений в гиперболических системах /Короткий А.И., Грибанова Е.И./ Тр. Ин-та математики и механики УрО РАН. — 2012. — Т. 18. — №2. — С. 154–169.
 6. *Ладыженская О.А.* Краевые задачи математической физики / Ладыженская О. А./ М.: Наука. — 1973. — С. 408.
-

ДВУМЕРНЫЕ РАСЧЕТЫ СЖАТИЯ И ГОРЕНИЯ МИКРОМИШЕНЕЙ ДЛЯ ЛТС С НЕПРЯМЫМ ВОЗДЕЙСТВИЕМ

*Соколов Л.В. (sokolovlv@vniitf), Лыков В. А.,
Чижков М.Н., Шушлебин А.Н.
(РФЯЦ-ВНИИТФ, г.Снежинск)*

Аннотация: Проведены двумерные расчеты сжатия и термоядерного горения сферических мишеней для ЛТС. Представлены результаты исследований развития возмущений как на стадии сжатия, так и термоядерного горения мишеней с прямым воздействием.

Введение

Технологические процессы изготовления лазерных мишеней на сегодняшний день несовершенны — относительные геометрические отклонения от заданных размеров бывают очень велики. Сжатие и зажигание мишеней не одномерны, поскольку осуществляются с помощью конечного числа лазерных «пучков». В результате задуманные одномерными, сферические мишени на самом деле таковыми не являются.

Цель описываемого моделирования — оценка влияния погрешности изготовления мишеней и неоднородности их зажигания на параметры их работы.

На этапе моделирования сжатия решается система двумерных уравнений газовой динамики с нелинейной теплопроводностью в трехтемпературном приближении, которая включает в себя три уравнения энергии для электронов, ионов и фотонов.

Для решения системы уравнений применяется эйлерово-лагранжева методика, в которой используется принцип расщепления по физическим процессам и координатным направлениям и дифференциальные уравнения аппроксимируются конечноразностными. Для решения расщепленных и линеаризованных по Ньютону разностных уравнений энергии на мелких шагах используется метод матричной прогонки при определении электронной, ионной и фотонной температур.

На этапе моделирования горения дополнительно моделируются кинетика термоядерных реакций, перенос энергии альфа-частицами и нейтронами в двумерной геометрии с осевой симметрией.

Цель моделирования — оценка влияния погрешности изготовления мишеней и неоднородности их зажигания на параметры их работы. Проведены двумерные расчеты сжатия и термоядерного горения сферических мишеней для ЛТС. Представлены результаты исследований развития возмущений как на стадии сжатия, так и термоядерного горения мишеней с прямым воздействием.

1. Система уравнений. Разностная сетка.

Расчет мишеней проводится в два этапа. Первый этап считается без учета термоядерных реакций в газе, до достижения заданных температур.

На первом этапе решается система уравнений двумерного движения теплопроводного газа в трехтемпературном приближении, которая представляет собой следствие выраженных в дифференциальной форме законов сохранения массы, импульса и энергии (электронов, ионов и фотонов) [1, 2]

Система уравнений первоначально записывается для цилиндрической системы координат. Полученная система уравнений после преобразований переводится в смешанную систему координат, где в качестве эйлеровой координаты выбираются прямые линии, непересекающиеся в области решения. Лагранжевые линии, как правило, совпадают с границами раздела веществ.

На момент времени $t=0$ задаются начальные условия, описывающие исходное состояние рассматриваемого объекта.

Взаимодействие объекта с внешней средой описывается условиями на его границе по газодинамике и теплопроводности, аналогично [3, 4]. Различие в том, что для теплопроводности вместо одной комбинации теплового потока и температуры берется три: по каждой компоненте (электроны, ионы и фотоны) в отдельности. Для газодинамики граничное давление $P^{ГР} = p_e^{ГР} + p_i^{ГР} + p_f^{ГР}$.

Система уравнений расщепляется на два основных физических процесса (движение газа и распространение тепла) и два координатных направления и аппроксимируется конечно-разностными неявными уравнениями [3, 4]. После расщепления по направлениям системы уравнений для процесса построение решения сводится к расчету совокупности двух «одномерных» задач. Обменные члены учитываются в расщепленных уравнениях с весами. Квадратичная вязкость отнесена к ионам, в которых сосредоточена масса вещества.

При проведении расчетов включение процессов осуществляется последовательно: сначала — движение газа, а затем — распространение тепла. При расчете движения газа вычисляются значения R, v, u, ρ для $n+1$ временного слоя, а температура в расчете берется с n временного слоя. При расчете уравнений энергии используются уже рассчитанные параметры течения газа.

На втором этапе к рассмотренной системе добавляются уравнения кинетики, энерговыделения, переноса нейтронов и α -частиц.

Для системы уравнений газодинамики, теплопроводности, кинетики, энерговыделения, переноса нейтронов и α -частиц, к которой присоединяются уравнения состояния, ставится смешанная задача Коши с соответствующими граничными условиями.

2. Мишень № 1 со стеклянной оболочкой

Мишень представляет собой стеклянную оболочку (SiO_2) внешним диаметром 2.15 мм, толщиной 0.075 мм с плотностью 2.5 г/см³, в которой на

внутренней поверхности стеклянной оболочки наморожен слой ДТ-льда толщиной 0.02 мм с плотностью 0.2 г/см³, а полость заполнена остаточным ДТ-газом с плотностью 10⁻⁴ г/см³. Полагалось, что мишень облучается чернотельным излучением, температура которого $T_f^*(t)$ нарастает линейно во времени до значения 300 эВ за время 10 нсек.

Влияние неоднородностей изготовления мишени оценивалось посредством задания в расчетах двумерных возмущений внешней поверхности стеклянной оболочки в виде 6, 12, 24 и 48 гармоник (варианты 1,2,3,4).

$$R = R_0 + a_0 \cos(k\theta),$$

где: R_0 — радиус границы, a_0 — амплитуда возмущения, k — номер гармоники.

Также проведены расчеты для неоднородного облучения стеклянной оболочки в виде 1 и 6 гармоник (варианты 5, 6), в которых

$$T_f^{*p} = T_f^*(t)(1 + b_0 \cos(k\theta)),$$

где: b_0 — амплитуда возмущения, k — номер гармоники.

На рисунках 1 и 2 показано состояние разностной сетки в центре мишени на момент максимального сжатия, полученное в расчетах без учета термоядерных реакций для неоднородностей изготовления и облучения. Размеры по осям x и y на двумерных графиках даны в микрометрах. Линии сетки захватывают область ДТ-топлива и неиспаренную часть оболочки. На рисунке 2 видно, что деформации в расчетах с неоднородностью облучения достигли критических значений, ДТ-топливо разрывается на части и никакого горения не будет.

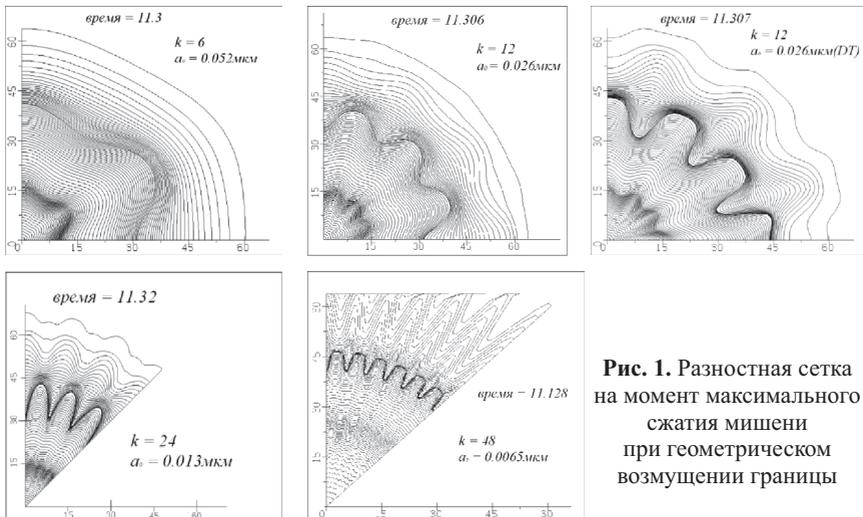


Рис. 1. Разностная сетка на момент максимального сжатия мишени при геометрическом возмущении границы

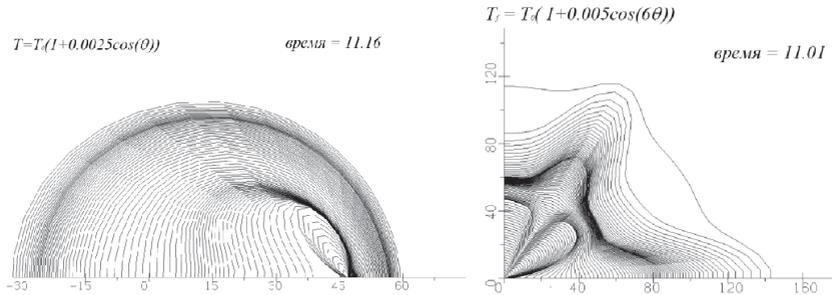


Рис. 2. Разностная сетка на момент близкий к максимальному сжатию мишени в случае неоднородного облучения внешней границы

Расчеты с учетом термоядерных реакций были начаты со времени $t = 10.49$ нсек и проведены для возмущений внешней границы мишени с гармониками $k = 0, 6, 12$.

В таблице 1 приведены достигнутые максимальные значения средней температуры электронов, ионов и фотонов и плотности, выгорание трития в слоях ДТ-топлива мишени. В проведенных двумерных расчетах с возмущениями наибольшее снижение выгорания топлива по сравнению с одномерными расчетами получено равным $\approx 5.3\%$. при возмущении внешней границы с $k = 12$ $a_0 = 0.026$ мкм.

k	0			6 ($a_0 = 0.052$)			12 ($a_0 = 0.026$)		
	газ	ДТ-1	ДТ-2	газ	ДТ-1	ДТ-2	газ	ДТ-1	ДТ-2
$\{T_i\}^{\max}$, кэВ	38.76	37.46	26.37	38.89	37.5	25.8	38.6	37.2	25.12
$\{T_e\}^{\max}$, кэВ	25.24	24.66	18.35	24.8	24.18	17.7	24.6	24.0	17.15
$\{T_f\}^{\max}$, кэВ	5.48	5.48	5.47	5.48	5.48	5.47	5.49	5.49	5.47
$\{\rho_{DT}\}^{\max}$, г/см ³	90	99.1	233.3	90.8	99.5	234	90.8	99.6	232.9
S , %	30.59	29.34	22.02	30.54	29.21	21.42	30.47	29.06	20.86
$S/S_{k=0}$	1	1	1	0.998	0.995	0.972	0.996	0.99	0.947

Таблица 1. Результаты расчета энерговыделения

Двумерные расчеты показали, что нарушение сферичности сжатия мишени приводит к снижению энерговыделения по отношению к расчетам без возмущений.

3. Мишень №2 с бериллиевой оболочкой

Мишень разработана для термоядерного зажигания на установке «ИС-КРА-6» и подобна мишени Лос-Аламосской лаборатории для NIF [5]. Параметры мишени приведены в таблице 2.

№ области	Радиус (мм)	Вещество	Плотность (г/см ³)
1	0.435	D _{0,5} T _{0,5} — газ	0.0003
2	0.475	D _{0,5} T _{0,5} — лед	0.25
3	0.5625	Be _{0,98} Cu _{0,02}	1.9

Таблица 2. Параметры областей мишени

В таблице 3 приведены результаты расчетов мишени с геометрическим возмущением поверхности мишени в виде 12-ой гармоники при различных значениях амплитуды возмущения. С ростом амплитуды возмущения наблюдается снижение выгорания ДТ-топлива в мишени.

	вариант 1	вариант 2	вариант 3
k	0	12	12
a_0 (нм)	0	13	26
$\{\rho_{DT}\}^{\max}$ (г/см ³)	739	711	690
$\{T_{ij}\}^{\max}$ (кэВ)	36,1	30,2	26,0
выгорание T (%)	19,4	16,7	14,4

Таблица 3. Результаты расчетов с возмущением внешней границы оболочки

По результатам расчетов с возмущением внешней поверхности мишени по отношению к соответствующему одномерному расчету наблюдается снижение выгорания на 26 %.

В таблице 4 приведены результаты расчетов мишени с возмущением температуры излучения от времени на внешней границе для 4-ой гармоники при разных значениях амплитуды возмущения.

С ростом амплитуды возмущения наблюдается значительное снижение выгорания ДТ-топлива в мишени. При неоднородности в падающем потоке излучения с $b_0 = -0.0025$ (расчет 3) термоядерный выход составил 18 % от этой величины при отсутствии возмущений.

В расчете 4 была задана та же амплитуда возмущения температуры, что и в расчете 2. Фаза возмущения была сдвинута на π . Это привело к тому, что на оси симметрии при $\theta=0$ (θ — азимутальный угол) температура излуече-

ния была не минимальной, а максимальной. Из таблицы 4 видно, что смена фазы возмущения температуры не привела к существенным изменениям: в расчетах 2 и 4 получены близкие параметры сжатия и горения.

	вариант 1	вариант 2	вариант 3	вариант 4
k	0	4	4	4
b_0	0	-0.00125	-0.0025	-0.00125
$\{\rho_{DT}\}^{\max}$ (г/см ³)	739	658	563	650
$\{T_j\}^{\max}$ (кэВ)	36.1	25.4	8.5	23.4
выгорание T (%)	19.4	14.4	3.5	13.2

Таблица 4. Результаты расчета мишени с возмущением температуры на границе

Результаты проведенных расчетов использовались для оценки необходимой точности изготовления оболочек мишеней и требуемой однородности облучения.

4. Расчеты некриогенной мишени с двумя оболочками

В РФЯЦ-ВНИИТФ была исследована возможность термоядерного зажигания двухоболочечной мишени, подобной мишени Ливерморской лаборатории, на лазерной установке «ИСКРА-6».

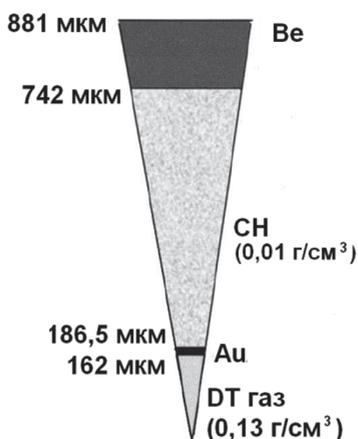


Рис. 3. Двухоболочечная мишень для установки «ИСКРА-6»

В таблице 5 представлены результаты расчетов некриогенной двухоболочечной мишени для лазерной установки «Искра-6». Приведены результаты одномерного расчета и расчетов с геометрическим возмущением внешней границы мишени в виде 12-ой гармоники для различных значений амплитуды возмущения ($R(\theta) = R_0 + \delta_0 \cos(k\theta)$). Также был проведен расчет, в котором одновременно задавалась разнотолщинность внешней оболочки (12-я гармоника), асимметрия облучения (4-я гармоника) и сдвиг центров оболочек (1-я гармоника). В проведенном расчете ведущей оказалась асимметрия облучения мишени.

Вариант	\square	δ_0 (нм)	$\langle \rho_{DT} \rangle_{\max} (\text{г/см}^3)$	$\{T_j\}^{\max}$ (кэВ)	\square_T (%)
1	0	0	250 (350)	18,8 (32)	24,2(36)
2	12	26	262 (370)	18,5 (29)	22,5 (34)
3	12	52	283	10,5	10,7
4	12	78	308	6,8	5,8
5 (*)	12	26	254	18.2	18.1

(*) в расчете дополнительно задавалось возмущение температуры рентгеновского излучения на внешней границе по формуле $T_j(\theta, t) = T_j^*(t)(1 - a_0 \cos(k\theta))$ с амплитудой $a_0 = 2,5 \times 10^{-3}$ и $k=4$, а также сдвиг центров оболочек $\Delta = 1$ мкм.

Таблица 5. Результаты расчетов некриогенной двухоболочечной мишени для лазерной установки «Искра-6» с геометрическим возмущением внешней границы бериллиевой оболочки.

Заключение

Проведены двумерные расчеты сжатия и термоядерного горения сферических мишеней для ЛТС. Представлены результаты исследований развития возмущений как на стадии сжатия, так и термоядерного горения мишеней с непрямым воздействием.

Результаты проведенных расчетов использовались для оценки необходимой точности изготовления оболочек мишеней и требуемой однородности облучения.

Литература

1. *Fraley C.S., Linnebur E.Y., Mason R.Y., Morse R.I.* «Thermonuclear burn characteristics of compressed deuterium — tritium microspheres.» *Phys.Fluids*, v.17, N 2, 1974.
2. *Барышева Н.М., Зуев А.И., Карлыханов Н.Г., Лыков В.А., Черняков В.Я.* «Неявная схема для численного моделирования физических процессов в лазерной плазме.» *Журнал вычис. мат. и мат. физики*, т. 22, N 2, 1982 г.

3. *Неуважаев В.Е., Фролов В.Д., Яненко Н.Н.* «О применении метода расщепления для численного расчета движения теплопроводного газа в криволинейных координатах.» Известия СО АН СССР, N 8, вып. 2, 1967.
 4. *Неуважаев В.Е., Фролов В.Д., Яненко Н.Н.* «Уравнения движения теплопроводного газа смешанных эйлерово — лагранжевых координатах.», АН СССР СО журнал ЧМСС т 3 N 1, 1972.
 5. *Wilson D.C. and Krauser W.J.* Laser Interaction with Matter. Proceedings of 23rd European Conference. IOP Conference Series Number 140. IOP Publishing, Bristol and Philadelphia, p. 459–462, 1995.
-

Методы хранения, поиска и анализа информации



**Новиков
Борис Асенович**

д.ф.-м.н.
профессор кафедры информатики СПбГУ

АДАПТИВНЫЙ ПРИБЛИЖЕННЫЙ АЛГОРИТМ СОЕДИНЕНИЯ

Долматова О. А.

студентка кафедры информатики СПбГУ, oxana.dolmatova@gmail.com

Введение

В этой работе мы рассматриваем выполнение декларативных запросов в распределенной среде или среде с различной интенсивностью данных. В таких случаях желательно иметь возможность выполнять запросы за предсказуемое время и, возможно, с неточным результатом из-за отсутствия достаточного количества ресурсов или временных ограничений. Общая структура исполнения сложных декларативных запросов описана в [5].

В контексте сложных запросов, соединение является одной из наиболее важных и интенсивно используемых операций. Таким образом, цель этой работы представить адаптивный приближенный алгоритм выполнения соединения с точным предикатом.

В традиционных базах данных существует три стандартных алгоритма выполнения соединения, и все они созданы для получения точного ответа. Однако наш алгоритм приближенной природы. В нашем случае термин «приближенный» означает неполноту результата. То есть какое-то количество кортежей, удовлетворяющих предикату соединения, может не попасть в ответ. Но все пары, что входят в ответ, точно удовлетворяют предикату. Это значит, что качество результата зависит от выделенных на исполнение операции ресурсов.

Описание нового алгоритма

В новом алгоритме выбор стратегии выполнения запроса напрямую зависит от свойств входных данных и количества выделенных ресурсов.

Мы предполагаем, что имеем ограниченное количество внутренней памяти и определенное количество ресурсов, и, возможно, большее количество данных. Так как входные данные представлены в виде потоков, то повторный доступ невозможен, и мы пытаемся найти как можно больше кортежей за минимальное время. Также, в какой-то момент необходимо будет избавляться от прочитанных и обработанных объектов, чтобы освободить память для новых.

Если оба потока отсортированы, тогда новый алгоритм работает как не блокирующее соединение слиянием сортированных списков. Алгоритм получает количество объектов у модели стоимости, которое может прочитать из правого и левого потоков, затем читает и делает соединение.

Если только один из входов отсортирован, то поведение алгоритма подобно смеси алгоритма соединения слиянием отсортированных списков (sort-merge join) и алгоритма соединения вложенными циклами (nested loop join).

В этом случае есть несколько различных вариантов поведения:

- Отсортированный вход помещается в память целиком и у нас есть достаточное количество времени для обработки данных. Тогда алгоритм считывает весь маленький отсортированный поток, а в оставшуюся свободную часть памяти загружает блок из несортированного входа, сортирует его и делает соединение слиянием отсортированных списков. После это считывает следующий блок и т. д., столько раз, сколько рассчитывает модель стоимости.
- Отсортированный вход помещается в память, но ресурсов не достаточно. Алгоритм загружает часть объектов из первого входа, часть из второго, сортирует последние и делает соединение слиянием отсортированных списков этих двух множеств. Все параметры рассчитывает модель стоимости.
- Оба потока не помещаются в память. Алгоритм читает часть первого входа и часть второго, сортирует второй и делает соединение слиянием отсортированных списков, после чего заменяет либо часть первого входа, либо второго на новую.

Если оба потока несортированные, то поведение схоже. Отличие заключается в количестве читаемых объектов и в сортировке сразу двух множеств.

Модель стоимости

Наша модель стоимости является расширенной по сравнению со стандартными моделями и зависит от четырех основных параметров: внутренней памяти, размеров входных потоков, упорядоченности объектов и количества выделенных ресурсов. Так как алгоритм приближенный, так же желательно определять предполагаемое качество результата.

Операция модели стоимости определена так:

$$Cost(data_order, source_in, internal_memory, resources) = \\ = (quality, objects_out),$$

где:

- *cost* функция стоимости, соединяет входные и выходные параметры;
- *data_order* состоит из *left_order* и *right_order*, которые показывают упорядоченность в первом и втором потоках соответственно;
- *source_in* состоит из *left_source* и *right_source* которые показывают Количество объектов в первом и втором потоке соответственно;
- *internal_memory* это размер внутренней памяти, который может быть использован алгоритмом;
- *resources* количество времени, выделенного на выполнение запроса (в миллисекундах);

- *quality* качество результата, соответствует полноте.
- *objects_out* состоит из *left_objects*, *right_objects*, *inner_count* и *outer_count*. Первые два параметра обозначают размер правого и левого блоков объектов. Последние два количество внешних и внутренних циклов для алгоритма.

Так же нам нужно ввести некоторые системные параметры:

- *T_f* время, нужное для чтения одного объекта из потока;
- *T_s* время, отвечающее за сортировку.

Так как размер внутренней памяти маленький относительно общего случая входных данных, то время сортировки можно ограничить сверху линейной зависимостью по отношению к числу объектов.

Для вычисления параметров, описанных выше, были проведены эксперименты для разного количества входных объектов. Затем, с помощью регрессии получены усредненные значения параметров.

Не умаляя общности можно считать, что *left_source* меньше, чем *right_source*.

Из-за ограничений по размеру работы, рассмотрим только один из случаев в модели стоимости. Пусть два входных потока не отсортированные, и не помещаются в память, тогда:

$$\begin{aligned}
 left_objects &:= \min(left_source, internal_memory * 0.9); \\
 right_objects &:= \min(right_source, internal_memory * 0.1); \\
 outer_count &:= (\min(resources / (T_f + T_s) * (2 * left_objects), \\
 &\quad left_source / left_objects)); \\
 inner_count &:= (\min((resources - left_objects * (T_f + T_s) * \\
 &\quad outer_count) / (right_objects * (T_f + T_s) * \\
 &\quad outer_count, right_source / (right_objects * outer_count)));
 \end{aligned}$$

Качество

Следуя определениям в [2,6], можно рассматривать абсолютное и относительное качество.

В нашей работе мы будем рассматривать только относительное качество. Оно показывает, насколько получившийся результат приближенного выполнения соотносится с результатом точного выполнения. В нашем случае это соотношение между числом возвращенных кортежей и числом всех кортежей, удовлетворяющих предикату. Следовательно, относительное качество лежит в промежутке между нулем и единицей. Однако, очевидно, что в некоторых случаях невозможно получить качество, равное единице даже при неограниченном времени из-за ограниченности внутренней памяти.

Эксперименты

Мы провели эксперименты, оценивающие точность, как построенной модели стоимости, так и полученного результата.

Так же мы выявляли зависимости количества найденных пар от того, как устроены входящие данные. Такие эксперименты помогают наглядно показать свойства алгоритма.

Экспериментальный прототип был однопоточным и выполнен на языке PL/SQL.

Все эксперименты проводились на двухъядерном Intel Pentium Dual 1,6 GHz CPU и 4GB RAM. Так же мы предположили, что внутренняя память может содержать не более 5 000 объектов, и что входные данные равномерно распределены. Таким образом, входные данные были автоматически сгенерированы с равномерным распределением.

График на рисунке 1 иллюстрирует точность модели стоимости. Шкала ординат обозначает время в миллисекундах (ожидаемое и реальное), а абсцисс количество объектов в первом и втором входе.

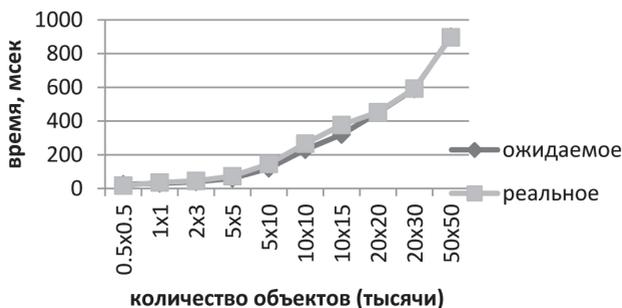


Рис. 1. Выполнение запроса с одним сортированным входом

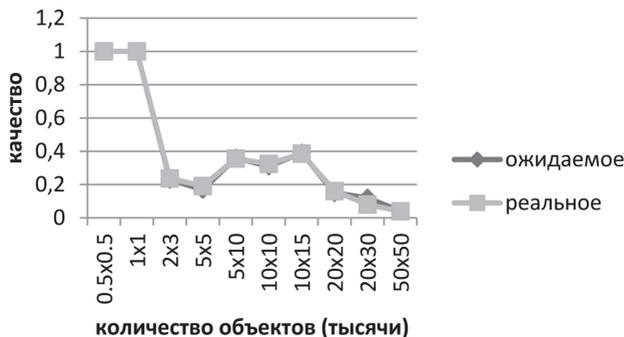


Рис. 2. Качество результата с одним сортированным входом

Рис. 1 демонстрирует выполнение запроса только с одним сортированным входом. Здесь можно заметить нелинейную зависимость между затрачиваемым временем и количеством объектов на входе. Это объясняется тем, что в первых трех случаях потоки целиком могут поместиться в память, а, значит, алгоритм наиболее эффективен, а в остальных случаях нелинейно возрастает количество циклов.

Рис. 2 иллюстрирует ожидаемое и реальное качество результата, соответственно рисунку 1. Можно видеть переоценку в точке 20×30 , но в целом средняя относительная ошибка по качеству составляет 7%. Ресурсы, выделенные на выполнение первых трех запросов, практически равны. И в первых двух случаях их достаточно для точного выполнения операции, то в третьем их не хватает, и качество резко падает. Так же падение качества в остальных случаях объясняется тем, что все объекты хотя бы из одного входа уже невозможно поместить в памяти целиком.

Остальные эксперименты не представлены из-за ограничений по размеру тезисов.

Связанные работы

Вдохновением для этой работы послужила работа [3]. Автор представляет эффективный адаптивный алгоритм выполнения соединения, который комбинирует преимущества традиционных алгоритмов соединения. Основным отличием этого алгоритма является приспособленность к работе с потоками и способность управлять качеством, изменяя количество выделенных ресурсов.

В [7] представлен адаптивный алгоритм, основанный на `hash join`, и подходящий для многопользовательской среды. Авторы предлагают гибкую технологию, созданную для работы со средой с динамически изменяющимся количеством доступной памяти. Главная цель работы заключается в регулировании использования ресурсов, чтобы позволить выполнять запрос одновременно с работой других приложений.

Сравнение параллельного алгоритма соединения вложенными циклами и соединения на основе хеширования представлено в [1]. Хотя авторы рассматривают соединение вложенными циклами, один из предложенных вариантов предполагает сортировку внешнего входа.

Выполнение запросов с ограничениями времени предложено в [4]. Авторы предлагают расширенные SQL запросы с различными параметрами.

Таким образом, есть широкий набор разнообразных алгоритмов для выполнения соединения. Однако в основном они созданы для соединения по подобию или представляют собой адаптивные алгоритмы для точного соединения.

Заключение

Мы представили детальное описание потокового адаптивного алгоритма для приближенного выполнения соединения на основе алгоритма соеди-

нения слиянием сортированных списков, в предположении, что он будет использоваться в контексте [5].

Так же в работе приведена модель стоимости предложенного алгоритма, вычисляющая предположительное качество результата, полученного за выделенное количество времени, и позволяющая управлять соотношением ресурсы/качество.

Литература

1. *DeWitt DJ, Naughton JF, Burger J* (1993) Nested loops revisited. In: PDIS, pp 230–242
 2. *Oxana Dolmatova, Anna Yarygina, Boris Novikov*: Cost Models for Approximate Query Evaluation Algorithms. DB&Local Proceedings 2012:20–28
 3. *Goetz Graefe*: New algorithms for join and grouping operations. Computer Science — R&D 27(1): 3–27 (2012)
 4. *Ying Hu, Seema Sundara, Jagannathan Srinivasan*. Supporting time-constrained sql queries in oracle. In Proceedings of the 33rd international conference on Very large data bases, VLDB'07, pages 1207–1218. VLDB Endowment, 2007
 5. *Boris Novikov, Natalia Vassilieva, Anna Yarygina*: Querying big data. CompSysTech 2012: 1–10
 6. *Anna Yarygina, Boris Novikov*: Optimizing the Resource Allocation for Approximate Query Processing. ADBIS (2) 2012: 297–308
 7. *Hansjörg Zeller, Jim Gray*: An Adaptive Hash Join Algorithm for Multiuser Environments. VLDB 1990: 186–197
-

MODELING OF HYBRID CDN-P2P FOR FULL HD VIDEO DOWNLOADING WITH REALISTIC DEMAND DISTRIBUTION

Салищев С.И.

ст. преп. кафедры информатики СПбГУ, sergey.i.salishev@gmail.com

Шеин Р.Е.

*студент кафедры системного программирования СПбГУ,
marso.des@gmail.com*

Introduction

With rising popularity of Full High Definition (HD) quality video the problem of delivering such content to end-users online appeared. The existing Internet infrastructure is physically unable to stream Full HD quality video on demand to a significant number of users even in well-developed areas not mentioning rural areas and developing countries. So the only feasible solution is content downloading.

Even for content downloading, the network bandwidth is still a problem. To overcome it a CDN is usually employed which distributes payload across different nodes and network segments. The number of films currently is over 100K and growing exponentially over time. Due to the long-tail property of the demand distribution [1, 2] the number of the most popular items is growing at the same rate. Large number of popular items reduces the efficiency of caching and multicasting of data as most of the neighboring users watch different content. So the CDN capacity should exponentially grow with the demand.

Another major problem is the content protection from redistribution. Unsecure digitally downloaded content is ready for redistribution with effort as small as one click, as opposed to the long process of grabbing a Blu-ray Disk (BD). So the content should be copy-protected. Digital Rights Management (DRM) is a standard technique in the downloaded video copy-protection. On the other hand, DRM hinders the user experience as it severely limits the ways of using the content below the level considered as the fair use. Also, DRM is always decrypted before showing, so it is susceptible to attack on the chain-of-trust, i.e. hacking the playback pipeline after the decoding. All known DRM including HDCP have already been hacked by hardware or software means.

Due to bandwidth and copy-protection problems there is no Full HD-quality online video service which could compete in popularity with physical BD sales. On the contrary, DVD physical sales are pushed by online services like Netflix, Hulu, and Amazon. Based on the multibillion size of the video market, developing such a service may create a promising business opportunity.

P2P is similar to CDN except two properties. First, peers are not reliable sources, they can appear and disappear at random leading to Quality of Service

(QoS) degradation which is a problem for video streaming scenarios. Second, peers are not trusted to own non-copy-protected content. The content copy-protection problem for P2P is yet to be solved.

P2P effectively solves the bandwidth problem as it effectively scales up with the number of users with the cost burdened on the users themselves. For the downloading scenario the constant QoS is a secondary concern. As in case of low QoS, a user can watch the video offline after downloading. The efficiency of P2P is supported by the latest Internet traffic analysis attributing about 30% of total Internet traffic to P2P [3]. P2P is future-proof for new content types like HDR video, retina resolution video, multi-view 3D video.

While DRM can be adapted to P2P networks it is not quite suitable for it, as P2P is based not only on Internet networking, but also on social networking between users which presumes hardware neutrality and user convenience. The solution is embedding digital fingerprint using video watermarking on each copy of video and using traitor tracing to discourage users from redistribution.

In this paper we consider Hybrid CDN-P2P system for video downloading. It is different usage model from streaming used by existing Adobe Live Video P2P system and system considered by LaFortune et al. [4]. As user can watch the content off-line QoS requirements are easier to comply as the system only needs to guarantee completing the download in reasonable time independently from order of blocks. While video streaming needs to guarantee a constant throughput for sequential blocks. In opposite to streaming, downloading system can provide reasonable service even for low bandwidth users and allows better load balancing as video is persistently stored on clients.

The contribution of this paper is as follows:

We analyze the data from Demonoid torrent tracker [5] and show that it is better fitted by Stretched Exponential (SE) distribution (complementary to Weibull) than Zipf distribution which supports the data of Guo et al. [6] on demand distribution in commercial VoD systems and for user generated video content [7]. It suggests that SE distribution is universal for video demand independently from content source and method of delivery.

We compare the data from Demonoid vs. IMDB [8], and demonstrate good correlation between user demand distribution and votes distribution within one year. It allows to predict the demand distribution for real video delivery systems based on publicly available popularity statistics independently of method of video delivery.

We implement the agent based model of P2P assisted CDN for video downloading using bit-torrent protocol with Demonoid demand distribution. Simulation shows 94% reduction in CDN traffic which exceeds the 75% reduction reported by LaFortune et al. [4] for Hybrid CDN-P2P video streaming model. The main difference of our model is that we drop all QoS requirements needed for streaming and only implement QoS requirements embedded into bit-torrent protocol which guarantee timely download completion. Reduction is similar to the numbers achieved experimentally for large scale P2P software distribution [9].

Data sets and demand analysis

For this analysis we gathered Demonoid user statistics snapshot on 13.10.2011 and IMDB votes snapshot on 26.10.2011.

We considered the CDN assisted with P2P network used for content downloading. Our hypothesis is that such a network would behave similarly to existing bit-torrent networks. We aimed to assess the impact of P2P assistance and downloading strategy on the CDN data traffic.

Our first goal was to accurately study the user behavior. We used Demonoid statistics to analyze the actual demand in P2P networks (Fig. 1). The Demonoid data was fitted with Zipf distribution common to Web content and Stretched Exponential Distribution. SE better captures the demand distribution $R^2=0.99$ than Zipf $R^2=0.77$ with the scale parameter close to reported by Guo et al. [6] for commercial VoD systems.

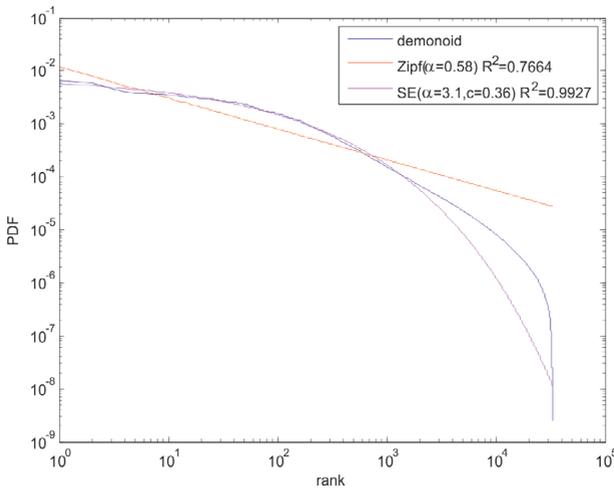


Figure 1. «Demonoid» downloaders in category «movie»

Tracker statistics snapshot in the category «Movies» by the end of 2011 containing 33K of unique film names with non-zero number of downloaders shows that 13.7% of films serve 86.3% of the users which is in agreement with statistics of Tan et al [10]. It should be noted that for niche markets the rate can be substantially higher, i.e. 28/72 for «noir» films (Fig.2).

As we are interested in easier and more reliable prediction of user demand we investigated the correlation of the number of torrent downloaders with a popularity rating on ratings sites. We observed that within one year there is a high correlation (Pearson's $r=0.99$) between the ranked number of votes on IMDB and

ranked number of torrent downloaders on overlapping subset of movies (Fig. 3). It suggests that both values are generated by similar processes.

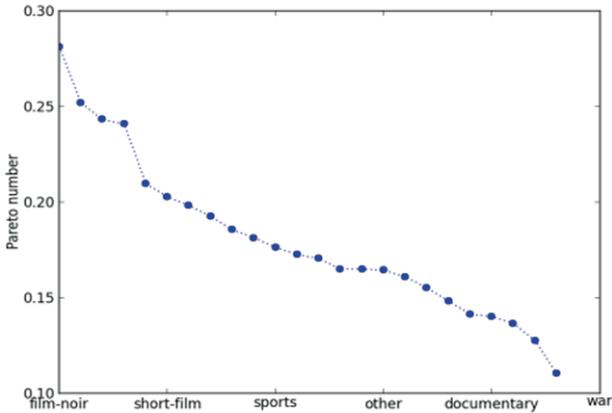


Figure 2. Pareto number per category

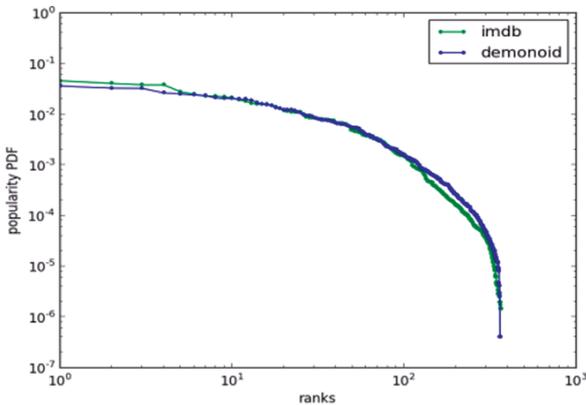


Figure 3. «IMDB» vs. «Demonoid» demand on overlapping subset (ranked), 2010 year

The correlation between votes and demand probability for the same film is much less explicit (Fig. 4); with $r=0.84$. It means that vote distribution only predicts the shape of demand distribution but not the demand probability for the specific film. This may be explained by the snapshot nature of Demonoid data and differences in audience between Demonoid and IMDB. The overall behavior for all years is substantially different as torrent popularity decreases exponentially over time, which does not happen on ratings sites.

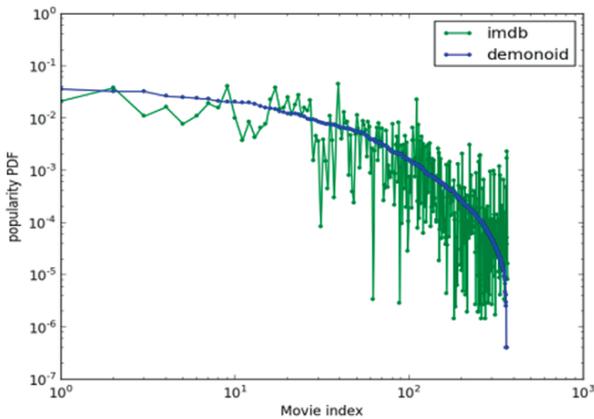


Figure 4. «IMDB» vs. «Demonoid» demand on overlapping subset, 2010 year

The usual model for the video demand probability is a Zipf distribution which is commonly observed in citation ratings and word frequencies in natural languages. However, it's noted that the actual video demand distribution has differences [6, 7], that are a limited fetch for top-rated items and an exponential cutoff for low-rated items so it is better fitted by Stretched Exponential distribution which is supported by our data. Currently there is no clear explanation of the process generating this distribution.

P2P simulations

To investigate the impact of changing the system architecture we implemented a behavioral simulator of a P2P network based on ROSS framework [11]. We used two-layer network model consisting of a star-like backbone model and local networks (LAN) with CDN modeled as a super-peer (Fig. 5). We implemented bit-torrent protocol simulation for communication between nodes. As we considered only downloading usage model we implemented no QoS requirements needed for streaming. We only implemented QoS requirements of bit-torrent protocol which guarantee the download completion in reasonable time. During the simulation with 10K peers we observed CDN traffic reduction by 94%. This is higher than 75% reduction reported by LaFortune et al. [4] for modeling of Hybrid CDN-P2P video streaming. Reduction is similar to the numbers achieved experimentally for large scale P2P software distribution [8]. We achieve better result due to weaker QoS requirements and larger swarm of uploaders for the downloading scenario. The result suggests that P2P assisted CDN can be profitable even if operating in niche markets, i. e. no top-rated movies.

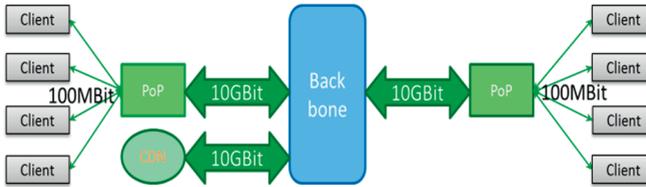


Figure 5. Star network topology

Larger-scale simulations based on GPS P2P simulator [12] were performed to eliminate potential mistakes in bit-torrent stack implementation. For these simulations a similar star topology network model with 200 LANs with 512 peers each was considered, CDN and tracker being super-peers. Due to performance issues in the simulator's core only a brief interval of time was feasible for simulation: after approximately 45–48 seconds simulated the simulator's core practically stalls due to overgrowth of events queue. The data obtained show that upon stabilizing the CDN traffic reduction is 94% or more.

Integrated copy-protection

The bit-torrent protocol does not provide the copy-protection. Copy protection is a critical requirement from the content owners for any practical implementation of a video distribution system. To amend this problem we consider the combined solution including a centralized billing system (Fig. 6).

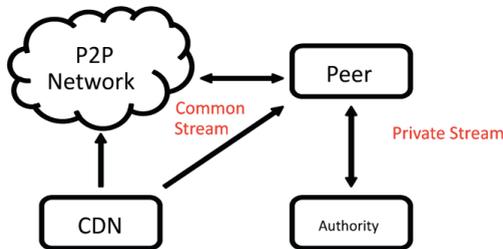


Figure 6. System with copy-protection layer

The content is divided into a public and a private layer. The public layer is distributed through P2P-assisted CDN, while the private layer is distributed through a centralized system and is unique per user. The public layer is «encrypted» making the content useless to the user without the private layer. This can be achieved by modifying the semantic elements of video stream preventing it from decoding or producing embarrassing artifacts [13]. After downloading both layers are jointly

«decrypted» to provide a useful copy. The private layer can be as small as 1 % of the data which does not substantially hinder the system performance. This approach is compatible with both DRM and fingerprinting copy-protection schemes.

Conclusions

The correlation between Demonoid and IMDB ratings allows us to predict user demand based on publicly available statistics. We use Demonoid demand probability distribution to simulate Hybrid CDN-P2P video distribution system for downloading. Simulations with different tools show 94% traffic reduction for CDN. This demonstrates that Hybrid CDN-P2P for video downloading can effectively solve bandwidth shortage for full HD quality video distribution. Paired with the copy-protection suggested above and some motivation for peers to keep and upload stored video to the system the approach can dramatically improve speed and reduce maintenance costs of video distribution services. Most likely, the model can be adapted to other categories of bulk downloads beyond video like video games.

References

1. *C. Anderson*, «The Long Tail: Why the Future of Business is Selling Less of More,» Hyperion, 2006.
2. *S. Goel, A. Broder, E. Gabrilovich, B. Pang*, «Anatomy of the long tail: ordinary people with extraordinary tastes,» In Proceedings of the third ACM international conference on Web search and data mining (WSDM '10). ACM, New York, NY, USA, 2010, pp.201–210.
3. *C. Labovitz*, «The Other 50% of the Internet», 54th North America Network Operators' Group Meeting (NANOG54), February 2012.
4. *R. LaFortune, C.D. Carothers, W.D. Smith, J. Czechowski, W. Xi*, «Simulating Large-Scale P2P Assisted Video Streaming,» 42nd Hawaii International Conference on System Sciences (HICSS '09), Jan. 2009.
5. «Demonoid torrent tracker,» <http://demonoid.me>
6. *L. Guo, E. Tan, S. Chen, Z. Xiao, X. Zhang*. «Does internet media traffic really follow Zipf-like distribution?» In Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '07). ACM, New York, NY, USA, 2007, pp.359–360.
7. *M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, S. Moon*. «Analyzing the video popularity characteristics of large-scale user generated content systems,» IEEE/ACM Trans. Netw. 17, 5 October 2009, pp.1357–1370.
8. «International Movie Database,» <http://imdb.com>
9. *C. Huang, A. Wang, J. Li, K.W. Ross*, «Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh,» In Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '08). ACM, New York, NY, USA, 2008, pp.75–80
10. *T.F. Tan and S. Netessine*. Is Tom Cruise threatened? Using Netflix Prize data to examine the long tail of electronic commerce. Working Paper, 2009.

11. *C.D. Carothers, D. Bauer, S. Pearce*, «ROSS: A high-performance, low-memory, modular Time Warp system,» *Journal of parallel and distributed computing* (Elsevier) 62 (11), 2002, pp.1648–1669.
 12. GPS peer-to-peer simulator <http://www.cs.binghamton.edu/~nael/gps/>
 13. *Z. Dekun, N. Prigent, J. Bloom*, «Compressed video stream watermarking for peer-to-peer based content distribution network,» *IEEE International Conference on Multimedia and Expo, (ICME 2009)*, pp.1390–1393, 2009, June 28
-

ИЗВЛЕЧЕНИЕ СОБЫТИЙ НА ОСНОВЕ АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ ЛИНЕЙНЫХ ШАБЛОНОВ

Дзендзик Д.А.

*студент кафедры системного программирования СПбГУ,
daryadzen@gmail.com*

Введение

С каждым годом количество информации, в том числе текстовой на естественном языке, во всём мире увеличивается с огромной скоростью¹. Данный факт существенно повышает интерес к автоматической обработке информации и анализу текстов.

Среди естественных языков по числу говорящих в мире английский язык находится на 3–4 месте². Однако, больше половины (55,1%) всех текстов в интернете написаны именно на английском языке³. Эти факты указывают, на распространённость английского языка во всём мире, поэтому описываемый алгоритм построения шаблонов работает именно с английским языком.

Задача извлечения событий формулируется следующим образом: автоматически выявить событие и, в идеале, определить «кто сделал что, с кем, когда, каким образом, где и зачем». Мы определяем события как комбинацию индикатора события и набора атрибутов, которые имеют определенное значение. Например, в предложении «Autonomy was acquired by Hewlett-Packard in October 2011» содержит в себе событие типа «Слияние и поглощение» (*Merger&Acquisition, M&A*) с двумя аргументами и одним индикатором. Аргументы: покупатель (Hewlett-Packard), покупаемый (Autonomy); индикатором рассматриваемого события является «was acquired» — слово или последовательность слов, которое однозначно указывает на возможное событие в рассматриваемом предложении.

Линейное правило — правило в виде регулярного выражения над графом аннотаций, элементы которого связаны друг с другом только порядком и не имеют больше никаких зависимостей. Правило, как таковое, состоит из двух частей: первая содержит информацию о контексте, и именно её строит рассматриваемый алгоритм; вторая часть является действием над сопоставляемой с первой частью текстом и однозначно определяется им. В нашем

¹ <http://www.wn.ru/internet/01.07.2011/3.html>

² http://www.ethnologue.com/ethno_docs/distribution.asp?by=size
<http://facte.ru/samyj-rasprostranennyj-yazyk-v-mire.html#.UKu9D4daw3o>
http://e-kniga.ru/advert/anglijskij_yazik.htm

³ http://w3techs.com/technologies/overview/content_language/all

случае действие всегда является «*пометить как событие*». В процессе разметки потенциальных событий система использует дерево разбора синтаксических зависимостей.

Термины «шаблон» и «правило» в данной работе используются синонимично.

Были проведены эксперименты, сравнивающие результат работы правил, построенных алгоритмом, и правил, написанных вручную человеком. Результат построенных правил получил более высокую оценку по трём метрикам: точность, полнота и F1-мера.

Процесс обработки текста и построение правил

Предлагаемый в данной работе алгоритм построения правил состоит из четырёх частей. В первой части на основе описания событий и синтаксического анализа выделяются потенциальные события. Следующим этапом все события группируются по типу синтаксической связи между индикатором и атрибутами. Третьим этапом следует проверка пользователем примеров из каждой группы. В результате этого шага получают размеченный корпус и основа для построения правил. На последнем этапе строятся и обобщаются линейные правила в виде регулярных выражений.

До начала работы алгоритма исходный текст подвергается предобработке. Каждый документ из корпуса делится на предложения. Каждое предложение делится на токены (слова). После этого в предложении извлекаются именованные сущности (Named Entity Recognition), такие как *Company*. Следующим шагом извлекаются индикаторы возможных событий с помощью словарей (обычно глагол или глагольная группа). Полученной информации достаточно, чтобы перейти к поиску потенциальных событий.

Построение потенциальных событий

Каждое событие характеризуется индикатором и списком обязательных и дополнительных аргументов и атрибутов. Например, рассматриваемое в данной работе событие «Слияние и поглощение» описывается индикатором типа *AcquisitionIndicator* и двумя обязательными аргументами типа *Company*.

Если в предложении есть индикатор и присутствуют атрибуты в достаточном количестве, то строятся события: для каждого индикатора рассматриваются все сущности, подходящие на роль аргументов. В итоге получается набор всех возможных комбинаций из индикатора и атрибутов.

После формирования потенциального события вычисляется путь от каждого аргумента до индикатора в дереве синтаксических зависимостей, полу-

ченном в результате синтаксического разбора при помощи *StenfordParser*⁴. В результате для каждого события существует не линейное правило содержащее индикатор, атрибуты и синтаксические связи. Например: «*AcquisitionIndicator Company:agent Company: nsubjpass*».

Группирование событий

Следующим шагом обработки корпуса является группирование предложений по типам синтаксических связей. Сформированные на предыдущем шаге синтаксические связи являются критерием группирования.

В данном разделе допускается следующее предположение: синтаксическая связь индикатора и аргументов однозначно определяет истинность рассматриваемого события. Описанная гипотеза находит своё обоснование в работе[1].

Далее из каждой группы выбирается несколько предложений для предоставления их пользователю. Они нужны для принятия более точного решения, поскольку одно единственное предложение, характеризующее целую группу, может иметь сложный синтаксис или тяжело восприниматься пользователем. Также в этом случае можно избежать неверного решения из-за ошибки разметки.

Проверка пользователем

Пользователю предоставляется несколько предложений, которые были случайно выбраны из каждой группы. С помощью специального редактора *CAS Editor* для *Eclipse IDE* предложения предстают для пользователя в графическом виде, что значительно упрощает проверку. Данный инструмент позволяет подсветить разными цветами в предложении само потенциальное событие, а также его индикатор и атрибуты.

Первыми на проверку отправляются наиболее многочисленные группы. Предполагается, что именно они содержат наиболее часто встречаемые события. Группы, содержащие небольшое количество элементов, могут даже не рассматриваться. Например, можно не рассматривать синтаксические связи и группы, содержащие всего один элемент. Даже если в этом случае будет упущено событие, оно мало повлияет на оценку работы всей системы.

После разметки пользователем каждой группе присваивается метка истинности или ложности на основе преобладания действительных или ложных событий в выборке из потенциальных событий. На основе групп, содержащих истинные события, строится размеченный корпус событий, который впоследствии используется как золотой стандарт. Часть этого корпуса становится основой для линейных правил, другая часть используется для оценки полученных правил.

⁴ <http://www-nlp.stanford.edu/software/lex-parser.shtml>

Построение линейных правил

Завершающей стадией алгоритма является построение линейных правил. Построение правил начинается с рассмотрения примера события из золотого стандарта. Начальная версия правила представляет собой часть предложения в самом частном виде. Далее с помощью одной из четырёх операций оно приводится к более общему виду. К модифицированному правилу снова применяется одна из операций: обобщение узла в системе типов TextMARKER[5]; изменение квантификатора, особая обработка запятых в предложении, особая обработка артикля. Процесс продолжается до тех пор, пока улучшается F1-мера применения обрабатываемого правила к тренировочному множеству. На каждом шаге для следующей модификации выбирается правило, получившее более высокое значение F1-меры, чем остальные.

В таблице 1 представлен пример правила.

<i>Company ANY*? AcquisitionIndicator ANY*? Company{-> GATHER (AcquisitionEvent, 1, 2, 3, 4, 5)};</i>	
Левая часть:	<i>Company ANY*? AcquisitionIndicator ANY*? Company</i>
Действие:	<i>GATHER(AcquisitionEvent, 1, 2, 3, 4, 5,)</i>
Контекст:	ANY*? Любой символ, который может встречаться в произвольное, в том числе нулевое, количество раз

Таблица 1. Подробное описание линейного правила для системы TextMARKER

В процессе обобщения алгоритм запоминает все промежуточные варианты правила и их оценку, и если в последующем мы получаем один из уже рассмотренных ранее вариантов, обобщение для данного правила прекращается, поскольку результат уже известен.

Так как оценка производится на одном и том же наборе данных, то нет необходимости начинать построение правила с тех событий (предложений), которые уже были извлечены ранее построенными правилами.

Эксперименты

В качестве итога данной работы были проведены эксперименты над некоторыми новостными статьями. В качестве альтернативного метода извлечения событий рассматривались правила, построенные вручную человеком. Данные правила были получены в результате анализа новостного корпуса и, в том числе, содержат самые общие правила для извлечения событий.

Всего было рассмотрено 1 379 предложений, из которых 264 содержали событие, согласно вышеописанной разметке. Весь корпус был разделён на тестовое и тренировочное множество в соотношении 1:2 (тестовое множе-

ство 455, тренировочное — 924 предложения). В таблице 2 представлены результаты эксперимента.

	РП		АП	
	Трен.	Тест	Трен.	Тест
Кол-во правил	10	—	5	—
Точность	0.667	0.667	0.802	0.785
Полнота	0.269	0.293	0.802	0.756
F1-мера	0.386	0.407	0.802	0.77

Таблица 2. Результат работы «ручных» правил (РП) и построенных алгоритмом (АП) на тренировочном (Трен.) и тестовом множествах (Тест).

Правила, построенные в результате работы описанного алгоритма, получили более высокую оценку, чем правила, построенные человеком (по метрикам: точность, полнота и F1-мера).

Связанные алгоритмы

Существует несколько алгоритмов построения правил. Наиболее известным из них, работающий с несколькими атрибутами, это Wisk[8]. Он использует аннотированный корпус документов и синтаксический разбор предложения. В отличие от предложенного алгоритма, который работает с уже размеченными аннотациями, Wisk требует непосредственного участия пользователя в процессе обучения.

Snowball[2][3][7] и ExDisco[3][9] — примеры алгоритмов для обучения правил, извлекающих информацию из неразмеченного текста. Оба эти алгоритма используют небольшое множество положительных примеров и итеративно обрабатывают текст. Snowball извлекает бинарные отношения из неразмеченного текста. К сожалению, не все события могут быть описаны таким образом. ExDisco разделяет весь корпус на множество релевантных и не релевантных документов и использует следующую концепцию: множество релевантных документов содержит релевантные события и релевантные события содержатся в релевантных документах.

Также существуют алгоритмы Open Event Extraction, извлекающие информацию из множества неаннотированных Веб-документов. Наиболее известные из них — KnowItAll [4][3] and TextRunner[4][3].

Использование регулярных выражений для обучения на аннотированном корпусе обсуждается в работе [5]. Авторы демонстрируют возможность применения их метода для поиска имён, url-ссылок и других сущностей.

Используемый в данной работе алгоритм обобщения правил похож на алгоритм, описываемый в работе [6]. Рассматриваемый алгоритм имеет другой критерий остановки обобщения правила.

Заклучение

В данной работе был рассмотрен алгоритм построения линейных правил при помощи полуавтоматической разметки. Полученные в результате работы алгоритма правила показали свои преимущества перед правилами, которые были построены вручную человеком.

Литература

1. *Hao Li A and Xiang Li A and Heng Ji A and Yuval Marton B.* Domain-independent novel event discovery and semi-automatic event annotation.
 2. *E. Agichtein and L. Gravano.* Snowball: extracting relations from large plain-text collections. In Proceedings of the 5th ACM conference on Digital libraries, DL '00, pages 85{94, New York, NY, USA, 2000. ACM.
 3. *N. Bach and S. Badaskar.* A Review of Relation Extraction. 2007.
 4. *O. Etzioni, M. Banko, S. Soderland, and D.S. Weld.* Open information extraction from the web. *Commun. ACM*, 51(12):68{74, Dec. 2008.
 5. *P. Kluegl, M. Atzmueller, and F. Puppe.* Integrating the rule-based ie component textmarker into uima. In J. Baumeister and M. Atzmueller, editors, LWA-2008 (Special Track on Information Retrieval), pages 73{77, 2008.
 6. *Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H.V. Jagadish.* Regular expression learning for information extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 21{30, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
 7. *R. McDonald.* Extracting relations from unstructured text. Technical report, 2005.
 8. *S. Soderland.* Learning information extraction rules for semi-structured and free text. *Mach. Learn.*, 34(1–3):233{272, Feb. 1999.
 9. *R. Yangarber, R. Grishman, and P. Tapanainen.* Automatic acquisition of domain knowledge for information extraction. In Proceedings of the 18th International Conference on Computational Linguistics, pages 940{946, 2000.
-

Кибернетика и робототехника



**Фрадков
Александр Львович**

Д.Т.Н.

профессор кафедры теоретической кибернетики СПбГУ
заведующий лабораторией
«Управление сложными системами» ИПМаш РАН



**Лучин
Роман Михайлович**

ст.преподаватель кафедры теоретической кибернетики СПбГУ
научный сотрудник лаборатории теоретической кибернетики

ПОСТРОЕНИЕ КАРТ СМЕЩЕНИЙ ПО НЕРЕКТИФИЦИРОВАННЫМ СТЕРЕОИЗОБРАЖЕНИЯМ

Корчемкин Д.А.

студент СПбГУ, dmitriy.korchemkin@gmail.com

Салищев С.И.

*старший преподаватель кафедры информатики СПбГУ,
sergey.i.salishev@gmail.com*

Аннотация: В данной работе описаны модификации алгоритмов слепой ректификации стереоизображений и построения плотных карт смещений для ректифицированных изображений.

Введение

Построение карт смещений является предварительным этапом решения многих практических задач в области компьютерного зрения и компьютерной графики. Специфика различных задач накладывает дополнительные ограничения. Появление автостереоскопических многокурсных дисплеев, стереоскопических систем отображения с отслеживанием взгляда и успехи в развитии голографических дисплеев приводят к необходимости разработки алгоритмов подготовки правдоподобных трехмерных моделей для отображения на этих устройствах на основе стереоскопических изображений, снятых бытовой фото и видео аппаратурой. Это приводит к отличиям в постановке задачи от случая компьютерного зрения. Во-первых, изображения могут быть сняты различными камерами, параметры которых неизвестны. Наличие функции автоматической фокусировки приводит к изменению параметров камеры во времени. Таким образом, необходима слепая калибровка камер. Это также позволяет ректифицировать изображения, проектируя эпиполярные линии на горизонтали, что упрощает алгоритмы построения карт смещений. Во-вторых, карта смещений должна быть плотной, поскольку даже точечные выбросы приводят к визуальным артефактам отображения. В-третьих, изображения имеют высокое разрешение.

Библиотеки OpenCV[1] и Mathworks Computer Vision System Toolbox[2] содержат функциональность для решения задач слепой ректификации. При использовании этих алгоритмов на практике они показали свою неустойчивость. В работе предлагаются модификации алгоритма слепой ректификации для повышения его устойчивости.

Задача построения плотной карты смещений решается Колмогоровым[3] с помощью алгоритма глобальной оптимизации на основе нахождения минимального разреза графа. На практике алгоритм неприменим для изображений высокого разрешения из-за вычислительных затрат. Гейгер[4] пред-

лагает менее вычислительно сложный двухшаговый алгоритм, во многих случаях дающий не худшие результаты, чем алгоритм Колмогорова. Алгоритм Гейгера чувствителен к недостаточной плотности опорных точек и к ошибке в смещении опорных точек. В работе рассмотрена модификация алгоритма, устраняющая эти недостатки.

Ректификация изображений

Для слепой ректификации использовался алгоритм предложенный Фузелло[5]. Этот алгоритм аналогичен алгоритмам из библиотек OpenCV и Mathworks Computer Vision System Toolbox. Алгоритм состоит из 4 этапов: выделение особых точек, сопоставление особых точек на паре изображений, вычисление фундаментальной матрицы, факторизация фундаментальной матрицы. Выделение особых точек осуществлялось с помощью алгоритма SURF[6]. Вычисление фундаментальной матрицы выполнялось с помощью алгоритма RANSAC[7]. При экспериментальной проверке было установлено, что наличие среди найденных сопоставлений неверных сильно влияет на результат ректификации. Для увеличения устойчивости ректификации к неверным сопоставлениям использовалась фильтрация особых точек исходя из предположения о том, что верные сопоставления преобладают среди найденных и образуют на изображениях группы, имеющие практически одинаковые размер и форму на обоих изображениях

Кроме того, производится несколько решений задачи оптимизации с различными начальными данными (фокусным расстоянием и углами поворота) с целью выбора решения, лежащего в заданных границах.

Введём обозначения: $m_i^l = (x_i^l, y_i^l)$, $m_i^r = (x_i^r, y_i^r)$ ($i=1, \dots, N$) для найденных с помощью SURF сопоставлений на левом и правом изображениях соответственно; обозначим однородные координаты сопоставлений $\bar{m}_i^l = (x_i^l, y_i^l, 1)$, $\bar{m}_i^r = (x_i^r, y_i^r, 1)$.

Фильтрация с помощью разбиения на кластеры

Для отсеивания неверных сопоставлений особые точки на одном из изображений разбиваются на кластеры. Разбиение точек на кластеры производилось с помощью алгоритма k-means для стандартного расстояния на плоскости; оценка на число кластеров — \sqrt{N} .

Обозначим C_j — индексы сопоставлений, принадлежащих j -ому кластеру, N_j — количество сопоставлений, принадлежащих кластеру C_j ; c_j^l, c_j^r ($j=1, \dots, N_c$) — центры кластеров на левом и правом изображениях; расстояние между точками из сопоставления обозначим $d_i = \|m_i^l - m_i^r\|$; и медиану расстояний для точек кластера $M_j = \text{median}\{d_i \mid i \in C_j\}$; Расстояние до центра кластера на левом изображении $d_i^l = \|m_i^l - c_j^l\|$ (j — индекс кластера, которому соответствует сопоставление); расстояние до центра

кластера на правом изображении $d_i^r = \|m_i^r - c_j^r\|$; $M_j^l = \text{median}\{d_i^l \mid i \in C_j\}$, $M_j^r = \text{median}\{d_i^r \mid i \in C_j\}$.

Исходя из предположения, что особые точки образуют сгущения на изображении, сопоставление считается неверным, если:

- Сопоставление входит в кластер, размеры которого (M_j^l и M_j^r) отличаются более чем в k_1 раз
- Расстояние d_i сопоставления, соответствующего кластеру j , превосходит M_j более чем в k_2 раз

Фильтрация по ошибке, вносимой точкой

Считая, что среди найденных сопоставлений преобладают верные, предлагается производить ректификацию в несколько этапов, отсеивая точки, вносящие наибольшую ошибку в промежутке между итерациями.



Рис. 1. Пример неверного сопоставления, которое будет исключено при фильтрации путём разбиения на кластеры

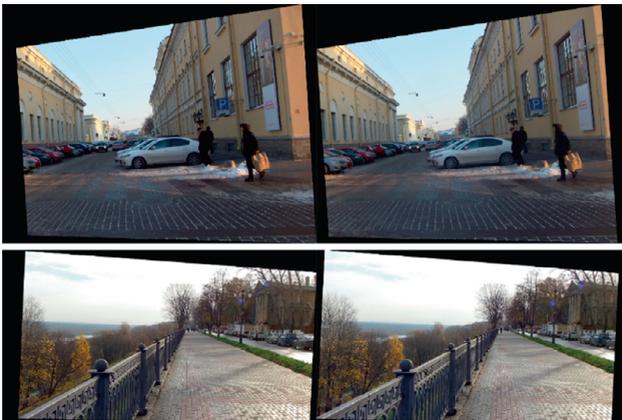


Рис. 2. Пример результата ректификации

Поиск смещений

Для поиска смещений рассматривался алгоритм, предложенный Гейгером. Алгоритм состоит из двух основных шагов: вычисление смещений в опорных точках, и решение оптимизационной задачи для внутренних точек треугольников триангулированной сетки опорных точек. Алгоритм существенно выигрывает в производительности у глобальных алгоритмов, но является чувствительным к выбору опорных точек. Рассматривались пути улучшения работы этапа выбора опорных точек на больших (>6Мрх) изображениях; изображения были предварительно ректифицированы с помощью алгоритма, описанного выше.

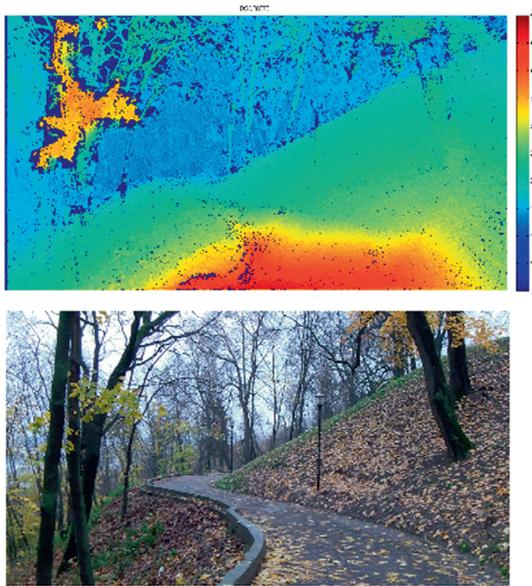


Рис. 3. Пример построенной карты смещений

Выбор дескриптора для точек изображения

Обозначив I — интенсивности точек изображения, результаты свёртки

оператором Собеля по горизонтали $S_x = I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; результаты свёртки

оператором Собеля по вертикали $S_y = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$.

В качестве дескрипторов точек изображения были рассмотрены значения I , S_x , S_y , а также — комбинированный дескриптор из S_x , S_y .

В качестве метрики схожести участков изображений использовалась сумма модулей разности (SAD). Для оценки того, насколько качественно дескриптор подходит для поиска смещений, использовался набор изображений middlebury [8], содержащий ground truth.

Было установлено, что наилучшие результаты (в смысле доли верных смещений, соответствующих минимуму) получаются при использовании комбинированного дескриптора.

Использование информации о цвете

В ходе работы была проверена гипотеза о том, что использование информации о цвете может привести к повышению плотности карты смещений и повышению точности найденных смещений. На наборе изображений middlebury (схожесть участков изображения определялась как сумма схожести по каждому из каналов, которая, в свою очередь, основывалась на описанном выше комбинированном дескрипторе) показала, что с использованием информации о цвете уменьшается количество точек, для которых совпадает смещение, задаваемое минимумом SAD дескрипторов точки и её (возможного) образа на другом изображении при поиске смещения слева направо и справа налево, но доля точек, для которых смещение найдено верно, увеличивается; это может быть полезно для нахождения опорных точек в рассматриваемом алгоритме.

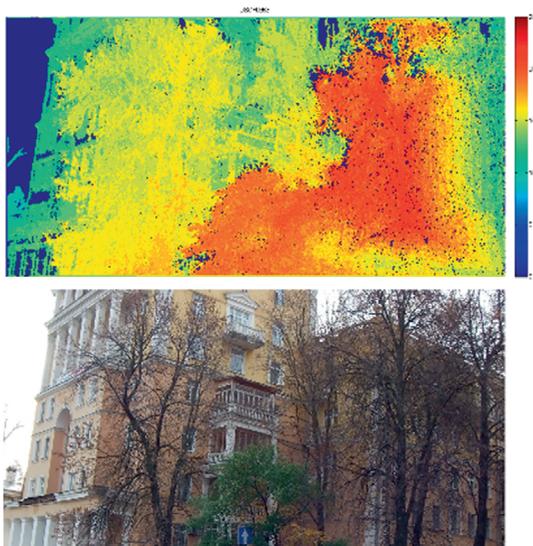


Рис. 4. Пример построенной карты смещений

Заключение

В данной работе рассматриваются улучшения алгоритмов слепой ректификации и построения плотных карт смещений по изображениям, полученным бытовой фото и видео аппаратурой. В качестве тестового стенда использовался автофокусный стереофотоаппарат Fujifilm Finerix Real3D W3. В целом был получен удовлетворительный результат как на изображениях, сделанных авторами, так и на стандартном тестовом наборе.

В дальнейшем предполагается использовать полученные карты смещений для восстановления промежуточных ракурсов и увеличения реалистичности отображения трехмерной сцены на стерео дисплее с помощью отслеживания положения головы пользователя.

Литература

1. OpenCV, <http://opencv.org>
 2. Mathworks Computer Vision System Toolkit www.mathworks.com
 3. *Vladimir Kolmogorov and Ramin Zabih*. Multi-camera Scene Reconstruction via Graph Cuts. In Proceedings of the 7th European Conference on Computer Vision-Part III (ECCV '02), 2002. Springer-Verlag, London, UK, 82–96.
 4. *Andreas Geiger, Martin Roser, Raquel Urtasun*. Efficient Large-Scale Stereo Matching // Asian Conference on Computer Vision (ACCV) — 2010
 5. *A. Fusiello, L. Irsara*. Quasi-euclidean uncalibrated epipolar rectification // Pattern Recognition — 2008 C 1–4
 6. *Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool* «SURF: Speeded Up Robust Features», Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008
 7. *David A. Forsyth and Jean Ponce* (2003). Computer Vision, a modern approach. Prentice Hall. ISBN 0-13-085198-1.
 8. D. Scharstein Middlebury Stereo Datasets <http://vision.middlebury.edu/stereo/data/>
-

Преподавание компьютерных дисциплин для математиков



**Гейн
Александр Георгиевич**

к.ф.-м.н., д.п.н., профессор
руководитель круглого стола

СРЕДЫ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ ДЛЯ ПРЕПОДАВАНИЯ

Луцив Дмитрий Вадимович

СПбГУ, ct.npen., dluciv@math.spbu.ru

Кириленко Яков Александрович

СПбГУ, ct.npen., jake@math.spbu.ru

Полозов Виктор Сергеевич

СПбГУ, ct.npen., victorp@math.spbu.ru

Аннотация: В работе освещены проблемы, возникающие при использовании языков и средств визуального моделирования в ориентированном на подготовку промышленных специалистов образовательном процессе. Обоснована целесообразность создания своих визуальных языков. Предложен краткий обзор доступных средств с акцентом на средства метамоделирования, выступающие в роли решения обозначенных проблем.

Введение

Область применения утверждения о том, что язык определяет сознание, уместно расширить с естественных человеческих языков и на такие синтетические языки, как языки программирования и языки визуального моделирования. В отношении языков программирования всё относительно ясно: решаемая задача влияет на выбор языка, а выбранный язык — на выбор инструментов для работы с ним. При этом уже не один десяток лет центральным инструментом обычно является универсальный текстовый редактор, пусть даже оптимизированный и настроенный для конкретного языка.

Нельзя, однако, забывать и то, что ряд языков программирования, таких, как Smalltalk [15] и испытавший его влияние Self [22], в канонических своих реализациях подразумевают комбинированное использование текстового редактора и графической среды, которая структурирует код, хотя отдельные их реализации, например GNU Smalltalk, графической среды не требуют и даже не подразумевают. Периодически эксперименты по графическому структурированию проводят и для классических ЯВУ, примером можно считать Code Bubbles [11] для Java.

Сказанное выше позволяет утверждать, что проблема графического представления ПО на определённом уровне детализации безусловно актуальна даже для классических «текстовых» языков.

Наибольшие надежды на средства визуального моделирования общего назначения возлагались приблизительно 15–20 лет назад, в последние же 10 лет интерес к данному классу программного обеспечения снизился, усту-

пив место ажиотажу вокруг качественных сред разработки, хорошо приспособленных для работы с определёнными языками и даже с отдельными библиотеками. Это позволяет достичь наибольшей производительности в применении подобных решений, но, к сожалению, обычно в ущерб универсальности используемых подходов. Кроме того, для решения конкретных задач часто применяются проблемно — ориентированные языки. Решения на их базе часто достаточно элегантны, но ни о каком единообразии при их создании и использовании речь уже идти не может. Следует однако отметить, что многие авторитетные производители средств разработки в последнее время уделяют особое внимание унификации инструментария для создания предметно-ориентированных «текстовых» языков. Такие подходы реализованы, например, в JetBrains MPS и Visual Studio SDK.

Вместе с тем, проблема наглядности представления большого количества информации и удобной навигации по ней не только не стала менее актуальной, но напротив, усугубилась вместе с естественным ростом сложности и объёма реализуемого, поддерживаемого и исследуемого программного обеспечения.

Обозначенная проблема одинаково остро стоит как в образовании, так и в промышленном программировании. В конечном итоге, студенты технических и математических специальностей и направлений ВУЗов обучаются с ориентацией на последующее эффективное применение своих знаний и навыков в промышленности [1]. Промышленность же с энтузиазмом идёт навстречу образованию, помогая готовить выпускников ВУЗов «для себя».

Данная работа предлагает обратить внимание на подход метамоделирования, который стремится сочетать в себе:

- свойственные графическим языкам высокие плотность и наглядность отображаемой информации,
- универсальность в контексте решаемых задач
- универсальность в контексте выбора инструментальных средств, а именно возможность использования единой среды визуального моделирования, которая позволяет легко создавать проблемно-ориентированные графические языки.

В качестве инструмента реализации подхода предлагается использование лёгкого в освоении мета-CASE средства и создание на его основе предметно-ориентированных языков а так же поддерживающих соответствующие им циклы разработки технологических решений.

План работы

- в общем и на примере курсов СПбГУ будет рассмотрен спектр применяющихся языков визуального моделирования и программирования,
- будут рассмотрены некоторые конкретные инструменты, которые можно использовать при обучении,

- будет проведён краткий экскурс по собственным средствам визуального моделирования, разработанным в СПбГУ и в сотрудничестве с СПбГУ.

Упомянув СПбГУ, будем рассматривать факты на примере специальностей и направлений отделения информатики Математико-механического факультета.

Языки, используемые в образовании и исследованиях

Общеизвестные распространённые языки

В настоящее время практически во всех ВУЗах, включая и СПбГУ, традиционно уделяется достаточно много внимания обучению объектно-ориентированному проектированию и программированию. Де-факто стандартом для описания объектно-ориентированных систем является язык UML [24]. UML хорошо проработан, самодостаточен, и, в то же время расширяем. При этом он является продуктом опыта, полученного при разработке и использовании массы других языков.

В ряде курсов для моделирования данных применяются графические языки, разработанные под влиянием SADT [21], такие как, например, IDEF1 [17]. Для моделирования процессов часто применяются IDEF0 [16] и IDEF3 [18].

Во многих курсах, в т. ч. вообще никак не связанных с визуальным моделированием, используется описание функционирования компонент систем, обрабатывающих данные, при помощи DFD [23].

В специфических курсах и исследованиях, посвящённых телекоммуникациям и реактивным системам реального времени традиционно используется язык SDL [26], а чаще «местные диалекты» его подмножеств, упомянутые ниже.

В целом, как и следует ожидать, при обучении прикладным дисциплинам набор обязательно используемых языков приблизительно совпадает с набором промышленных языков для соответствующих областей.

Начинается преподавание обычно с вводных курсов, например, [4]. На старших курсах СПбГУ (в магистратуре) отдельные дисциплины посвящены CASE-средствам. При этом подводится итог тому, что студенты уже успели изучить. В частности, в качестве общего подхода рассматривается и метамоделирование. Внимание уделяется общим вопросам, таким, как рассмотрение графических грамматик, и конкретным языкам метамоделирования, например MOF [19].

Языки, разрабатываемые внутри или совместно с СПбГУ

Так же, как и текстовые языки, языки визуального моделирования разрабатываются людьми, которым требуется решать определённый спектр за-

дач и хочется пользоваться для этого наиболее удобными выразительными средствами.

Решение задач, связанных с описанием и реализацией телекоммуникационных систем, подтолкнуло специалистов ЛГУ, а позже СПбГУ, к созданию своих диалектов подмножеств языка SDL. На данный момент актуальными являются диалекты, реализованные в рамках методологий RTST [7] и REAL [9]. Помимо телекоммуникационных задач, методология REAL и реализованные в её рамках одноимённое средство и набор инструментов используются для проектирования и реализации готовых информационных систем.

В данный момент новые языки визуального моделирования продолжают разрабатываться как в рамках масштабных проектов [12, 2] с большим количеством участников, так и в рамках отдельных квалификационных работ уровня магистерских диссертаций [8].

Основной принцип, которому следовали и продолжают следовать разработчики и пользователи перечисленных систем: диаграммы играют важную коммуникативную роль, но при этом они не просто картинки, и на основе отображаемых ими моделей по возможности следует строить генерационные технологические решения [13].

Рынок ПО

Универсальное ПО для создания диаграмм

Рынок, в широком смысле, представлен множеством открытых, закрытых и коммерческих продуктов, которые в разной степени удовлетворяют идеям, изложенным в работе выше. Рассмотрим некоторые из них.

Microsoft Visio доступен ВУЗам по программе MSDN AA, хотя в целом он не является бесплатным. Для UML поддерживаются не только диаграммы, но и простой репозиторий с моделью, которая позволяет отображать одни и те же сущности и связи на разных диаграммах, и которую можно экспортировать в формате, удовлетворяющем XMI [25]. Это позволяет построить на основе Visio генерационные решения. В то же время, Visio является в существенной степени «офисной» программой. Основной вес лежит именно на графической подсистеме, и Visio — скорее специализированный графический диаграммный редактор, нежели CASE-инструмент. Возможности расширения графической нотации у Visio достаточно просты, и, вместе с тем, богаты. В итоге Visio можно использовать для прототипирования собственных графических языков, но построение полноценных технологических решений на его базе затруднено.

В смысле редактирования диаграмм наиболее серьёзным претендентом на роль Visio со стороны мира открытого ПО является пакет Dia. Графические возможности пакета тоже достаточно богаты и легко расширяемы,

однако пакет рассчитан исключительно на рисование диаграмм и вообще не подразумевает такого понятия, как модель. Тем не менее, нельзя недооценивать роль бесплатности ПО, т.к. часто пользователи предпочитают уверенность в том, что условия предоставления продукта не ухудшатся, качеству и функциональности.

В ряде случаев, например тогда, когда надо поместить простую диаграмму прямо в документ или на презентацию, можно использовать даже средства рисования, доступные в офисных пакетах Microsoft Office (доступен ВУЗам по MSDN AA) или LibreOffice/OpenOffice (свободное ПО). Однако здесь о CASE речь не идёт, т.к. подобные средства — не более, чем ориентированные на диаграммы векторные графические редакторы.

ПО визуального моделирования в образовании

Как уже было сказано выше, использование решений, ориентированных на конкретные языки, часто бывает более эффективным и элегантным. Однако об этом в основном имеет смысл говорить по отношению к большим и долгоживущим проектам. Когда речь идёт об ознакомлении студентов с тем или иным языком моделирования или программирования, гораздо продуктивнее использовать набор стандартных средств, поддерживающих выбранные языки как минимум на удовлетворительном уровне. Тем не менее, специализированные средства часто заслуживают отдельного внимания, поскольку они могут предоставлять уникальную функциональность и выполнять свои задачи очень качественно. Перечислим некоторые из них.

Средство	Языки	Доступность
Ideogramic UML	UML	коммерческий
ERwin Data Modeller	IDEF0, IDEF3, DFD	Shareware
AllFusion Process Modeler 7	IDEF0, IDEF3, DFD	Shareware
Enterprise Architect	UML	Shareware
Visual Paradigm	UML	коммерческий с обр. программой
REAL	REAL — язык на основе UML, ROOM, SDL и WPMC	Shareware, для СПбГУ бесплатен
Umbrello UML Modeller	UML	открытое ПО
Argo UML	UML	открытое ПО
IBM Rational / Telelogic SDL Suite (панель Telelogic Tau)	SDL	коммерческий

Перечисленные продукты достаточно качественно реализуют соответствующие методологии, а так же предоставляют программный интерфейс к своим моделям для реализации, например, генерационных технологических

решений на их базе. К сожалению, некоторые из упомянутых инструментов сейчас активно не разрабатываются.

Метамоделирование

Рассмотрим несколько систем мета-CASE, поддерживающих, с одной стороны, хранение информации в модели, а с другой — описание графической нотации пользовательских языков. Как уже было сказано выше, возможность создания собственных языков и разработки технологических решений для них важна как в промышленности, так и в образовании.

Инструменты проекта Eclipse

Eclipse — очень крупный программный проект с открытым исходным кодом. Проект объединяет под своим началом как базовые модули (Framework), так и множество специализированных инструментов, обеспечивая, фактически, операционную среду для программиста конкретных инструментов на базе Eclipse, и их будущего пользователя (часто прикладного программиста). Такие общность и универсальность не достаются бесплатно. Как минимум, инструменты метамоделирования на базе Eclipse достаточно тяжеловесны и в смысле инсталляции. Помимо периодических технических проблем версионирования компонент при инсталляции и обновлении модулей, данное программное обеспечение весьма требовательно и к аппаратным ресурсам.

Тем не менее, проект Eclipse не зря обладает мировым признанием и решения на его основе часто рассматриваются в качестве предмета почти безальтернативного выбора доступных и сравнительно легко осваиваемых средств метамоделирования.

Eclipse GEMS (<http://www.eclipse.org/gmt/gems/>) является полноценным средством метамоделирования [14], позволяющим описывать сущности модели, включая атрибуты, и связи между ними. Описание при этом можно выполнять на языках Prolog и OCL [20], что является, безусловно, преимуществом, т.к. позволяет представить декларативную по сути информацию в декларативной форме. Эта система достаточно тяжеловесна и требует большого количества специальных знаний. Её можно порекомендовать для крупных промышленных или исследовательских проектов.

Eclipse EuGENia (<http://eclipse.org/gmt/epsilon/doc/eugenia/>) является системой, позволяющей сохранить простое простым. Для примитивных графических языков она позволяет генерировать готовые описания Eclipse GMF (<http://www.eclipse.org/gmf/>), прилагая минимум усилий.

Как GEMS, так и Eclipse EuGENia требуют от пользователя навыков программирования, т.е. быть продвинутым пользователем для их использования не достаточно.

Meta-CASE средство QReal

QReal (<http://se.math.spbu.ru/SE/qreal/>) — средство, разрабатываемое в СПбГУ, поддерживающее достаточно простое метамоделирование и создание новых типов диаграмм [5]. Построенные на базе QReal CASE-пакеты, помимо, собственно, метамоделей и редакторов диаграмм поддерживают стандартный интерфейс для обработки моделей, на основе которого можно легко строить генерационные решения.

Как и Eclipse, QReal является продуктом с открытым исходным кодом. Пакет изначально ориентирован на метамоделирование: штатный режим его работы подразумевает редактирование модели специфического визуального редактора (метамоделей), генерацию данного редактора, а следом его сборку и подключение к пакету для работы с описанным при помощи него графическим языком.

Инструмент является полноценным CASE-средством: сущности и связи, представленные на диаграммах, хранятся в репозитории, с которым могут работать сгенерированные редакторы.

Системные требования продукта невелики, кроме того, в отличие от Eclipse, он сразу устанавливается в комплекте с метаредактором, позволяющим редактировать собственные визуальные языки и генерировать специализированные редакторы. QReal реализован на C++ с использованием Qt, что позволяет ему работать приблизительно одинаково как в среде Windows, так и в UNIX-подобных средах (в данный момент поддерживаются версии для Linux).

Инструмент легко устанавливается и сразу содержит все необходимые для создания своих визуальных языков компоненты.

Справедливо отметить, что текущее активное развитие QReal не может не сказываться на его качестве для конечного пользователя. Однако это же позволяет рассчитывать и на быструю адекватную реакцию со стороны разработчиков инструмента на поступающие заявки.

Помимо лежащего в основе архитектуры инструмента принципа метамоделирования, следует упомянуть и то, что разработчики инструмента уделяют внимание особым качествам его пользовательского интерфейса, включая перспективные возможности работы с инструментом при помощи жестов устройства ввода, актуальные для набирающих популярность планшетных компьютеров и электронных досок [6, 3].

На основе QReal создан редактор программ для роботов на элементной базе Lego Mindstorms, при апробации он оказался доступен для лёгкого освоения школьниками 5–6 классов [10].

Создание редакторов для своих языков технически доступно для продвинутых пользователей Windows и Linux и вообще не требует специальных знаний программиста. Однако это является скорее приятной особенностью, а не ключевым свойством, так как сам процесс создания как визуальных, так

и «текстовых» языков подразумевает наличие у автора достаточно богатого практического опыта их использования.

Заключение

В образовании и промышленности мы имеем дело с большим количеством языков визуального моделирования, которые реализуются большим количеством инструментов. Многие из инструментов являются коммерческим ПО, и большинство из них предназначены для работы с заранее заданным языком, т.е. их можно отнести к «классическим» CASE-средствам.

В то же время, потребность в реализации собственных языков, как «текстовых», так и графических, есть, и за последние десятилетия она на спад не идёт.

Легко расширяемые инструменты, такие как, например, Dia и Visio, фактически представляют собой не CASE-средства, а векторные графические редакторы, предназначенные в первую очередь для рисования диаграмм, но не для моделирования. С их помощью можно определить нотацию («синтаксис») нового языка, но не его модель.

Обилие используемых языков влечёт за собой обилие используемых программных средств. Это повышает нагрузку на преподавателей и студентов и затрудняет усвоение материала, становящегося менее единообразным. Вместо компактного «университетского» курса, поддержанного одним-двумя инструментами, студенты должны потратить силы на изучение набора разнородного ПО, отвлекаясь от собственно предмета. Очевиден и другой недостаток, хотя и технический, но вполне весомый: обилие ПО усложняет системное администрирование и часто неоправданно увеличивает объёмы закупок.

На основании сказанного можно говорить о хороших перспективах использования в образовании достаточно простых, и, вместе с тем, достаточно мощных систем мета-CASE вообще, и, в частности, упомянутых систем на базе Eclipse и QReal.

Литература

1. *Бойко Н.И.* Рекомендации по преподаванию программной инженерии и информатики в университетах / Бойко Н.И., Зверинцева М.Е. и др. // Интернет-университет Информационных технологий, 2007. Пер. с англ. ACM Computing Curricula 2001
2. *Брыксин Т.А.* Среда визуального программирования роботов QReal:Robots / Брыксин Т.А., Литвинов Ю.В. // Материалы международной конференции «Информационные технологии в образовании и науке», Самара, 2011
3. *Кириленко Я.А.* Способы дополнительного повышения производительности труда в средах визуального проектирования / Кириленко Я.А., Осечкина М.С. // Материалы второй научно-технической конференции молодых специалистов

- «Старт в будущее», посвященной 50-летию полета Ю.А. Гагарина в космос. СПб. 2011. С. 225-227
4. *Кознов Д.В.* Языки визуального моделирования. Проектирование и визуализация программного обеспечения / Кознов Д.В. // Учебное пособие. Издательство СПбГУ, 2004.
 5. *Кузенкова А.С.* Средства быстрой разработки предметно-ориентированных решений в metaCASE-средстве QReal / Кузенкова А.С., Дерипаска А.О., Таран К.С., Подкопаев А.В., Литвинов Ю.В., Брыксин Т.А. // Научно-технические ведомости СПбГПУ, Информатика, телекоммуникации, управление. Вып. 4 (128). СПб.: Изд-во Политехнического Университета. 2011, С. 142–145.
 6. *Осечкина М.С.* Поддержка жестов мышью в meta-CASE-системах / Осечкина М.С., Брыксин Т.А., Литвинов Ю.В. и др. // Системное программирование. Вып. 5: Сб. статей / Под ред. А.Н.Терехова, Д.Ю.Булычева. — СПб.: 2010
 7. *Парфёнов В.В.* RTST — технология программирования встроенных систем реального времени / Парфёнов В.В., Терехов А.Н. // Системная информатика. Вып. 5: Архитектурные, формальные и программные модели. Новосибирск, 1997. С. 228–256.
 8. *Соколов Н.Е.* Подход к разработке распределенных гетерогенных реактивных систем / Соколов Н.Е., Луцив Д.В. // Материалы 2-й межвузовской научной конференции по проблемам информатики «СПИСОК-2011». — 2011. — С. 18–22. <http://spisok.math.spbu.ru/txt/SPISOK-2011.pdf>
 9. *Терехов А.Н.* Real: методология и CASE-средство для разработки систем / Терехов А.Н., Романовский К.Ю., Кознов Д.В., Долгов П.С., Иванов А.Н. // Программирование, 1999, N.5. С.44–52.
 10. *Филиппов С.А.* Робототехника для детей и родителей / Филиппов С.А. // М. Наука 2011.
 11. *Andrew Bragdon, Robert Zeleznik, Steven P. Reiss, Suman Karumuri, et. al.* Code Bubbles: A Working Set-based Interface for Code Understanding and Maintenance. Proceedings of the 28th International Conference on Human Factors in Computing Systems, 2010. http://www.andrewbragdon.com/codebubbles_site.asp
 12. *Timofey Bryksin, Yuri Litvinov, Valentin Onossovski, Andrey N. Terekhov.* Ubiq Mobile + QReal — a technology for development of distributed mobile services. To appear in Proceedings of the FRUCT 10 conference, Tampere, Finland, Nov. 2011
 13. *Krzysztof Czarnecki, Ulrich W. Eisenecker.* Generative Programming — Methods, Tools, and Applications. Addison-Wesley, June 2000.
 14. Eclipse GEMS metamodeling tutorial. http://wiki.eclipse.org/GEMS_Metamodeling_Tutorial
 15. *Adele Goldberg.* Introducing the Smalltalk-80 System. BYTE, Vol. 6, No. 8, Aug. 1981, McGraw-Hill. http://www.em.net/Byte_1981_08/Byte_1981_08_150dpi.pdf
 16. Integration Definition for Function Modelling (IDEF0). Draft Federal Information Processing Standards Publication 183, 1993.
 17. Integration Definition for Information Modelling (IDEF1X). Draft Federal Information Processing Standards Publication 184, 1993.
 18. *Richard J. Mayer, Ph. D. Christopher P. Menzel, et. al.* IDEF3 — AL-TR-1995-XXXX Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Sep. 1995.
 19. OMG MetaObject Facility 2.4.1, 2011. <http://omg.org/mof/>
 20. OMG Object Constraint Language Specification Ver. 2.0. <http://www.omg.org/spec/OCL/2.0/PDF/>

21. *Douglas T. Ross*. Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transactions on SE, Vol. SE-3, No. 1, Jan. 1977. <http://www.cs.txstate.edu/~rp31/papersSP/structuredAnalysisRoss.pdf>
 22. *Randall B. Smith, David Ungar*. Programming as an Experience: The Inspiration for Self. ECOOP '95 Conference Proceedings, Aarhus, Denmark, August, 1995. http://selflanguage.org/_static/published/programming-as-experience.pdf
 23. *W. Stevens, G. Myers, L. Constantine*. Structured Design. IBM Systems Journal, 13 (2), 115–139, 1974.
 24. OMG Unified Modelling Language Specification v. 2.4.1. Aug. 2011. <http://omg.org/>
 25. OMG MOF 2 XMI Mapping Specification ver. 2.4.1 <http://www.omg.org/spec/XMI/2.4.1>
 26. ITU Recommendation Z. 100: Specification and Description Language. 08/2002. 206 p.
-

Иерархические языки, фабрики вычислений, облачные технологии, мультимедиа сети и интернет-видеосвязь



**Прохоров
Владимир Валентинович**

д.ф.-м.н., профессор УрФУ
генеральный директор
ООО «Научно-производственный центр „Видикор“»

МЕНЕДЖЕР УПРАВЛЕНИЯ МУЛЬТИМЕДИА-СЕТЬЮ¹

Манакова И.П.

асп. УрФУ, iman@vidicor.ru

Аннотация: В работе рассматривается проблема функционирования сети доставки мультимедиа-контента (CDN) с ограниченными ресурсами её элементов. Выделена сущность «Менеджер управления CDN», требующая реализации. Указаны требуемые компоненты менеджера и их функции. Приведены результаты проектирования менеджера. Предложены алгоритмы управления нагрузкой мультимедиа-сети для менеджера.

1. Введение

Развитие мультимедиа-технологий, расширение и улучшение каналов связи повлияло на распространение мультимедиа-сетей. Вещание событий в реальном времени, просмотр записей из архивов на сетевых серверах и вещание интернет-телеканалов приобретают всё большую популярность.

Однако в настоящее время существуют проблемы, связанные с ограниченностью ресурсов элементов мультимедиа-сетей (скорости каналов связи, производительности узлов-маршрутизаторов и др.). Поэтому важной задачей при организации массового просмотра мультимедиа-данных через сеть Интернет является построение поверх IP-сети специфической мультимедиа-сети CDN (Content Distribution Network — сеть доставки контента).

Такая сеть должна решать вопросы управления доставкой мультимедиа-контента конечным потребителям (зрителям) в условиях ограничений на ресурсы элементов сети. Кроме того в этой сети должны решаться задачи распределённого хранения контента и транскодирования (тоже с ограниченными ресурсами). Должны также осуществляться мониторинг состояния сети, предсказание возможной нагрузки, устранение неполадок, выработка рекомендаций по развитию сети.

В работе приводится анализ существующих решений по управлению CDN. Предлагается программно-аппаратный комплекс «Менеджер управления CDN», который позволяет устранить ряд проблем, оговоренных ранее. Приводятся результаты проектирования менеджера. Предлагаются алгоритмы управления нагрузкой мультимедиа-сетей.

2. Представление мультимедиа-системы

Рассмотрим пример системы мультимедиа-вещания [1]. Она представляет собой мультимедиа-сеть (Рис. 1), включающую источники («стриме-

¹ Научный руководитель работы д. ф.-м. н., профессор УрФУ В.В. Прохоров. Работа выполнена при поддержке НПЦ «Видикор» и НТИ(ф) УрФУ.

ры» и «репликаторы»), хранилище-архив, зрителей медиаданных (клиентов) и транспортную сеть передачи данных.

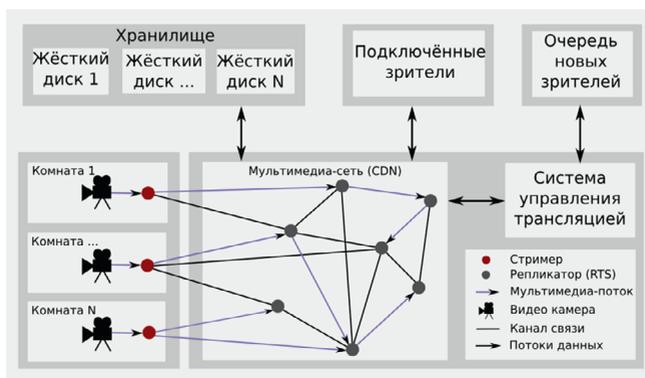


Рис. 1. Пример CDN

Основная задача данной системы — доставка мультимедиа-контента зрителю (конечному потребителю) через публичные интернет-каналы. На систему налагаются условия, связанные с ограниченностью ресурсов элементов мультимедиа-сетей (скорости каналов связи, производительности узлов-маршрутизаторов и др.).

Репликаторы выполняют функции как простой ретрансляции (передачи) потока, так и репликации — ретрансляции с размножением данных (поток, полученный от источника в одном экземпляре, может раздаваться сразу нескольким получателям). Мультимедиа-контент представляет собой две составляющие: мультимедиа-потоки, передаваемые в реальном времени (Streaming Media — живое/потокое медиа), и записи, которые хранятся на сервере (Media on Demand — медиа по запросу).

Рост количества зрителей может приводить к перегрузке элементов сети, что будет вызывать ухудшение качества приёма у клиентов. Это делает актуальным решение задачи автоматического (программного) управления системой в реальном времени. При этом система в целом должна быть построена иерархическим образом как совокупность вложенных подсистем, что целесообразно осуществлять с использованием мультипарадигменного подхода [2].

3. Анализ существующих подходов к управлению CDN

Исследователями и разработчиками предлагается ряд подходов к построению CDN. Скажем, концепция предлагаемая в [3], подразумевает построение системы из множества серверов, содержащих распределённые копии данных, которые клиенты могут копировать с «ближайшего» к ним

сервера. Даже такой подход является оптимизационным, поскольку нагрузка на систему распределяется между серверами; в нашем случае мультимедиа вещания нагрузка на стримеры распределяется между репликаторами (Рис. 1). Однако этот подход не рассматривает возможные ограничения на ресурсы элементов сети для разных топологий её построения и на динамику изменения топологии. В то же время учёт этих особенностей является важным при управлении нагрузкой мультимедиа-сетей. В связи с этим в известных публикациях предлагаются другие решения. Они опираются на два аспекта управления системой.

Первый аспект — раздача мультимедиа-контента клиентам, т.е. управление системой при условии, что количество получателей остаётся неизменным. При этом для решения проблемы распределения нагрузки между элементами (узлами и дугами графа сети) можно изменять маршруты следования мультимедиа-потокa [4]. Что активно используется при организации маршрутизации. Однако в ходе анализа существующих решений было установлено, что в случае построения мультимедиа-сети важен не столько выбор «кратчайшего пути» доставки контента (что обычно требуется от алгоритмов маршрутизации), сколько оптимальное распределение нагрузки между элементами в течение больших промежутков времени за счёт управления схемой маршрутизации. Алгоритмы маршрутизации не решают эту задачу.

Второй аспект — подключение новых клиентов, т.е. управление системой при условии роста количества потребителей. В этом случае необходимо определить, к какой из точек раздачи контента подключить нового клиента и как сформировать путь доставки контента в эту точку.

Наиболее часто используемыми методами распределения нагрузки между узлами считаются циклический алгоритм («Round Robin») и «взвешенное обслуживание» («Weighted Fair Queuing») [5]. Для специфики CDN предлагаются специальные решения. Кроме этого предлагаются решения с использованием протоколов сбора статистической информации о сети и готового программного и аппаратного обеспечения. Однако эти решения не учитывают многих аспектов мультимедиа-систем. Например, существование нескольких потоков данных на узлах мультимедиа-сети, разные схемы кодирования и декодирования данных и др.

Опираясь на сказанное выше, были сделаны следующие выводы по задаче улучшения работы мультимедиа-сети:

- при наличии программно-аппаратного комплекса по доставке мультимедиа-контента зрителям есть проблема оптимального управления таким комплексом;
- на данный момент нет готового решения по управлению, которое учитывало бы все аспекты мультимедиа-систем;
- для управления мультимедиа-сетью необходим дополнительный программно-аппаратный комплекс; в качестве такого комплекса предлагается «Менеджер управления CDN».

4. Менеджер управления CDN

Согласно анализу, представленному в п. 3, были выделены следующие задачи, которые должен решать «Менеджер управления CDN»: оптимизация подключения новых клиентов, сбор, хранение, отображение статистической информации о системе вещания, управление нагрузкой на аппаратное обеспечение узлов и на каналы мультимедиа-сети (Рис. 2).



Рис. 2. Менеджер управления CDN

Указанные на Рис. 2 компоненты можно охарактеризовать следующим образом:

«**Менеджер подключения пользователей**» активируется в том случае, если в систему поступает запрос на подключение нового клиента. Данный компонент вырабатывает оптимальный план подключения, отвечая в каждый данный момент времени на вопросы: «возможно ли подключение данного клиента?», «какой из имеющихся возможностей воспользоваться, чтобы подключить клиента?» и «к какой точке подключить клиента?».

«**Менеджер сбора и хранения данных о мультимедиа-системе**» предполагает несколько режимов активации: круглосуточный мониторинг, активация «по требованию» (например, раз в день, раз в час, раз в минуту). Этот компонент вырабатывает оптимальный план сбора и хранения статистической информации о системе в целом, узлах, клиентах, отвечая на вопросы: «как и когда собирать данные о системе?», «где хранить собранные данные?», «какие данные необходимы для сбора и хранения?», «какие возможности обработки данных необходимо осуществить?».

«**Менеджер отображения статистической информации**» взаимодействует с теми данными, которые удалось получить во время работы «Менеджера сбора и хранения данных о мультимедиа-системе». Он может активироваться «по требованию» или же работать в реальном времени. Менеджер должен выработать (выбрать) оптимальную стратегию (алгоритм) отображения статистической информации. Основные вопросы, которые решает этот компонент — «в каком виде отобразить данные из хранилища?», «какие данные необходимо отобразить в указанный момент времени?», «какие методы статистической обработки информации необходимо использовать?», «какой пользовательский интерфейс необходимо предоставить?».

«**Менеджер управления нагрузкой мультимедиа-системы**» активируется тогда, когда на участках системы возникает повышенная нагрузка.

Менеджер должен выработать (выбрать) оптимальную стратегию (алгоритм) снижения нагрузки на точку (точках). Основные вопросы, которые должна решать данный компонент – «каким образом разгрузить точку(и) во время трансляции?», «каким образом разгрузить точку(и) во время подключения нового клиента?».

Поскольку задачи, которые должны решать указанные на Рис. 2 сущности, оговорены, далее приводится краткое описание работ, проведённых в каждом из направлений.

4.1. Менеджер подключения пользователей

Результаты по моделированию менеджера подключения пользователей приведены в [6, 7]. Алгоритм подключения новых пользователей включает в себя следующие составляющие (Рис. 4):

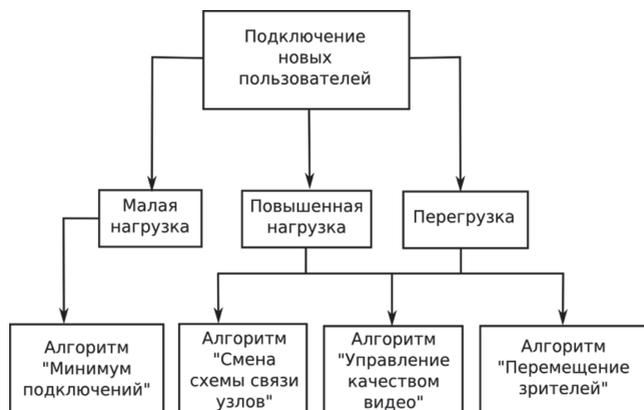


Рис. 4. Подключение новых клиентов

Алгоритм «Минимум подключений» широко используется при управлении нагрузкой в многосерверных системах. Алгоритмы «Управление качеством видео» и «Перемещение зрителей» предлагаются автором в [6, 7]. Алгоритм «Смена схемы связи узлов» в известных публикациях не рассматривается и требует изучения.

4.2. Менеджер управления нагрузкой мультимедиа-системы

Задачи, которые решает «Менеджер управления нагрузкой мультимедиа-системы», во многом схожи с задачами, которые решает «менеджер подключения пользователей» при повышенной нагрузке и перегруженности. Их решение должно снимать проблему подключения к системе дополнительного клиента, т. е. они являются частным случаем более общей задачи,

возможности управления системой также включают в себя «Смену схемы связи узлов», «Управление качеством видео», «Перемещение зрителей».

4.3. Сбор, хранение, отображение статистической информации

Комплексное наблюдение за системой является одним из важных условий качественного управления. Оно необходимо для экстренного реагирования, предсказания поведения клиентов и будущей нагрузки. Имеется достаточно много специализированных средств мониторинга сетей и сетевых узлов; интересная таблица их сравнения приведена в [8]. Однако в этих системах учитываются не все аспекты мультимедиа-сетей, необходима разработка дополнительного программного комплекса.

5. Дальнейшие направления исследований

В работе был приведён анализ существующих решений по управлению нагрузкой в CDN. Установлено, что существующие подходы не учитывают всех особенностей мультимедиа-сетей, работающих в условиях существования ограничений на ресурсы элементов сети. В связи с этим был предложен дополнительный программно-аппаратный комплекс «Менеджер управления CDN». В работе обозначен круг задач, которые должен решать менеджер. Были приведены результаты исследований, описаны алгоритмы по управлению нагрузкой мультимедиа-сетей. Дальнейшие направления исследований включают в себя:

- исследование вопроса изменения схемы связи узлов во время раздачи мультимедиа-контента и подключении новых клиентов;
- исследование вопроса сбора, хранения, отображения информации о мультимедиа-сети.

В результате работ будет реализован и внедрён в существующую мультимедиа-сеть «Vidicor Video System» [1] «Менеджер управления CDN».

Автор благодарит В.В. Прохорова за постановку задачи и оказанную помощь.

Литература

1. Интернет-видеосистема Vidicor Video System. Официальный сайт продукта. — [Электронный ресурс]. — Режим доступа: <http://vidicor.ru> (дата последнего обращения: 01.04.2013).
2. Prokhorov, V.V. On Microcontext Approach to the construction of Knowledge Representation Languages and Human-Computer Interaction//Journal of Computer and Systems Sciences International, 1997, № 5. — P.5–16.
3. Риз, Дж. Облачные вычисления: Пер. с англ. — СПб: БХВ-Петербург, 1201. — 288 с.: ил. ISBN 978-5-9775-0630-4, с. 262.

4. *Gerald, R. Ash*, Prosper Chemouil 20 Years of Dynamic Routing in Circuit-Switched Networks. — [Электронный ресурс] — Режим доступа: http://perso.rd.francetelecom.fr/chemouil/gcn_ieee/DynRout20.pdf (дата последнего обращения: 01.04.2013).
 5. *Носков В., Криницын К., Пономарёв А.* Балансировка масштабируемых приложений. Открытые системы №08, 2012. — [Электронный ресурс]. — Режим доступа: <http://www.osp.ru/os/2012/08/13019244/> (дата последнего обращения: 01.04.2013).
 6. *Манакова И.П., Петров К.Б.* Распределение пользователей по видеосерверам онлайн трансляции с условием минимального перемещения зрителей. Технические науки — от теории к практике: Материалы X международной заочной научно-практической конференции (28 мая 2012 г.) [под ред. Я.А. Полонского]. Новосибирск: Изд. «Сибирская ассоциация консультантов», 2012. — 27–35 с.
 7. *Манакова И.П., Петров К.Б.* К вопросу о подключении пользователей к мультимедиа-сети. «Инновации науке»: материалы XVI международной заочной научно-практической конференции. Часть 1 (28 января 2013 г.); Новосибирск: Изд. «СибАК», 2013, 94–108 с.
 8. Comparison of network monitoring systems. — [Электронный ресурс]. — http://en.Wikipedia.org/wiki/Network_monitoring_comparison (дата последнего обращения: 01.04.2013).
-

Мультиагентные технологии и их приложения в информатике



**Тимофеев
Адиль Васильевич**

Д.Т.Н.
профессор
заведующий лабораторией информационных технологий
в управлении и робототехнике СПИИРАН

МЕТОДИКА ОБРАБОТКИ СИГНАЛА ОДНОКОМПОНЕНТНОГО ДАТЧИКА МОМЕНТА ДЛЯ ПРИМЕНЕНИЯ В СИЛОМОМЕНТНОМ УПРАВЛЕНИИ ЭЛЕКТРОМЕХАНИЧЕСКИМ ШАРНИРОМ

Титов В.В.

программист ЦНИИ РТК, victortitov2005@mail.ru

Шардыко И.В.

программист ЦНИИ РТК, ddeer@bk.ru

Даляев И.Ю.

инженер ЦНИИ РТК, mehatroniks@gmail.com

Аннотация: Статья посвящена калибровке датчика момента, применяемому в силомоментном управлении исполнительным модулем. Предложены способы расчётной компенсации паразитных составляющих сигнала по моменту, вызванных механическими особенностями установки датчика в модуле.

Введение

Для того чтобы иметь возможность управления электромеханическим исполнительным модулем (шарниром) в силомоментном режиме, необходимо иметь информацию о внешнем моменте, приложенном к выходному валу шарнира [1–3], в то время как фактически на интегрированном в шарнир датчике момента проявляются различные механические эффекты, приводящие к появлению паразитных моментных составляющих.

Исполнительный электромеханический шарнир предназначен для перемещения и позиционирования полезной нагрузки, присоединенной к выходному звену, вокруг своей оси с заданным законом управления (положением, скоростью, моментом). Шарнир состоит из следующих основных составляющих: бесколлекторного двигателя постоянного тока, датчика положения ротора двигателя, волнового редуктора, датчика абсолютного положения на базе вращающегося трансформатора, подвижного и неподвижного корпусов, несущих подшипников, платы привода, стыковочных конусных фланцев с электрическими соединителями и однокомпонентного датчика момента, фланец которого представлен на рисунке 1.

Недостатки фактической информации, получаемой с датчика момента

При работе шарнира момент с вала двигателя передается на генератор волнового редуктора. Генератор деформирует гибкое зубчатое колесо волно-



Рис. 1. Фланец однокомпонентного датчика момента

вого редуктора чашечного типа, которое входит в зацепление с жестким колесом редуктора. Момент с гибкого колеса через вал передается на внутренний обод датчика момента [4] и через три гибких спицы — на его внешний обод и, далее, на подвижный корпус. Плата датчика момента обрабатывает информацию от тензорезисторов, закрепленных на гибких спицах датчика.

При включении шарнира в режиме холостого хода (внешний момент отсутствует) с заданной постоянной скоростью, сигнал, получаемый с датчика момента, имеет вид, показанный на рисунке 2.

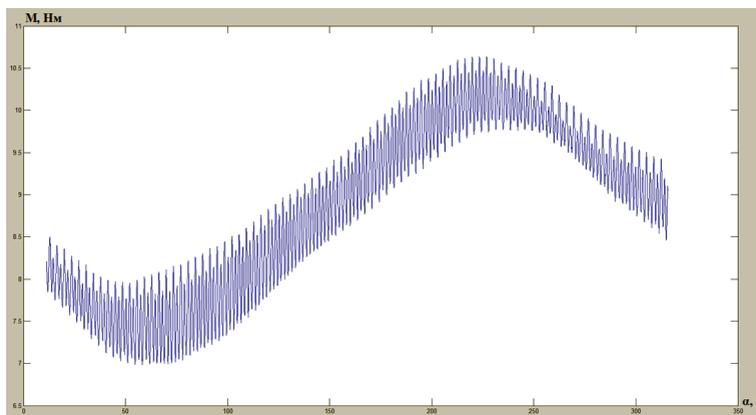


Рис. 2. Выходной сигнал датчика момента

Как видно из графика, сигнал обладает двумя явными паразитными составляющими — низкочастотной и высокочастотной. Низкочастотная составляющая имеет форму гармонического колебания с периодом, соответствующим обороту выходного вала. Высокочастотная составляющая имеет сложную форму, однако в определённой степени повторяет свою форму с постоянным периодом, равным $3,6^\circ$. С учётом того, что передаточное отношение волнового редуктора данного шарнира $i=100$, период высокочастотной составляющей соответствует одному обороту двигателя либо перемещению гибкого колеса волнового редуктора на два зуба. Чтобы рассмотреть данную составляющую подробнее, на рисунке 3 приведён участок графика в увеличенном виде.

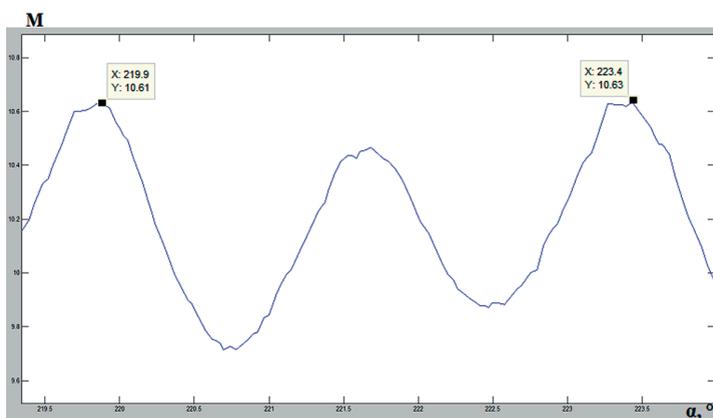


Рис. 3. Фрагмент сигнала с датчика момента

По представленному графику можно предположить, что два явно выраженных максимума на одном периоде данной помехи соответствуют двум зубьям, на которые перемещается за это время гибкое колесо волнового редуктора. Опытным путём установлена высокая повторяемость данной кривой за серию испытаний шарнира на холостом ходе.

Низкочастотную составляющую можно объяснить неточностью закрепления выходного вала шарнира и вместе с ним датчика момента — несоосностью подшипниковой опоры. Повторяемость данной кривой также подтверждена испытаниями.

Варианты коррекции сигнала от датчика момента

Коррекция сигнала осуществляется вычитанием из сигнала, поступающего с датчика момента, значений, рассчитанных нижеописанными способами, основанными на методах, описанных в [5].

*Коррекция, основанная на измерении периода
высокочастотного паразитного сигнала*

При данной коррекции каждая составляющая корректируется независимо. Во-первых, на основании полученной информации (рисунок 2) по методу наименьших квадратов полученная кривая аппроксимируется на всём промежутке изменения аргумента полиномом высокой степени вида

$$P_i(x) = \sum a_i x^i \quad (1)$$

Для коррекции используется совокупность точек, соответствующая данному полиному (значениям a_i) с фиксированным шагом по оси угла. При значениях угла поворота шарнира, попадающих в диапазон между некоторыми двумя заданными точками, значение момента определяется линейной интерполяцией, рассчитываемой как

$$M(x) = M(x) + [M_{i+1}(x) - M_i(x)] \cdot [x_{i+1} - x_i] / [x - x_i], \quad (2)$$

где x , x_i , x_{i+1} — текущее положение, левая и правая границы диапазона соответственно; $M(x)$, $M(x)$, M_{i+1} — значения коррекции для перечисленных положений соответственно.

Для компенсации высокочастотной составляющей используются данные, снятые на одном из периодов. Компенсация проводится по описанному выше принципу. Рассчитанные для одного периода значения распространяются по периодическому закону на всю ось угла.

*Коррекция, построенная по принципу
кусочно-линейной интерполяции*

В этом случае обе составляющие компенсируются одним набором данных. В этот набор входят все локальные максимумы и минимумы кривой, представленной на рисунке 2. При значениях угла поворота шарнира, попадающих в диапазон между некоторыми двумя заданными точками, значение момента вновь определяется линейной интерполяцией.

Двухуровневая непериодическая коррекция

Данная коррекция сочетает два предыдущих варианта. Сначала из сигнала, поступающего с датчика, вычитается низкочастотная составляющая аналогично первому варианту коррекции, а уже в полученном сигнале, как и в случае второго варианта, формируется набор локальных максимумов и минимумов с линейной интерполяцией участков между ними. Этот вариант иллюстрируется рисунком 4.

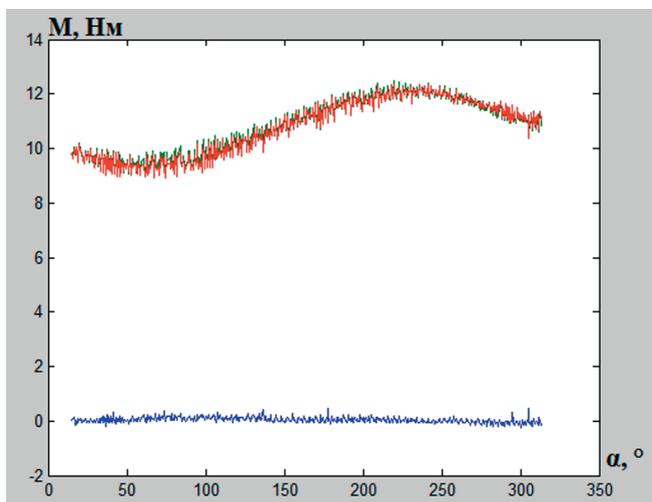


Рис. 4. Результаты коррекции

На представленном рисунке кривая красного цвета соответствует исходному сигналу с датчика момента, зелёная — значениям сигнала коррекции, красная — скомпенсированным значениям сигнала, поступающего в качестве сигнала обратной связи систему управления. Таким образом, достигается снижение амплитуды паразитного аддитивного сигнала в 3 раза по высокочастотной и в 20 раз по низкочастотной составляющей (амплитуда паразитного сигнала, оставшаяся после коррекции не превышает 1% от диапазона измерения датчика).

Заключение

В статье обозначена задача выделения полезного сигнала с датчика момента путем введения коррекции аддитивной составляющей паразитного сигнала различными способами. Показано, что применение коррекции позволяет в несколько раз снизить паразитный сигнал и повысить точность датчика. Достигнутая точность датчика позволяет использовать информацию с него в качестве обратной связи для управления шарниром в силомоментном режиме.

Литература

1. http://www.lorenz-messtechnik.de/english/files/publications/Management_Systems_Calibration_Torque.pdf

2. http://www.lorenz-messtechnik.de/english/company/torque_measurement_technology.php
 3. <https://measurementsensors.honeywell.com/ProductDocuments/Manuals/1800%20Operating-Service%20Manual.pdf>
 4. *И.Ю. Даляев.* «Датчик выходного момента шарнира». Экстремальная робототехника // Сборник докладов Всероссийской научно-технической конференции. — Санкт-Петербург: Изд-во «Политехника-сервис», 2011. — 454с.;
 5. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. М.: Наука, 1973 — С. 683–694.
-

ПРОБЛЕМЫ И МЕТОДЫ МНОГО-АДРЕСНОЙ МАРШРУТИЗАЦИИ В ГЛОБАЛЬНЫХ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ НОВОГО ПОКОЛЕНИЯ

Бакурадзе А.М.

*ведущий электроник лаборатории информационных технологий
в управлении и робототехнике СПИИРАН, klauzert@yandex.ru*

Аннотация: Описываются основные проблемы и методы много-адресной маршрутизации в глобальных телекоммуникационных системах нового поколения.

Введение

Бурное развитие глобальных телекоммуникационных систем (ТКС) свидетельствует о том, что их колоссальные возможности по массовому доступу людей к корпоративным, региональным и мировым информационным и вычислительным ресурсам уникальны и впечатляющи. Однако дальнейшее развитие глобальных ТКС (создание ТКС новых поколений, GRID-системы и т. п.) в первую очередь связано с совершенствованием, автоматизацией, оптимизацией и интеллектуализацией систем сетевого управления потоками данных.

Причиной этого является тот факт, что сегодня сетевое управление глобальными ТКС в значительной степени зависит от «человеческого фактора», а именно от знаний, опыта, интуиции и воли сетевых администраторов и операторов. В то же время хорошо известно, что способности людей (в том числе профессионалов в области сетевого управления ТКС) принципиально ограничены пределами их психофизиологических возможностей. Даже если будет разработан наилучший (в том или ином смысле) человеко-машинный интеллектуальный интерфейс, сетевые администраторы и операторы не способны эффективно управлять многомерными сложными параллельными процессами передачи и обработки мультимедийных потоков данных. Поэтому они не смогут обеспечить высокое качество сетевого управления, надёжность и отказоустойчивость глобальных ТКС. Однако на современном этапе развития телекоммуникационных и информационных технологий особенно важно гарантировать пользователям глобальных ТКС высокое качество запрашиваемых услуг и информационно-вычислительных ресурсов.

Эффективным путём совершенствования сетевого управления потоками данных в глобальных ТКС является его автоматизация на базе динамических моделей ТКС как сложных объектов управления с переменной структурой, методов оптимизации процессов маршрутизации потоков данных и принципов адаптивного и интеллектуального управления с использованием нейросетевых, мульти-агентных и GRID-технологий. На этом новом пути возможен как учёт

реальной динамики ТКС, т. е. фактического изменения структуры (топологии узлов и каналов связей) и параметров (весов каналов связи) ТКС, так и адаптация к различным факторам неопределённости и нестационарности.

1. Задача и режимы много-адресной (групповой) маршрутизации

В современных ТКС существует три основных принципа маршрутизации передаваемых информационных потоков:

- одно-адресная (unicast) маршрутизация;
- много-адресная (multicast) маршрутизация;
- широковещательная (broadcast) маршрутизация;

Одно-адресная маршрутизация осуществляется в случае, когда заданы один узел-источник и один узел-получатель глобальной ТКС.

Много-адресная маршрутизация реализует один из двух характерных режимов групповой передачи данных:

- «один источник-группа получателей», когда задан один узел-источник и несколько узлов-получателей,
- «группа источников — один получатель», когда задан один узел-получатель и несколько узлов-источников.

При широковещательной (лавинной) маршрутизации пакеты передаются от одного узла ТКС ко всем остальным.

Много-адресная маршрутизация с одним узлом-получателем и несколькими узлами-источниками может рассматриваться как модификация одно-адресной маршрутизации, при которой от каждого узла-источника строится свой маршрут. В этом режиме много-адресной маршрутизации содержимое передаваемых пакетов может быть своим для каждого узла-источника из группы. Это означает, что потоки данных от группы узлов ТКС могут рассматриваться как самостоятельные потоки, параллельно передаваемые узлу-получателю ТКС.

Важно отметить, что в этом режиме совокупность локально-оптимальных маршрутов, независимо определяемых каждым узлом-источником, соответствует глобально-оптимальному решению задачи много-адресной маршрутизации для всей группы узлов-источников глобальной ТКС.

Особенность много-адресной маршрутизации в другом характерном режиме, когда потоки данных передаются от одного узла-источника к группе узлов-получателей ТКС, состоит в том, что потоки данных, принимаемых узлами-получателями, будут одинаковыми.

Графовым аналогом решения этой задачи будет дерево оптимальных маршрутов от узла-источника к группе узлов-получателей ТКС. При этом интенсивности потоков данных на всех каналах связи, соответствующих ребрам этого дерева, будут равными (в случае, когда каналы связи ТКС обладают неограниченной пропускной способностью).

Таким образом, при геометрико-графовом подходе к много-адресной маршрутизации во втором режиме, интенсивность потока данных не имеет значения. Поэтому её можно исключить из рассмотрения, что значительно облегчает решение задачи много-адресной маршрутизации потоков данных в глобальных ТКС в режиме «один источник — группа получателей».

2. Формализация задачи много-адресной маршрутизации в характерных режимах групповой передачи данных

Постановка задачи маршрутизации потоков данных в глобальной ТКС в режиме «группа источников — один получатель» заключается в следующем.

Пусть заданы граф ТКС $G(A, R, W)$, группа узлов-источников $s_i \in A$, $i = (1, 2, \dots, n)$ и один узел-получатель $f \in A$, причем $f \neq s_i$. Требуется построить дерево кратчайших маршрутов $T_{\text{опт}}$, такое, что стоимость любого маршрута

$$s_i \rightarrow f, \quad i = 1, 2, \dots, n, \quad (1)$$

входящего в $T_{\text{опт}}$, будет минимальной из всех возможных маршрутов из s_i в f .

Формализация задачи маршрутизации в режиме «один источник — группа получателей» заключается в следующем.

Пусть ТКС соответствует граф $G(A, R, W)$ и заданы узел-источник $s_0 \in A$ и группа узлов-получателей $f_i \in F \subset A$, $i = 1, 2, \dots, n$, причём $s_0 \neq f_i$. Требуется построить дерево кратчайших маршрутов $T_{\text{опт}}$, такое, что стоимость любого маршрута (1), входящего в $T_{\text{опт}}$, будет минимальной из всех возможных маршрутов из s_0 в f_i .

Таким образом, решением данной задачи является поддерево дерева кратчайших маршрутов, построенного от узла-источника к остальным узлам глобальной ТКС.

3. Особенности много-адресной маршрутизации в режиме «один источник — группа получателей»

Маршрутизация в этом режиме «от одного ко многим» требует от маршрутизатора построения оптимальных или субоптимальных маршрутов от заданного узла-источника ко всем узлам-получателям из заданной группы (множества) узлов ТКС.

Группы узлов-получателей формируются, например, из групп пользователей ТКС «по интересам». Они могут включать достаточно большое число узлов. Однако это число (мощность группы) должно быть существенно меньшим, чем общее число узлов (мощность) глобальной ТКС.

Каждой группе узлов ТКС можно поставить в соответствие её групповой адрес. Такие группы могут получать потоки запрошенных данных как из одного (единственного) узла-источника, так и из разных источников. При этом число узлов ТКС в группе может изменяться.

Для устранения перегрузки сетевого трафика и улучшения приёма потоков данных при много-адресной маршрутизации каждый узел-получатель может послать маршрутизатору ТКС запрос на резервирование или приоритетное обслуживание определённых каналов связи. Для этого он может использовать, например, обратную маршрутизацию потоков данных.

В этом случае маршрутизатор централизованно задаёт (через соответствующий протокол) полные маршруты передачи потоков данных от узла-источника ко всем узлам-получателям через все промежуточные узлы ТКС.

Формально метод маршрутизации в режиме «от одного ко многим» можно отнести к так называемой маршрутизации потоков данных в режиме «маршрутизации от источника» (Source Routing — SR).

Однако этот термин, вообще говоря, имеет более широкий смысл. Например, он подразумевает также распределённую маршрутизацию, когда каждый маршрутизатор отвечает только за один следующий шаг передачи данных, а весь маршрут является результатом работы всех маршрутизирующих узлов ТКС. Такой принцип распределённой маршрутизации обычно используется в динамических и адаптивных маршрутизаторах.

Важно отметить, что в новой версии IP-протокола IPv6 в глобальной сети Internet наряду с распределённой маршрутизацией предусмотрена и SR-«маршрутизация от источника».

Предельными случаями маршрутизации в режиме «от одного ко многим» (в частности, «от одного к одному») является лавинная («от одного ко всем») маршрутизация.

В случае лавинной маршрутизации входные пакеты данных передаются от узла-источника по всем выходным каналам связи ко всем остальным узлам ТКС. При этом маршрутизатор «заполняя» протокол, устанавливает в заголовке пакета данных счётчик пройденных каналов связи ТКС для каждого маршрута. В начале показания этого счётчика равны длине оптимального маршрута или диаметру (длине максимального маршрута) ТКС, а затем они уменьшаются на 1 после передачи пакета данных по заданному маршруту через очередной канал ТКС.

Основными областями применения алгоритмов и протоколов лавинной маршрутизации являются следующие приложения:

- военные (оборонные) ТКС, в которых значительная часть узлов и каналов связи может быть уничтожена;
- одновременное обновление распределённых БД и БЗ в КС, обслуживаемых глобальной ТКС;
- тестирование алгоритмов маршрутизации различных типов в процессе имитационного моделирования ТКС.

4. Модификация алгоритма Дейкстры для много-адресной маршрутизации в режиме «от одного ко многим»

Пусть узел-источник s_0 и множество (группа) узлов-получателей $\Psi = \{F\}_{i=1..K}$ зафиксированы. Тогда задача много-адресной маршрутизации заключается в построении дерева кратчайших маршрутов от s_0 ко всем узлам из группы узлов-получателей $F_i \in \Psi$ и определении на каждом узле дерева распределения потока данных для каждой группы. Не умаляя общности, будем считать, что этими узлами из множества (группы) Ψ будут все узлы ТКС, кроме узла s_0 ($A \setminus s_0$).

Предлагаемый алгоритм является модификацией статического алгоритма Дейкстры на случай много-адресной маршрутизации в режиме передачи потоков данных «от одного ко многим».

Введём исходные данные и необходимые обозначения:

$A_i \subset A$ — множество соседних a_i узлов;

$T_0 \subset A$ — множество узлов, обработанных алгоритмом;

$T_1 \subset A$ (T_0 и $T_1 - \text{dis}$) — множество узлов, обрабатываемых алгоритмом;

$T_2 = A \setminus (T_0 \cup T_1)$ — остальные узлы ТКС;

$w(i, j)$ — стоимость ребра $(i, j) \in f$; $w(i, i) = 0$; $w(i, j) = \infty$, если узлы a_i и a_j не соединены каналом связи непосредственно; $w(i, j) \geq 0$, если эти два узла непосредственно соединены;

$L(j)$ если $j \in T_1$, — оценка алгоритмом стоимости маршрута от узла-источника s_0 до узла n ;

$L(j)$ если $j \in T_0$, — стоимость кратчайшего маршрута от s_0 до n . (Для остальных j это значение $L(j)$ не определено).

Требуется построить дерево кратчайших маршрутов от узла-источника s_0 до остальных узлов ТКС и вычислить их стоимости, а также для каждого узла $a_i \in A$ и группы узлов-получателей F определить множество $\theta_i(F) \subset A_i$, соответствующее распределению потока от s_0 к F через узел a_i .

Алгоритм решения этой задач заключается в следующем:

1. Инициализация.

T_1 — пусто, $T_1 = \{s_0\}$, т.е. вначале множество обработанных узлов ТКС состоит только из узла-источника. Узлу-источнику сопоставляется пустой маршрут.

$L(n) = w(s_0, n)$ для $n \neq s_0$, т.е. начальные стоимости маршрутов к соседним узлам — это веса соответствующих ребер графа G , $L(s_0) = 0$.

Все $\theta_i(F)$ пусты.

2. Шаг алгоритма.

Выберем из T_1 узел ТКС с минимальной оценкой стоимости маршрута и поместим его в T_0 , обозначая соответствующий ему маршрут как кратчайший. Иными словами, находим

$$x \in T_1 : L(x) = \min_{j \in T_1} L(j). \quad (2)$$

При этом рассматриваем все рёбра графа ТКС с начальным узлом x .

Для каждого ребра (x, j) выполняется одно из трех действий:

А) если $j \in T_0$, то никаких действий не производится;

В) если $j \in T_2$, то j переводится из множества T_2 в T_1 , причём j -ому узлу сопоставляется значение

$$L(j) = L(x) + w(x, j), \quad (3)$$

а также маршрут, состоящий из кратчайшего маршрута до x и ребра (x, j) .

С) если $j \in T_1$, то производится новая оценка

$$L(j) = \min(L(x) + w(x, j)), \quad (4)$$

и j -ому узлу сопоставляется соответствующий оценке маршрут (либо прежний, либо согласно в В)).

Для каждого узла a_i , составляющего полученный кратчайший маршрут, кроме узла x , и всех групп узлов-получателей F , содержащих x , добавляем в $\theta_i(F)$ узел a_j , такой, что ребро (a_i, a_j) входит в кратчайший до x маршрут.

Итерации алгоритма повторяются до тех пор, пока T_0 не станет равно s .

Таким образом, по окончании работы алгоритма для каждого узла x ТКС будет определен кратчайший маршрут к нему от узла-источника, а значение $L(x)$ будет соответствовать стоимости этого маршрута.

5. Особенности нейросетевого подхода к много-адресной маршрутизации

Математические модели нейронных сетей являются удобным вычислительным механизмом, обладающим следующими важными свойствами:

- обучаемость,
- адаптивность,
- высокий параллелизм и скорость вычислений
- предсказуемая вычислительная устойчивость (робастность), т. е. способность находить точные решения при неточных входных данных.

Аппаратные и некоторые программные реализации нейронных сетей позволяют в значительной степени повысить эффективность и скорость сложных расчетов, проводимых при решении широкого класса вычислительных задач (в частности, задач маршрутизации потоков данных в ТКС).

Задача много-адресной маршрутизации в ТКС является, по существу, основной задачей управления потоками данных. Это объясняется тем, что решение этой задачи существенно используются для организации распределенного управления потоками данных, организации совместной работы узлов-маршрутизаторов, а также для передачи мультимедийных данных с высокими QoS-параметрами качества предоставляемых услуг (например, управление мультимедийными потоками данных во время сетевых видеоконференций).

Именно поэтому методы синтеза адаптивных, эффективных (экономных по времени) и надежных алгоритмов маршрутизации потоков данных в глобальных ТКС и их нейросетевое представление имеют первостепенное значение в задачах управления передачей данных.

Нейронные сети как вычислительные модели потенциально соответствуют всем этим требованиям к алгоритмам и средствам маршрутизации информационных потоков в глобальной ТКС.

Нейросетевые алгоритмы определяются нейросетевой постановкой задачи маршрутизации (точнее её нейросетевой интерпретацией) и некоторой подходящей моделью нейронной сети, решающей задачу в данной постановке.

Рассмотрим задачу много-адресной маршрутизации в нейросетевой постановке. В качестве модели ТКС будем рассматривать граф $G(A, R, W)$. Рассмотрим режим маршрутизации «от одного ко многим» узлам ТКС.

Пусть заданы узел-источник $s \in A$ и множество (группа) узлов-получателей

$$F = \bigcup_i f_i, f_i \in A, \quad i = 1, 2, \dots, n. \quad (5)$$

Множество данных (2.5) образуют пространство входных данных, подаваемых на вход нейронной сети.

Требуется построить дерево минимальной стоимости $Y_{\min} \subset Y$, состоящее из оптимальных маршрутов от узла-источника ко всем узлам-получателям ТКС.

Множество таких маршрутов образуют пространство решений, формируемых на выходе нейронной сети.

В качестве нейросетевой модели для решения задачи много-адресной маршрутизации рассмотрим модификацию нейронной сети Хопфилда.

Нейронные сети Хопфилда являются однослойными сетями с обратными связями от каждого нейрона к входам всех остальных. Они эффективно используются для решения задач дискретной оптимизации (задача коммивояжера и т. п.), распознавания образов и фильтрации сигналов.

Общий вид сети Хопфилда показан на рис. 2.1. Входной сигнал для i -го нейрона здесь обозначен через x_i , выходной сигнал — через y_i , вес связи от i -го нейрона к j -му обозначен через T_{ij} .

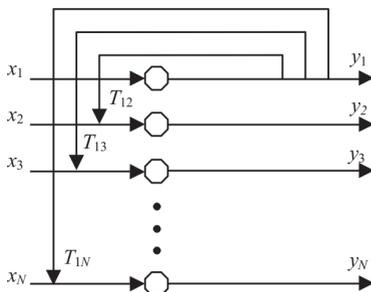


Рис. 2.1. Архитектура нейронной сети Хопфилда

6. Метод построения нейросетевой модели маршрутизатора

Для построения нейронной сети Хопфилда необходимо последовательно выполнить следующие шаги (этапы синтеза):

- 1) определить функцию активации для нейронов;
- 2) определить пространство входных данных для нейронной сети (оно может не совпадать с пространством решений задачи в нейросетевой постановке, и в этом случае потребуется дать интерпретацию решения, т. е. привести решение, полученное нейронной сетью к решению в терминах нейросетевой постановки задачи маршрутизации);
- 3) на объединении пространства входных данных и пространства решений задать энергетическую функцию нейронной сети вида

$$E(x, y) = - \sum_{i,j=1}^N T_{ij} y_i y_j - \sum_{i=1}^N x_i, \quad (6)$$

обладающую следующими свойствами:

- a) функция E является положительно определённой квадратичной формой,
 - b) глобальный минимум функции E соответствует решению поставленной задачи.
- 4) в соответствии с энергетической функцией (2.6) построить модель нейронной сети;

Полученная нейронная сеть будет способна решать поставленную задачу с определенной степенью точности. Точность решения определяется особенностями поставленной задачи и наличием локальных минимумов энергетической функции. Решением сети Хопфилда является совокупность выходных значений всех её нейронов.

В качестве функции активации i -го нейрона выберем нелинейную функцию вида

$$g_i(z) = \frac{1}{1 + e^{-\lambda_i z}}, \quad (7)$$

где z — суммарный входной сигнал i -го нейрона, а λ_i — некоторый положительный коэффициент — параметр i -го нейрона.

Такая функция активации позволяет рассматривать динамику нейронной сети, т. е. описать её поведение во времени.

Пусть узлы графа ТКС проиндексированы, т. е. $A = \{a_i\}$, $i = 1, 2, \dots, N$, где N — число узлов в ТКС. В качестве пространства решений сети Хопфилда Y будем рассматривать следующее множество:

$$Y = \bigcup_{i,j,m} y_{ij}^m, \quad \text{где } y_{ij}^m = \begin{cases} 1, & \text{если маршрут от узла-источника } s \text{ к узлу-} \\ & \text{получателю } a_m \text{ содержит ребро } (a_i, a_j) \\ 0, & \text{если иначе} \end{cases} \quad (8)$$

Таким образом, множество нейронов y_{ij}^m сети Хопфилда можно условно разбить на N подмножеств, и организовать в виде квадратных матриц (обозначим их y^m) размерностью $N \times N$. Каждая такая матрица будет соответствовать некоторому узлу-получателю, а элементы матрицы – каналам связи между соответствующими узлами ТКС. На основе соотношения (2.8) можно построить интерпретатор решений нейронной сети.

Теперь необходимо определить энергетическую функцию сети. Введём величину $K(Y)$ как функционал, определяющий стоимость дерева проложенных маршрутов. На допустимые решения наложим следующие ограничения:

- 1) Выходное значение нейронов должно быть либо 1, либо 0;
- 2) От узла-источника до каждого узла-получателя существует только один маршрут (т.е. в каждой матрице должен быть только один маршрут);
- 3) В каждый маршрут в решении могут входить только существующие ребра графа ТКС.

Заключение

В работе были представлены модели и методы много-адресной маршрутизации в телекоммуникационных системах нового поколения. Результаты могут быть полезны для организации адаптивного много-агентного обслуживания пользователей GRID-инфраструктур различного масштаба и назначения и для создания мировой сети Internet нового поколения.

Литература

1. *Олифер В.Г., Оливер М.А.* Компьютерные сети. Принципы, технологии, протоколы. — СПб.: Питер, 2001. — 672с.
2. *Harshall F.* Data Communication, Computer Networks and Open Systems. — Adisson — Wesley, 1996.
3. *Tanenbaum A.S.* Computer Networks. — Prentice Hall, 1996.
4. *Тимофеев А.В.* Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях. — Анаголия, Санкт-Петербург, 2012, 280 с.

РАЗРАБОТКА БИБЛИОТЕКИ КЛАССОВ И АЛГОРИТМОВ КОМПЬЮТЕРНОЙ СИСТЕМЫ ДЛЯ ЗАДАЧИ СИНТЕЗА ИНТЕЛЛЕКТУАЛЬНОГО ПОВЕДЕНИЯ АГЕНТА НА ОСНОВЕ РЕКУРСИВНОЙ КОГНИТИВНОЙ АРХИТЕКТУРЫ¹

З.В. Нагоев, В.А. Денисенко

*Россия, Нальчик, Институт информатики
и проблем регионального управления КБНЦ РАН*

Введение

В [Нагоев, 2011a, Нагоев, 2011b] предложен подход к решению фундаментальной проблемы искусственного интеллекта (ИИ) — формализации семантики разумного мышления, с использованием *когнитивного моделирования* на основе концепции *рекурсивной (фрактальной) когнитивной архитектуры* и гипотезы об *инварианте организационно-функциональной структуры процесса интеллектуального принятия решения на основе когнитивных функций*. Эти разработки, как и предложенный в [Нагоев и др., 2011] метод обучения мультиагентных нейроподобных систем на основе *онтонейроморфогенеза*, направлены на создание самоорганизующихся мультиагентных эмерджентных систем, способных к эмуляции функций психики, целеполаганию и адаптивному целенаправленному поведению на основе семантизации действительности и построения социальных связей.

Принципы когнитивной организации интеллектуальных систем тесно связаны с мультиагентным разделением функций в многоклеточном организме. Когнитивные центры, как в мозге человека, так и в искусственной интеллектуальной системе, на наш взгляд, должны представлять собой активные системы, взаимодействующие друг с другом на основе принципов коллективной оптимизации параметров, критичных, в первую очередь, для сохранения целостности всей системы, ее выживаемости.

Формальное описание агента должно учитывать, что поведение агента состоит из его действий, которые осуществляются в среде обитания агента и требуют затрат энергии. При этом необходимо помнить, что обмен энергией и информацией происходит не только между агентом и внешней средой, но и между внутренними агентами (органами, нейронами) агента верхнего уровня рекурсивной когнитивной архитектуры.

Формальное описание структуры рекурсивного агента

В [Нагоев и др., 2011] дано рекурсивное определение искусственно-го интеллекта (ИИ), использующее концепцию интеллектуального на-

¹ Работа выполнена при поддержке грантов РФФИ 12-01-00367-а, 13-01-00929-а, 12-07-00744-а

блюдателя, способного к перцепции и синтезу понятий. Рассматриваемые ниже концепция и определения также связаны с возможностями перцепции наблюдателя и его порогами чувствительности. С учетом того, что предложенное нами определение интеллектуальной системы рекурсивно и использует идею наблюдателя как интеллектуальной системы, принимающей решение об интеллектуальности других систем, определим формально те части наблюдателя, которые функционально взаимодействуют с системами, претендующими на интеллектуальность. Это позволит нам понять и формализовать именно те (наряду с другими) свойства таких систем, которые критичны для наблюдателя для классификации данной системы как интеллектуальной. В свою очередь, такая формализация этой системы, позволяющая понять ее сущность, даст нам возможность определить «вторую» (оставшуюся) часть наблюдателя — ведь он сам является интеллектуальной системой. Таким образом, мы должны выйти на полностью формальную систему, описывающую и наблюдателя, и интеллектуальную систему. Начнем с формального описания феноменологии перцепции наблюдателя I_i , погруженного в среду W , так как, на наш взгляд, это — именно та часть наблюдателя, которая непосредственно взаимодействует с внешними по отношению к нему системами, в том числе и с интеллектуальными.

Пусть $Z^i = \{z_0 = 0, z_1 = 1, \dots, z_k = z_{\max}^i \mid z_{j+1} = z_j + 1, z_j \in \mathbb{Z}\}$, $i = 1, \dots, 4$ — множества, элементы которых являются неотрицательными значениями отрезков целочисленных числовых прямых. Определим дискретное пространство-время наблюдателя как множество $Z^4 = \{Z^1 \times Z^2 \times Z^3 \times Z^4\}$. Обозначим через $z_i (z_{i1}, z_{i2}, z_{i3}, z_{i4})$ i -й элемент множества Z^4 . Будем называть его также — *точкой в пространстве-времени наблюдателя*, а значения z_{ij} — *координатами* этой точки.

Назовем *материей* $O = \{o_1^j, \dots, o_m^j\}$ — множество объектов (*объекты материи, объекты среды*) $o_i^j (q_{i1}^j, \dots, q_{in}^j) \in Q^n$, фиксируемых наблюдателем (*перцептивно значимых* для наблюдателя). Здесь $q_{ik}^j \in \mathbb{Z}$, $q_{ik}^j \geq 0$ — *целочисленные неотрицательные свойства* -го объекта o_i^j , находящегося в j -м состоянии, которые могут быть зафиксированы наблюдателем, а Q^n — *пространство свойств объектов*.

Сопоставим каждому значению свойства объекта целочисленное значение *энергии* $e(q_{ik}^j) = q_{ik}^j \geq 0$, которая понимается как субъективная (с точки зрения наблюдателя) мера способности объектов и явлений, внешних по отношению к данному объекту o_i^j , изменять конкретные свойства q_{ik}^j этого объекта. Мы будем называть эти характеристики также *энергией свойств*. Условно можно считать, что *полная энергия объекта* (или просто *энергия объекта*), рассматриваемого в нашей системе, в отличие от объекта реального мира, связана с работой, которую необходимо совершить для того, чтобы придать объекту те свойства, которыми он обладает. Отсюда, мы имеем возможность связать значения энергии с конкретными свойствами и счи-

тать, что *полная энергия* объекта o_i^j состоит из суммы значений энергии всех его свойств:

$$E(o_i^j) = \sum_{k=1}^N e(q_{ik}^j).$$

Среда $W = \{\mathbb{Z}^4 \times O\}$ (*мир*) воспринимается агентом как множество точек дискретного пространства-времени \mathbb{Z}^4 и множество объектов материи O , которые могут располагаться в любой из этих точек. Таким образом, эта среда, представляющая собой множество всевозможных состояний содержащихся в ней объектов, *с точки зрения наблюдателя дискретна (а, может быть, как раз с этой точки зрения она непрерывна)*.

Состояние j объекта o_i^j в точке (на шаге) дискретного времени t опишем как:

$$o_{it}^j(z_{i1}, z_{i2}, z_{i3}, t, q_{i1}^j, \dots, q_{in}^j) = o_{it}^j(z_i, o_i^j), \quad o_i^j \in O, \quad z_i \in \mathbb{Z}^4, \quad o_{it}^j \in W.$$

Будем считать, что:

$$\forall o_{it}^j \in O, \exists z_i \in \mathbb{Z}^4, \exists o_{it}^j(z_i, o_i^j) \in W, \nexists o_k^h \in O, k \neq i \wedge h \neq j, \nexists o_{it}^h(z_i, o_k^h) \in W.$$

Таким образом, каждому состоянию объекта материи всегда соответствует единственная точка в пространстве-времени и она не может соответствовать какому-либо другому объекту. Поэтому будем также называть среду W множеством *пространства-времени-материи (ПВМ)*, или *пространством состояний объектов*. Использование *дискретной модели ПВМ* в данном случае оправдано тем, что наша задача состоит в описании субъективного восприятия (перцепции) наблюдателя, которое дискретно в силу наличия *порогов чувствительности сенсоров и анализаторов наблюдателя*, с помощью которых он получает данные для восстановления перцептивной картины (модели) внешней среды.

Пусть $c(o_{it}^j)$ — количество ненулевых значений свойств $q_{ix}^j > 0$ объекта o_i^j в точке времени t . Назовем *элементарным явлением* изменение состояния объекта с течением дискретного времени, шаг которого равен порогу чувствительности наблюдателя по времени:

$$p_t(o_{is}^j) = p_t o_{is}^j = p_t = \begin{cases} o_{if}^y, c(o_{it}^j) = c(o_{if}^y) \\ o_{kf}^h, c(o_{it}^j) \neq c(o_{kf}^h) \end{cases},$$

где $p_t: W \rightarrow W$ — *оператор явления* (или просто *явление*), который начинает действовать в момент времени $t = s$ и изменяет некоторые координаты z_{ij} объекта o_i^j и некоторые его свойства q_{ix}^j , соответственно, меняя значения энергии

$e(q_{ik}^i)$ этих свойств и значение полной энергии объекта $E(o_i^i)$. Таким образом, в результате применения явления к объекту, он может увеличить, или уменьшить свою полную энергию за счет соответствующей модификации энергии свойств. В том случае, если полная энергия объекта увеличивается, или уменьшается, будем говорить, что объект, соответственно, *приобретает*, или *теряет энергию*.

Назовем *энергией явления* (или *энергией оператора явления*) (1) величину:

$$\Delta E(p_l) = E(o_i^y) - E(o_i^i).$$

Будем считать, что операторы $t_s(p_l) = s$ и $t_f(p_l) = f$ возвращают, соответственно, время начала и время окончания действия оператора.

Библиотека классов для реализации модели рекурсивного агента

Для реализации предложенной формализации была разработана библиотека классов на языке программирования C++, описывающая предложенные в работе конструкции. Состав библиотеки представлен на рисунке 1.

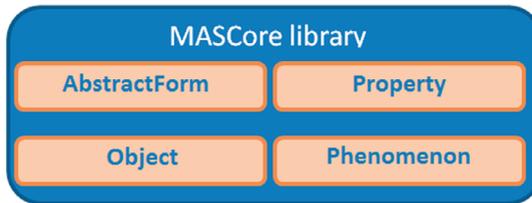


Рис. 1. Структура библиотеки MASCore

Иерархия классов в библиотеке представлена на рисунке 2.

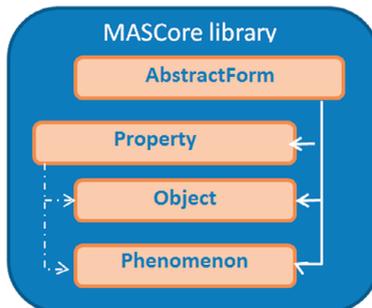


Рис. 2. Иерархия классов в библиотеке MASCore. Сплошная линия — наследование класса, пунктирная линия — использование класса

Разработанная система классов служит основой реализации моделей и методов рекурсивной мультиагентной интеллектуальной обработки информации.

Заключение

В работе описана лишь часть разрабатываемой мультиагентной, рекурсивной, когнитивной архитектуры, однако предложенная реализация уже позволяет исследовать простейшие элементы взаимодействия агентов. В дальнейшем, усилия коллектива разработчиков будут направлены на разработку формального описания системы и ее компьютерной реализации.

Литература

1. *Нагоев З.В.* Формализация агента для задачи синтеза интеллектуального поведения на основе рекурсивной когнитивной архитектуры. — Материалы международного конгресса по интеллектуальным системам и информационным технологиям IS&IT12, 2–9 сентября, Дивноморское, 2012.
2. *Nagoev Z.V.* Multiagent recursive cognitive architecture. — *Biologically Inspired Cognitive Architectures 2012, Proceedings of the third annual meeting of the BICA Society*, in *Advances in Intelligent Systems and Computing series*, Springer, 2012, pp. 247–248.
3. *Нагоев З. В, Нагоева О.В.*, Инвариант интеллектуальной фрактальной мультиагентной когнитивной архитектуры // Материалы II Международной конференции «Автоматизация управления и интеллектуальные системы и среды», Т. II, 2011, с. 195–199.
4. *Нагоев З. В, Ойхунов М.М.*, Метод дискретно-динамических частиц в задачах механики деформируемого твердого тела // *Известия РАН. Механика твердого тела*, №4, М.: Наука, 2011, с. 155–169
5. *Иванов П.М., Нагоев З.В.* Самоорганизующаяся система принятия решений на основе автоматного представления рекурсивной мультиагентной когнитивной архитектуры для систем обволакивающего интеллекта. // *Известия КБНЦ РАН*, Нальчик: Издательство КБНЦ РАН, 2012, № 5 (49), с. 30–37.

РАЗРАБОТКА ПРИНЦИПОВ ПОДДЕРЖКИ ЭКОНОМИЧЕСКИХ ИНТЕРЕСОВ ПОЛЬЗОВАТЕЛЕЙ В МУЛЬТИАГЕНТНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СРЕДЕ ОБИТАНИЯ¹

О.В. Нагоева

*Федеральное государственное бюджетное учреждение науки
Институт информатики и проблем регионального управления
Кабардино-Балкарского научного центра РАН*

Введение

Интеллектуальные среды обитания (ИСО) (ambient intelligence) — одно из самых перспективных направлений развития интегрированных информационных систем — в настоящее время содержит много методологических неясностей и неразрешенных теоретических проблем. Одной из важнейших проблем такого рода является проблема формализации разумного поведения такой системы как на уровне формирования политики, стратегии и тактики, так и на уровне управления многочисленными устройствами, входящими в ее состав. Такая формализация, в свою очередь, тесно связана с задачей формализации интеллектуального мышления и другими классическими задачами искусственного интеллекта. Определяющее значение для формирования методологической основы решения этих задач является проблема мотивации искусственной интеллектуальной системы. В данной работе предлагается подход к проблеме формализации мотивации системы ИСО, связанный с постановкой задачи оптимизации, где в качестве целевой функции рассматривается прогнозируемое общее сальдо биоэнергии, суммированное по всем агентам — клиентам системы ИСО. При этом биоэнергия рассматривается как мера активности агента.

В статье приводится обобщенная архитектура системы ИСО, основанная на применении гетерогенных мультиагентных систем. Дается определение экономического поведения обобщенного агента, подчеркивается его отличие от простого интеллектуального поведения, связанное с конкуренцией агентов за внешние ресурсы.

В статье формулируется постановка задачи оптимизации сальдо биоэнергии обобщенного интеллектуального агента, приводится структурно-функциональная схема агента, описываются некоторые алгоритмы обработки информации.

1. Экономические интересы и интеллектуальные системы

Основная идея систем ИСО [2,3] связана с насыщением окружающей среды (в территориально-пространственном и инфраструктурно-техниче-

¹ Работа выполнена при поддержке грантов РФФИ 12-01-00367-а, 13-01-00929-а

ском смысле) гетерогенными электронными сенсорными и эффекторными устройствами и объединением их в системы распределенного автоматического интеллектуального анализа ситуаций и синтеза решений для универсального круга задач на основе развитых аппаратно-программных сетевых решений и интегрированных систем искусственного интеллекта.

Центральным интегрирующим ядром такой системы, очевидно, является именно интеллектуальная система принятия решений, являющаяся логической надстройкой над сетецентрическим базисом и роботизированными исполнительными устройствами. Технологической основой ИСО являются сетевые системы и, по нашему мнению, — гетерогенные (программные, роботизированные, киборги) мультиагентные системы [4] (рисунок 1).

Концепция ИСО предполагает, что люди будут окружены интеллектуальными устройствами, призванными облегчить их деятельность. Окружающая среда будет сама подстраиваться под потребности и привычки поль-

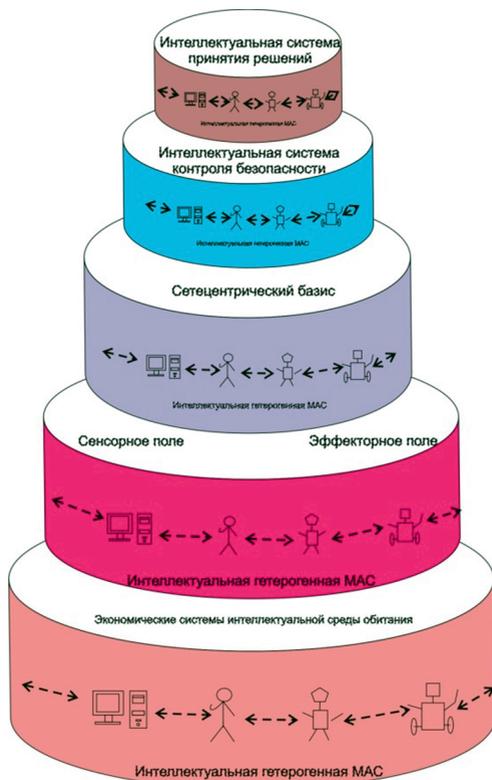


Рис. 1. Архитектура гетерогенной мультиагентной интеллектуальной среды обитания для поддержки экономических интересов пользователей

зователей, создавая своеобразные экосистемы ИСО. Группа зарубежных ученых, осуществляющая в рамках проекта «Serenity» исследования в области обеспечения безопасного и надежного функционирования экосистем ИСО, в качестве методологической основы предлагает использование шаблонов и моделей поведения агентов в различных ситуациях [3].

Одной из сфер жизни человека, в которой, как нам представляется, с внедрением технологии ИСО произойдут наибольшие изменения, является экономика в целом и практически полная автоматизация экономического поведения человека, в частности. С учетом очевидной коммерческой привлекательности распространения технологии ИСО на экономические системы, можно предположить, что именно эти системы станут локомотивом внедрения ИСО в повседневную жизнь.

В этой связи весьма интересным представляется вопрос о том, какими специфическими свойствами должна обладать система ИСО для того, чтобы осуществлять поддержку экономического поведения, предоставляя пользователям спектр услуг по требованию, либо даже превосходящая события его экономической жизни.

Ключевым здесь, на наш взгляд, является вопрос определения экономического поведения и разграничения его отличий от обычного поведения человека.

Однако, как показывает анализ литературы, провести такое разграничение оказывается довольно трудно.

Центральное место во всех трактатах по экономической теории с незначительными отступлениями занимает одна и та же мысль: все исследования в области экономики направлены на анализ и предсказание человеческого поведения [1]. Так, по мнению А. Маршала: «Экономическая наука занимается исследованием нормальной жизнедеятельности человеческого общества; она изучает ту сферу индивидуальных и общественных действий, которая теснейшим образом связана с созданием и использованием материальных основ благосостояния» [5].

Л. Роббинс дает следующее определение: «Экономическая теория — это наука, изучающая человеческое поведение с точки зрения соотношения между целями и ограниченными средствами, которые могут иметь различное употребление» [7].

Таким образом, есть все основания связать проблематику моделирования экономического поведения и моделирования поведения вообще. Последняя часть есть, как известно, лежит в сфере интересов искусственного интеллекта.

Таким образом, **актуальность** данного исследования определяется сложившимися к настоящему времени предпосылками ускоренного развития и широкого внедрения систем поддержки принятия экономических решений в рамках концепции ИСО с учетом базовой методологии искусственного интеллекта и свойств особого вида человеческого поведения — экономического поведения.

2. Потребление, производство и распределение в интеллектуальной среде обитания

Как известно, экономическая наука анализирует такие основные категории, как производство, распределение, обмен и потребление материальных благ и услуг, которые необходимы для жизнедеятельности людей. Эти категории в целом описывают экономическую сферу общества [6]. Сам анализ базовых категорий осуществляется на основе использования следующих аналитических категорий: ограниченность ресурсов, издержки, предпочтения и выбор. Эти понятия структурируются в рамках отдельных взаимосвязанных процессов оптимизации на уровне индивидуальных решений и равновесия на уровне всего общества [10].

Переходя на язык исследователя человеческого поведения, экономисты интересуются тем, каким образом люди используют свои ограниченные ресурсы для производства, распределения и обмена товаров и услуг в целях потребления, т. е. исследуют процесс выбора между альтернативными вариантами использования редких ресурсов, методы организации ресурсов, способы распределения богатства и вознаграждений за экономическую деятельность [9].

С точки зрения искусственного интеллекта экономическое поведение — весьма интересный объект исследования, так как является частью общего поведения и в то же время, базируется на некоторых специальных его формах. В данной работе мы придерживаемся подхода к моделированию поведения, предложенного в работе [7], согласно которому основным мотивом поведения (в том числе и экономического) является максимизация сальдо изменений биоэнергии, прогнозируемой интеллектуальной системой принятия решения агента на отрезке планирования. При этом биоэнергия определяется как мера активности агента в среде.

Объектом данного исследования является семантика согласованного разумного поведения и коллективного решения задач обобщенными агентами в составе гетерогенной мультиагентной системы интеллектуальной среды обитания.

Предмет исследования — возможность применения гипотезы биоэнергетической оптимальности мышления и поведения для формализации экономического поведения.

Целью данного исследования является разработка формализации процессов мышления, связанных с экономическим поведением.

Основной задачей данного исследования является постановка задачи формирования поведения ИСО в режиме поддержки принятия экономических решений.

Таким образом, в данной работе нас интересует, прежде всего, формальное описание процесса принятия решения в ИСО, когда целью системы является поддержка экономических интересов пользователей.

4. Гетерогенная мультиагентная система поддержки экономических интересов

На рисунке 2 приведен вариант функциональной архитектуры ИСО поддержки принятия экономических решений.

Предполагается, что ИСО предоставит каждому пользователю в режиме индивидуального обслуживания услуги гетерогенной мультиагентной системы. При этом каждый участник такой системы (робот, программный агент, киборг, или человек) будет защищать интересы пользователя. Таким образом, интеллектуальная система принятия решений (ИСПР) ИСО, структурно представляющая собой гетерогенную МАС, будет находится в режиме диалога с конгломератами (коалициями) других МАС разного уровня,

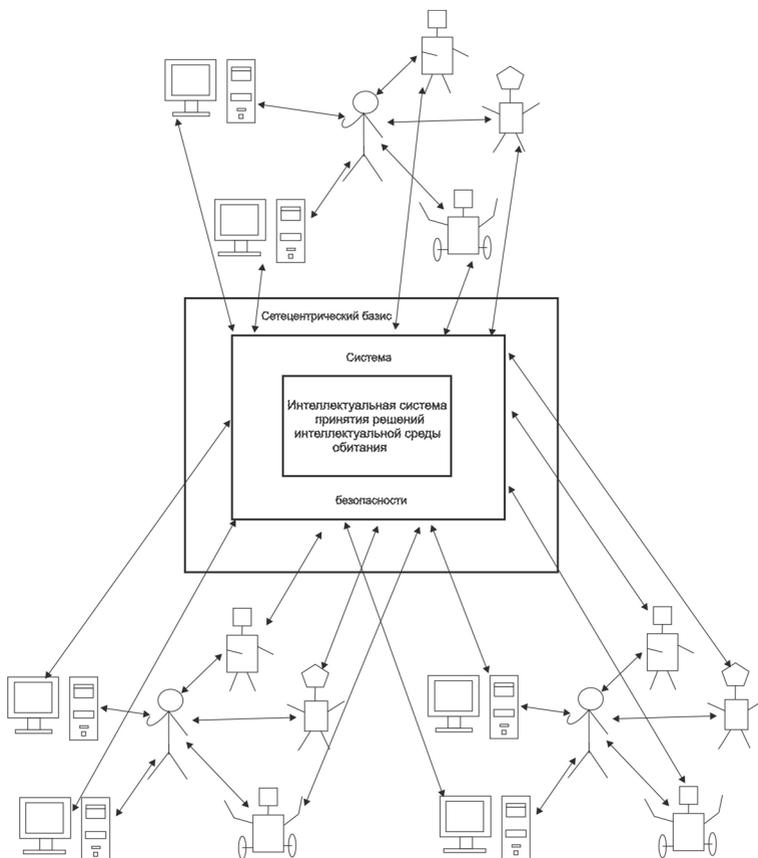


Рис. 2. Структура экономических запросов к ИСО

сгруппированных вокруг конкретных пользователей, или их объединений. В этом случае в составе коалиций, как правило, будут разделены роли и каждый из агентов в той или иной мере будет уполномочен вести непосредственные переговоры с ИСПР, отстаивая интересы лидера коалиции.

Архитектура ИСПР представлена на рисунке 3.

Построим модель обобщенного агента, которая затем будет использоваться для определения оптимизационной задачи верхнего уровня, которую должна решать ИСПР ИСО.

Пусть R_i — i -й рецептор системы, описывающийся функцией:

$R_{it} = f(w_{i\Delta t})$ от наблюдаемого на отрезке времени $\Delta t = t - t_s$, где t_s — момент времени начала наблюдения, а t — время завершения наблюдения, фрагмента внешней или внутренней среды $w_{i\Delta t}$,

$$R_{it} \in [R_i^{\min}; R_i^{\max}] \in R.$$

Для простоты считаем, что момент окончания наблюдения совпадает с моментом появления значения R_{it} на выходе рецептора.

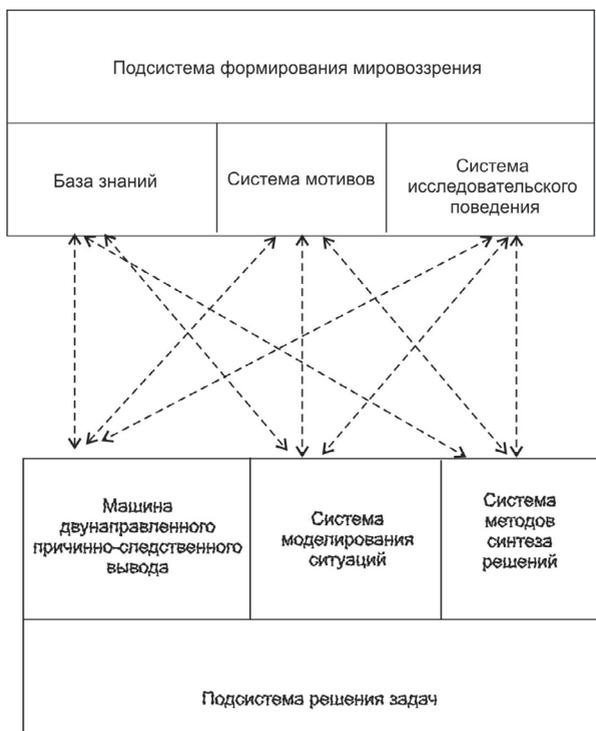


Рис. 3. Структура экономических запросов к ИСО

Назовем значение R_{it} *сигналом рецептора* R_i в момент времени t . Рецептор R_i генерирует сигналы R_{it} с частотой φ в равные промежутки времени τ .

Таким образом, сигналы рецептора образуют дискретную последовательность $R_{i1}, R_{i2}, \dots, R_{iN_i}, \dots$

Пусть в обобщенном теле обобщенного агента n рецепторов $R_{i1}, R_{i2}, \dots, R_{in}$. Для простоты будем считать, что все они выдают сигналы R_{it} с одинаковой частотой φ . Тогда в момент времени t общий выход всех рецепторов можно представить в виде сенсорного образа $R_t(R_{t1}, R_{t2}, \dots, R_{tN_t})$. Следовательно на некотором интервале времени $T = t_h - t_1$ система рецепторов сгенерирует $[T/\tau]$ сенсорных образов $\bar{R}_T(R_{t1}, R_{t2}, \dots, R_{tN_t})$.

Представим, что у обобщенного агента есть *анализаторы* A_i , состоящие из процессорных элементов p_{ij} , $A_i(p_{i1}, \dots, p_{ij})$, связанных друг с другом входами и выходами в иерархической топологии, так что входом p_{ij+1} -го процессорного элемента является выход p_{ij} -го процессорного элемента: $p_{ij} = g_{ij}(p_{ij-1})$, где $g_{ij}(p_{ij-1})$ — некоторая дискретная функция распознавания, а $p_{i1} = g_{i1}(R_T)$ — процессорный элемент, получающий на вход последовательности сенсорных образов \bar{R}_T .

Будем считать, что последовательности \bar{R}_T являются сообщениями, закодированными последовательным кодом, а функции g_{ij} — задают отношения декодирования и имеет вид:

$$p_{i1} = g_{i1}(\bar{R}_{iT}) = \begin{cases} 0, \bar{R}_{iT}, \text{ не распознано} \\ m_{i1} \in N, \bar{R}_{iT}, \text{ распознано} \end{cases}$$

Следует отметить, что выходы процессорного элемента p_{i1} генерируются с разными интервалами, так как интервал T , в течение которого анализируется сигнал сообщения \bar{R}_{iT} , может быть произвольным.

Последовательности m_{i1} , таким образом, приходят на процессорный элемент p_{i2} через неравные промежутки времени. Будем считать, что эти сигналы образуют последовательности вида: $\bar{m}_{i1} = (m_{i1}^{T_1}, \dots, m_{i1}^{T_q})$, где T_c — интервал времени, охватывающий все сигналы, входящие в сообщение \bar{R}_{iT_c} . Тогда выход процессорного элемента p_{ij} можно записать в виде:

$$p_{ij} = g_{ij}(\bar{m}_{ij-1}) = \begin{cases} 0, \bar{m}_{ij-1}, \text{ не распознано} \\ m_{i2} \in N, \bar{m}_{ij-1}, \text{ распознано} \end{cases}$$

Таким образом, преобразования сигналов, имеющие место в анализаторе A_i , будут описываться выражением:

$$S_{ic} = g_{jm}(g_{im-1}(\dots(g_{i1}(\bar{R}_{iT}))\dots)),$$

где S_{ic} — текущая *ситуация*, идентифицируемая анализатором A_i .

Для простоты будем считать, что анализаторы A_i генерируют ситуации S_{ic} синхронно и подают их на вход внутренней модели M .

Будем считать, что функция этой модели состоит в формировании прогноза изменения состояния биоэнергии Δb_{ct_j} в будущем в момент времени t_f , с учетом наличия текущей ситуации $S_c(S_{1c}, \dots, S_{dc})$, где S_{1i} — ситуации, идентифицируемые i -м анализатором A_i :

$$\Delta b_{ct_j} = M(S_c).$$

Таким образом, иерархическая система афферентных преобразователей информации сопоставляет наблюдаемым фрагментом среды, прогнозы состояния биоэнергии в будущем.

Будем считать, что ситуация S_c и прогноз биоэнергии b_{ct} подаются на вход модуля синтеза решений D , вычислительная структура которого соответствует требованиям, связанным с необходимостью решения задачи максимизации биоэнергии к моменту времени t_f .

При этом в качестве ограничений оптимизационной задачи используются знания агента о среде $Kn_i(W)$ и о себе самом $Kn_i(Ag)$.

При этом решение принимается, исходя из принципа максимизации сальдо биоэнергии $B_{ct_j}^i$ агента Ag_i к моменту времени t_f :

$$B_{ct_j} = \sum_{t=t_c}^{t_f} \Delta b_{ct},$$

где Δb_{ct} — прогноз изменения состояния биоэнергии агента в момент времени t .

Сущность решения оптимизационной задачи в данном случае состоит в подборе последовательности (плана) действий Ac_{ij} , выполнение которой в условиях наступления ситуации S_c , доставляет максимум сальдо биоэнергии B_{ct_j} , которое в этом случае можно записать как функцию от Ac_{ij} :

$$B_{ct_j} = f(Ac_{ij}).$$

Таким образом, постановку задачи поиска оптимального плана действия агента, можно записать в виде:

$$B_{ct_j} = f(Ac_{ij}) \xrightarrow{Kn_i(W), Kn_i(Ag_i)} \max. \quad (1)$$

В данной работе мы не рассматриваем структуру функции и формализмы модуля принятия решений, осуществляющего синтез решений.

Далее мы рассмотрим, как изменяется эта постановка в случае погружения обобщенного агента в интеллектуальную среду обитания.

Пусть в системе ИСО находятся одновременно агенты из множества: $Ag = \{Ag_1, Ag_2, \dots, Ag_i\}$ и пусть каждый из них стремится решить задачу вида (1).

Проблема состоит в том, чтобы определить, задачу какого вида должна решать ИСО для того, чтобы максимально полно и эффективно обеспечить экономические потребности наибольшего количества пользователей и их представителей — гетерогенных агентов.

Следует отметить, что даже в системах ИСО локального масштаба, очевидно, наличие конфликта интересов пользователей — ведь речь идет об экономической составляющей поведения, основной специфической чертой которого, как мы выяснили, является конкуренция и борьба за ресурсы.

Таким образом, выбор критериев оптимальности и синтез методов решения задач в системе поддержки экономических интересов ИСО будет далеко не тривиальным.

В качестве наиболее простого варианта можно указать на критерий максимизации общего сальдо биоэнергии всех агентов системы:

$$F(f(Ac_{i1}), f(Ac_{i2}), \dots, f(Ac_{in})) \xrightarrow{Kn(W), Kn(Ag)} \max,$$

где F — целевая функция системе поддержки экономических интересов ИСО, а $Kn(W)$ и $Kn(Ag)$ — ее знания о реальном мире и об агентах.

Заключение

В ходе выполнения работы получены следующие результаты:

- разработана обобщенная архитектура системы ИСО, основанная на применении гетерогенных мультиагентных систем;
- разработана модель обобщенного агента в составе гетерогенной системы;
- дана постановка оптимизационной задачи формирования экономического поведения обобщенного агента;
- дана постановка задачи оптимизации системой ИСО сальдо биоэнергии всех агентов, находящихся в сфере ее действия;
- разработана структурно-функциональная схема ИСПР агента в составе ИСО.

Литература

1. *Hirschleifer J.* The Expanded Domain of Economics. — American Economic Review, December. 1985, V.75.
2. *Snidaro L., Foresti G.L.* «Knowledge representation for ambient security», Department of Mathematics and Computer Science, University of Udine, Italy, 2007.

3. *Spanoudakis G., Gómez A.M., Kokolakis S.*, (Eds.) «Security and Dependability for Ambient Intelligence», *Advances in Information Security*, Vol. 45, 2009, IX, 392 p. 80 illus., 30 in color., Hardcover.
 4. *Башоров З.А., Макаревич О.Б., Нагоев З.В.*, Мультиагентная система обволакивающей безопасности на основе автономных программных агентов и мобильных роботов. — Материалы XI Международной научно-практической конференции «Информационная безопасность 2010». Ч. 1. — Таганрог: Изд-во ТТИ ЮФУ, 2010. — 256 с., с.153–157.
 5. *Маршалл А.* Принципы политической экономии: Пер. с англ. М.: Прогресс, 1983. Т.1.
 6. *Заславская Т.Н., Рыбкина Р.В.* Социология экономической жизни. Новосибирск, 1991. С.51.
 7. *Нагоев З.В.* Моделирование целенаправленного поведения в мультиагентных системах на основе принципа биоэнергетической оптимальности. — Материалы международной научно-технической конференции «Инновационные технологии XXI века в управлении, информатике и образовании», КБГУ, 2008.
 8. *Роббинс Л.* Предмет экономической науки // 1 THESIS, 1993. Зима. Т.1. Вып.1.
 9. *Смельсер Н. Дж.* Социология экономической жизни. Перевод ГПНТБ № 11304. Новосибирск, 1985. С.51.
 10. *Хайлбронер Р.* Экономическая теория как универсальная наука // THESIS, 1993. Зима. Т.1. Вып.1. С.41.
-

МЕТОДИКА СИНТЕЗА РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ДИНАМИЧЕСКИХ ОБРАЗОВ

Никитин К.В.

*ассистент кафедры компьютерных систем
и программных технологий СПбГПУ, exciter@mail.ru*

Аннотация: В работе приводится описание разработанной методики синтеза импульсных рекуррентных нейронных сетей в составе машины неустойчивых состояний для решения задачи распознавания динамических образов в рамках парадигмы резервуарных вычислений.

Введение

Среди множества различных моделей нейронных сетей (НС) наибольшими вычислительными возможностями обладают НС с обратными связями или рекуррентные нейронные сети (РНС) [1]. РНС можно рассматривать как сложные нелинейные динамические системы, функционирующие как в устойчивых, колебательных и неустойчивых (хаотических) режимах. Однако платой за все преимущества РНС является значительное усложнение алгоритма их настройки или обучения.

Сложность и неэффективность существующих алгоритмов обучения РНС вынудила исследователей искать новые подходы, стратегии для использования их вычислительных возможностей. Одним из таких подходов стало новое направление — резервуарные вычисления (РВ).

Направление по РВ на данный момент развивается довольно динамично, но проблема синтеза резервуара под конкретную задачу является актуальной и нерешенной [4]. В работе была разработана методика синтеза резервуара — импульсной РНС в составе машины неустойчивых состояний (МНС) [3] — одной из основных моделей РВ, максимально приближенной к биологическому прототипу.

Модель машины неустойчивых состояний

Основная идея РВ заключается в использовании РНС как резервуара с богатой динамикой и мощными вычислительными возможностями. При этом резервуар формируется случайным образом, что исключает необходимость проводить его обучение [4]. При подаче на вход резервуара непрерывного сигнала он попадает в определенный динамический режим или состояние. Это состояние зависит от входного сигнала. Выход резервуара соединен со специальными устройствами — считывателями, которые выделяют из состояния резервуара полезную информацию и таким образом

решают поставленную. Схема РВ в общем и МНС в частности, показана на рисунке 1.

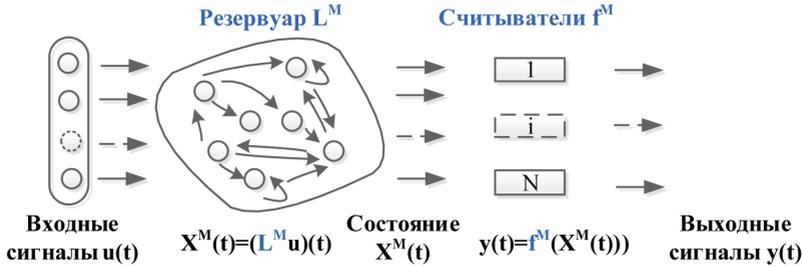


Рис. 1. Схема машины неустойчивых состояний

Резервуар в МНС является импульсной РНС [2] со случайной 3-х-мерной структурой, в которой импульсные нейроны соединены с помощью химических синапсов. Вероятность связи между двумя нейронами определяется по формуле [3]:

$$P(a, b) = \min [1, C(a, b) \cdot \exp(-D(a, b)/\lambda^2)],$$

где буквами a и b обозначаются два нейрона, $D(a, b)$ — евклидово расстояние между нейронами, λ — средняя длина связи между нейронами, $C(a, b)$ характеризует плотность связей.

Обобщенная методика синтеза импульсных РНС

Анализ результатов экспериментального исследования позволил построить обобщенную методику синтеза импульсных РНС и встроить ее в схему решения задачи распознавания динамических образов. Соответствующая схема представлена на рисунке 2.

Далее рассмотрим подробно каждый шаг методики.

Входные данные и их преобразование

Входные данные представлены набором примеров динамических образов в виде пар «входной сигнал—соответствующий ему выходной сигнал». Входной и выходной сигнал в общем случае являются многомерными функциями времени, размерность входного сигнала n_{in} , выходного — n_{out} . Возможны два способа задания — непрерывная и импульсная. В непрерывном случае каждая входная и выходная функция задаются аналитически (точно) либо численно (приближенно).



Рис. 2. Методика синтеза импульсных РНС при решении задачи распознавания динамических образов с помощью МНС

При численном задании входная функция времени разбивается на набор интервалов и на каждом интервале задается значение этой функции. Количество интервалов определяет точность задания функции. Важными параметрами является количество и качество примеров.

После того, как определен набор примеров, необходимо выбрать **способ их подачи** на вход резервуара — импульсной РНС. Предпочтительной является импульсная форма представления входных сигналов. Поэтому если входной сигнал задан непрерывно, следует выполнить **преобразование** этого сигнала в импульсную форму. Среди всех алгоритмов преобразования из непрерывной в импульсную форму следует отдать предпочтение алгоритмам на основе пространственно-временного кодирования (п. 2.4.4), в частности, алгоритму на основе рецептивных полей. Этот алгоритм автоматически выполняет нормировку входных данных, поэтому никакая дополнительная предобработка не требуется. Ключевыми параметрами этого алгоритма являются k_x — во сколько импульсных входов преобразуется каждый непрерывный вход и T_{\min} — минимальный период импульсов.

При выборе их значений следует руководствоваться соображениями, что чем сложнее форма непрерывного сигнала, тем больше значение k_x следует задавать). Следует начинать со значения k_x , равного 8 и далее увеличивая до 16, 32 и т. д. Минимальный период T_{\min} следует задавать не меньше чем 1 мс и не больше, чем постоянная времени нейрона (20–30 мс).

Если входной сигнал подается напрямую в непрерывной форме, то необходимо произвести его нормирование, так чтобы значения, которые он принимает, лежали в диапазоне от 0 до 1.

Помимо проблемы представления входных данных в импульсной форме существует проблема, связанная с временем представления входных образов. Это время не должно превышать глубину памяти резервуара — в противном случае часть информации о входном сигнале будет «забыта». Для решения этой проблемы может быть использована техника нормализации сигналов во временном диапазоне. Средняя глубина памяти импульсной РНС равна 0.1–0.5 с. Если время представления входного образа превышает эти значения, следует сжать входной образ во времени до t_{norm} (например, до 0.25 с).

Синтез резервуара

Вначале определяется **количество входов** n_{vh} РНС. Если преобразование из непрерывной в импульсную форму не происходит, то оно равно количеству входных сигналов n_{in} , если преобразование выполняется, то количество входов равно произведению количества входных сигналов на коэффициент размножения по входам k_x .

Вероятность связи между входами и нейронами резервуара p_{vh} следует задать равной 10–20 %.

Количество нейронов n_{nn} резервуара определяется сложностью задачи распознавания и количеством входов. Чем выше сложность и чем больше количество входов, тем больше должно быть нейронов. Минимально-допустимое значение числа нейронов равно 125 и соответствует трехмерной решетке $5 \times 5 \times 5$.

Процент подавляющих нейронов p_{inh} задается равным 20–30 % для любых типов нейронов.

Средняя длина связей λ вначале задается равной 2. Затем можно будет скорректировать это значение, немного увеличив его (предельное значение равно 3).

Плотность связей задается матрицей C размерности 2×2 ($C[i, j]$ — плотность связей между нейронами типа i и нейронами типа j , где тип 1 — подавляющий, тип 2 — возбуждающий). Возможные значения матрицы C :

$$C = \begin{bmatrix} 0.4 & 0.4 \\ 0.2 & 0.4 \end{bmatrix}, \quad C = \begin{bmatrix} 0.1 & 0.4 \\ 0.2 & 0.3 \end{bmatrix}.$$

На основании значений n_{nn} , n_{vh} , p_{vh} , p_{inh} , C , λ и формулы (2.36) **случайным образом генерируется структура** РНС, содержащая информацию о наличии/отсутствии связей между входами и нейронами и между нейронами. Формально структуру можно описать с помощью матриц H_{vh} , H_{in} : $H_{vh}[i, j] = 1$, если связь между i входом и j нейроном есть; $H_{vh}[i, j] = 0$, если связь отсутствует; $H_{in}[i, j] = 1$, если связь между i и j нейронами есть; $H_{in}[i, j] = 0$, если связь между i и j нейронами отсутствует.

Тип нейронов. В качестве нейронов могут использоваться как простые нейроны типа «интегрирование и возбуждение», так и более сложные нейроны — Ижикевича, Ходжкина-Хаксли, Морриса-Лекара. Выбор той или иной модели зависит от сложности задачи. Для простых задач можно обойтись более простыми моделями «интегрирование и возбуждение», для более сложных задач — модели нейронов с более сложной динамикой (Ижикевича и др.). Выбор более сложной модели нейрона позволяет увеличить память резервуара за счет более длительной и уникальной реакции в ответ на входной изменяющийся сигнал.

При использовании **нейронов** типа «интегрирование и возбуждение» следует инициализировать их базовые параметры следующими значениями: $v_{rest} = v_{reset} = 0$, $\theta = 15$ мВ, $T_{refrac} = 3$ мс, $\tau = 30$ мс, утечка есть.

При использовании **нейронов Ижикевича** в качестве возбуждающих нейронов следует использовать модель RS с параметрами: $a = 0.02$, $b = 0.2$, $c = -65$, $d = 8$, а в качестве подавляющих нейронов модель FS с параметрами: $a = 0.1$, $b = 0.2$, $c = -65$, $d = 2$.

Тип связей — синапсов так же, как и тип нейронов определяется сложностью задачи, а также типом связи. Для связей, передающих входные непрерывные сигналы, используются электрические синапсы. Для связей, передающих импульсные входные сигналы, а также для всех связей между импульсными нейронами используются химические синапсы. Для увеличения вычислительных возможностей (нелинейность, глубина памяти) используются динамические синапсы с кратковременной памятью. Для адаптации РНС к входному импульсному сигналу используются синапсы с пластичностью, обладающие долговременной памятью.

Сила связей для всех синапсов вначале может быть задана детерминированной.

Сила связей между входами и нейронами для электрических синапсов задается вектором $W_{vh(el)}$ из двух значений (1-е — для подавляющих нейронов, 2-е — для возбуждающих нейронов). Для нейронов «интегрирование и возбуждение» возможные значения $W_{vh(el)}$ следующие:

$$W_{ch(el)} = [0.01 \ 0.01], \quad W_{vh(el)} = [0.03 \ 0.03],$$

а для нейронов Ижикевича:

$$W_{ch(el)} = [0.005 \ 0.005], \quad W_{vh(el)} = [0.03 \ 0.03].$$

При этом значения непрерывного входного сигнала должны быть нормализованы и приведены к диапазону от 0 до 1.

Сила связей между входами и нейронами для химических синапсов задается вектором $W_{vh(chem)}$ из двух значений (1-е — для подавляющих нейронов, 2-е — для возбуждающих нейронов). Для нейронов «интегрирование

и возбуждение» (а) и нейронов Ижикевича (б) возможные значения $W_{vh(chem)}$ следующие:

$$\text{а) } W_{vh(chem)} = [0.03 \ 0.03], \quad \text{б) } W_{vh(chem)} = [0.01 \ 0.01].$$

Сила связей между нейронами (для химических синапсов) задается матрицей W_{int} размерности 2×2 ($W_{int}[i,j]$ — сила связи между нейроном типа i и нейроном типа j , где тип 1 — подавляющий, тип 2 — возбуждающий). Для нейронов «интегрирование и возбуждение» (а) и нейронов Ижикевича (б) матрица W_{int} задается следующим образом:

$$\text{а) } W_{int} = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.05 \end{bmatrix}, \quad \text{б) } W_{int} = \begin{bmatrix} 0.001 & 0.01 \\ 0.01 & 0.015 \end{bmatrix}.$$

В дальнейшем силы связей можно сделать **случайными** — распределенными нормально с МО, равными приведенным выше значениям и СКО, равными 10–30% от значений математических ожиданий.

Постоянная времени затухания τ_{syn} импульсного тока задается для химических синапсов от входов и возбуждающих нейронов равной 3 мс и для подавляющих нейронов — 6 мс.

Задержки τ_{delay} для химических синапсов задаются равными 1 мс.

Выбор считывающих модулей

Последовательность выбора считывателей можно представить следующим образом:

- определение времени считывания — лучшие результаты получаются при времени считывании, соответствующем подаче $\frac{3}{4}$ входного образа;
- выбор алгоритма преобразования из импульсной формы в непрерывную — рекомендуемым является алгоритм, основанный на замене каждого импульса экспоненциально затухающим сигналом с постоянной времени $\tau_{window} = 5-20$ мс;
- выбор алгоритма сокращения размерности числа признаков — могут быть использованы как «фильтр» (PCA, LDA, Relieff, ISOMAP и др.), так и «вращатель» методы; количество компонент — от 5 до 20;
- выбор алгоритма решения конечной задачи: классификация — НСПР, SVM, PNN, LVQ, k-NN, аппроксимация — НСПР, РБФ-НС, кластеризация — k-means, GMM.

Оценка качества распознавания

После того, как пройдены все предыдущие этапы, можно приступить непосредственно к решению поставленной задачи распознавания динамич-

ческих образов и оценивать качество распознавания. Для этого лучше всего подходит процедура кросс-валидации.

Для расчета ошибки классификатора лучше всего вначале посчитать матрицу неточностей и по ней уже определить вероятности ошибок 1-го, 2-го родов для каждого из m классов. В качестве ошибки аппроксимации можно использовать усредненную (максимальную) абсолютную или относительную ошибки по всему множеству тестовых примеров. При решении задачи кластеризации качество анализируется с помощью специальных показателей — индекса Дэвиса-Болдуина, индекса Дана и др.

В случае, если полученное качество распознавания не устраивает, можно вернуться на один из предыдущих этапов. В итоге выполнения всех описанных выше действий (шагов) и рекомендаций задача распознавания динамических образов будет решена с требуемым качеством.

Заключение

Разработанная методика синтеза импульсных РНС в составе МНС отличается от известных подходов системностью, универсальностью, большей детализацией, что позволяет проводить целенаправленный синтез резервуара с требуемыми показателями, снизить трудоемкость разработки систем распознавания динамических образов и использовать новую модель МНС в системах резервуарных вычислений для задач распознавания динамических образов.

Применение методики для решения с помощью МНС задачи обработки реальных данных — электрокардиограмм с различными формами аритмии позволило повысить качество распознавания на 5–7% по сравнению с известными классическими подходами. Таким образом, предложенная методика синтеза импульсных РНС может дополнить существующие подходы в системах обработки информации и распознавания динамических образов для повышения качества и скорости их работы.

Литература

1. *Хайкин, С.* Нейронные сети: полный курс: [пер. с англ.] / С. Хайкин, 2-е изд. — М.: Вильямс, 2000. — 1104 с.
2. *Gerstner, W.* Spiking Neuron Models. Single Neurons, Populations, Plasticity / W. Gerstner, W. M Kistler. — Cambridge: Cambridge University Press, 2002. — 480 p.
3. *Maass, W.* Real-time computing without stable states: a new framework for neural computations based on perturbations / W. Maass, T. Natschläger, H. Markram // Neural Computation. — 2002. Vol. 11. — P. 2531–2560.
4. *Schrauwen, B.* An overview of reservoir computing theory, applications and implementations / B. Schrauwen, D. Verstraeten, J.V. Campenhout // Proc. of the 15th European Symposium on ANN. — 2007. — P. 471–482.

МУЛЬТИ-АГЕНТНОЕ УПРАВЛЕНИЕ И ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ИНФОРМАЦИОННЫХ ПОТОКОВ

А.В. Тимофеев

*зав. лаб. ИТУР СПИИРАН, профессор кафедры информатики
Санкт-Петербургского государственного университета, tav@iiias.spb.su*

Аннотация: Обсуждаются проблемы и методы мульти-агентного управления и интеллектуального анализа потоков данных. Значительное внимание уделяется описанию этих проблем и методов, изложенных в монографии автора «Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях». Рассматриваются также некоторые новые научные результаты, полученные в этой области (обнаружение и распознавание сетевых атак, интеллектуальный анализ потоков видеоданных, мульти-агентные технологии в GRID-сетях и т. п.).

Введение

Глобальные телекоммуникационные системы (ТКС) и распределённые информационные компьютерные сети (ИКС) обеспечивают колоссальные возможности по массовому доступу людей к корпоративным, региональным и мировым информационным и вычислительным ресурсам. Однако дальнейшее развитие глобальных ТКС и ИКС (создание ТКС новых поколений, GRID-сети и т. п.) в первую очередь связано с совершенствованием, автоматизацией, оптимизацией и интеллектуализацией систем сетевого управления и анализу потоков данных.

Сегодня сетевое управление глобальными ТКС как мощным средством массового доступа пользователей к распределённым ИКС в значительной степени зависит от «человеческого фактора», а именно от знаний, опыта, интуиции и воли сетевых администраторов и операторов. В то же время хорошо известно, что способности людей (в том числе профессионалов в области сетевого управления ТКС) принципиально ограничены пределами их психофизиологических возможностей. Даже если будет разработан наилучший (в том или ином смысле) человеко-машинный интеллектуальный интерфейс, сетевые администраторы и операторы не способны эффективно управлять многомерными сложными параллельными процессами передачи и обработки мультимедийных потоков данных. Поэтому они не всегда могут обеспечить высокое качество сетевого управления, надёжность и отказоустойчивость глобальных ТКС и распределённых ИКС. Однако на современном этапе развития телекоммуникационных и информационных технологий особенно важно гарантировать пользователям глобальных ТКС и распределённых ИКС высокое качество запрашиваемых услуг и информационно-вычислительных ресурсов.

Эффективным путём совершенствования сетевого управления и интеллектуального анализа потоков данных в глобальных ТКС и распределённых ИКС является его автоматизация на базе динамических моделей ТКС и ИКС как сложных объектов управления с переменной структурой. Для этого необходимо развивать методы оптимизации процессов маршрутизации потоков данных и совершенствовать принципы адаптивного и интеллектуального управления с использованием нейросетевых, мульти-агентных и GRID-технологий [1,2]. На этом новом пути возможен как учёт реальной динамики ТКС и ИКС, т. е. фактического изменения структуры (топологии узлов и каналов связей) и параметров (весов каналов связи) ТКС и ИКС, так и адаптация к различным факторам неопределённости и нестационарности.

Неопределённость и нестационарность реальных условий эксплуатации глобальных ТКС и ИКС заключается в неопределённости количества пользователей и характера (профиля) их запросов, в непредсказуемых изменениях структуры (сетевой топологии) и параметров (пропускной способности узлов и каналов связи), а также в возможных перегрузках трафика, сетевых конфликтах, сбоях и отказах. В этих условиях неопределённости и нестационарности системы сетевого управления ТКС и ИКС неизбежно должны быть адаптивными и интеллектуальными [1,2].

Поскольку глобальные ТКС и распределённые ИКС обслуживают интересы (запросы) большого количества пользователей, их системы сетевого управления и распределённой обработки потоков данных не могут быть локальными, а должны иметь глобальный и мульти-агентный (групповой) характер.

Глобальное сетевое управление может строиться как на традиционных принципах централизованного или децентрализованного управления, так и на новых принципах мульти-агентного управления, обработки и передачи информации. При этом важную роль играет интеллектуальный сетевой анализ и распознавание потоков данных [1].

Краткое описание монографии «Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях»

В последней монографии автора [1] рассматриваются динамические модели глобальных ТКС и распределённых ИКС с переменной структурой, методы динамической, адаптивной, нейросетевой и мульти-агентной маршрутизации потоков данных в сложных ТКС и ИКС с изменяющейся динамикой, принципы адаптивного, интеллектуального и мульти-агентного сетевого управления передачей и обработкой информационных потоков. Определённый интерес представляют также методы мультифрактального проектирования глобальных ТКС и распределённых ИКС нового поколения и программные средства их проектирования и имитационного моделирования.

Эти модели, методы и программные средства являются важной составной частью современной теории адаптивного, интеллектуального и мульти-агентного управления информационными потоками в глобальных ТКС. При этом в роли внешних агентов выступают пользователи (клиенты) с высокими требованиями к качеству обслуживания, кооперации по интересам и надёжности глобальных ТКС и распределённых ИКС.

Монография [1] включает в себя следующие разделы:

1. СОСТОЯНИЕ, КОНВЕРГЕНЦИЯ И ТЕНДЕНЦИИ РАЗВИТИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ
2. АРХИТЕКТУРА КОМПЬЮТЕРНЫХ СЕТЕЙ И ПРИНЦИПЫ ОБРАБОТКИ ИНФОРМАЦИОННЫХ ПОТОКОВ
3. МАТЕМАТИЧЕСКИЕ МОДЕЛИ И СИСТЕМНЫЙ АНАЛИЗ КОМПЬЮТЕРНЫХ СЕТЕЙ
4. ПРОБЛЕМЫ И ПРИНЦИПЫ СЕТЕВОГО УПРАВЛЕНИЯ ПОТОКАМИ ДАННЫХ
5. КРИТЕРИИ КОММУНИКАбельности и МЕТОДЫ СТАТИЧЕСКОЙ МАРШРУТИЗАЦИИ ИНФОРМАЦИОННЫХ ПОТОКОВ
6. МЕТОДЫ СТАТИЧЕСКОЙ и МНОГО-АДРЕСНОЙ МАРШРУТИЗАЦИИ ИНФОРМАЦИОННЫХ ПОТОКОВ
7. МЕТОДЫ АДАПТИВНОЙ, НЕЙРОСЕТЕВОЙ, МУЛЬТИ-АГЕНТНОЙ и МНОГОПОТОВОЙ МАРШРУТИЗАЦИИ ИНФОРМАЦИОННЫХ ПОТОКОВ
8. ИЕРАРХИЧЕСКАЯ ДЕКОМПОЗИЦИЯ и МУЛЬТИФРАКТАЛЬНОЕ МОДЕЛИРОВАНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ
9. БИБЛИОТЕКА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ и КОМПЛЕКС ПРОГРАММ МАРШРУТИЗАЦИИ
10. ПРИНЦИПЫ НЕЙРОСЕТЕВОГО и МУЛЬТИ-АГЕНТНОГО УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ
11. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ и РАСПОЗНАВАНИЕ ПОТОКОВ ДАННЫХ в МУЛЬТИ-АГЕНТНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ
12. МНОГОКРИТЕРИАЛЬНАЯ ОЦЕНКА ЭФФЕКТИВНОСТИ и ОПТИМИЗАЦИЯ ТОПОЛОГИЧЕСКИХ СТРУКТУР для ПРОЕКТИРОВАНИЯ GRID-СЕТЕЙ
13. СОВЕРШЕНСТВОВАНИЕ ГЛОБАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ на БАЗЕ НЕЙРОСЕТЕВЫХ, МУЛЬТИ-АГЕНТНЫХ и GRID-ТЕХНОЛОГИЙ.

Задачи и методы мультиагентного управления и интеллектуального анализа потоков данных

Некоторые новые задачи сетевого управления и обработки потоков данных и принципы их решения описаны в работах [2–19]. Методы ди-

намической, адаптивной и нейросетевой маршрутизации потоков данных предложены в [2,7–10]. Вопросы интеллектуального и нейросетевого анализа информационных потоков в глобальных ТКС рассматриваются в [4–6, 15,18,19].

Однако проблемы сетевого управления и параллельной обработки потоков данных остаются частично нерешёнными и требуют разработки новых подходов, моделей и методов. Среди них важное значение имеют модели, методы и новые информационные технологии для решения следующих задач:

- адаптивные, нейросетевые и мульти-агентные технологии маршрутизации потоков данных [1–7];
- интеллектуализация мультимодального человеко-машинного интерфейса [8–13];
- обработка потоков данных с помощью квантовых, нейросетевых и генных вычислений [4,5,6,15];
- мультифрактальное проектирование и многокритериальная оценка сетевых архитектур [16,17];
- когнитивный анализ и распознавание сложных видеоданных [5,6,18];
- обнаружение и классификация сетевых атак [19].

Результаты решения прикладных задач информатики

Полученные теоретические результаты успешно применялись для имитационного моделирования и решения следующих прикладных задач:

- моделирование генетического кода и квантовых вычислений [4,15];
- распознавание сложных изображений и сцен [18];
- классификация web-сайтов и сетевых атак [1,13,19];
- интеллектуальный анализ потоков видеоданных для оценки потенциальной террористической опасности на вокзалах и стратегических охраняемых объектах.

Заключение

Проблемы адаптивного сетевого управления и интеллектуального анализа потоков данных в глобальных ТКС и распределённых ИКС решены лишь частично и требуют новых подходов для их исследования. Среди этих подходов важную роль играют мульти-агентные, нейросетевые и GRID-технологии и когнитивный анализ информационных потоков в условиях неопределённости и нестационарности.

Работа выполнена при частичной поддержке грантов РФФИ №12-08-01167-а, №12-08-07022-д и Программы №14 (GRID) Президиума РАН.

Литература

1. Тимофеев А.В. Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях. — СПб.: Анатолия, 2012, 280 с.
2. Тимофеев А.В., Сырцев А.В. Модели и методы маршрутизации потоков данных в телекоммуникационных системах с изменяющейся динамикой. — М.: Новые технологии, 2005, 82 с.
3. Тимофеев А.В., Сырцев В.В., Йотсов В.С., Лютикова Л.В. Развитие и применение многозначных логик и сетевых потоков в интеллектуальных системах. — Труды СПИИРАН, вып. 2, 2004. С. 72–84.
4. Тимофеев А.В. Оптимизационный синтез и минимизация сложности генно-нейронных сетей по целочисленным базам данных. — Нейрокомпьютеры: разработка и применение, 2002, № 5–6. С. 34–39.
5. Timofeev A.V. Polynomial Neural Network with Self-Organizing Architecture. — International Journal on Optical Memory and Neural Networks, 2004, N 2.
6. Timofeev A.V. Parallelism and Self-Organization in Polynomial Neural Networks for Image Recognition. — Pattern Recognition and Image Analysis, 2005, vol. 15, No.1, pp. 97–100.
7. Тимофеев А.В., Сырцев А.В. Мультиагентная и нейросетевая маршрутизация потоков данных в телекоммуникационных сетях // Труды 10-й международной конференции «Knowledge–Dialogue–Solution» (16–26 июня. 2003, Варна) 2003. С. 187–190.
8. Тимофеев А.В. Модели мульти-агентного диалога и информационного управления в глобальных телекоммуникационных сетях // Труды 10-й международной конференции «Knowledge–Dialogue–Solution» (16–26 июня, 2003, Варна). 2003. С. 180–186.
9. Timofeev Adil. Adaptive Routing and Multi-Agent Control for Information Flows in IP-Networks. — Proceedings of Xith International Conference Knowledge-Dialogue-Solution (KDS-2005), June 20–30, Varna, Bulgaria, 2005. Pp. 442–445.
10. Syrtzev A.V., Timofeev A.V. Neural and Multi-Agent Routing in Telecommunicational Networks. — International Journal «Information Theories and Their Applications», 2003, № 2.
11. Timofeev A.V. Models for Multi-Agent Dialogue and Informational Control in Global Telecommunicational Networks. — International Journal «Information Theories and Their Applications», 2003, № 1.
12. Timofeev A.V. Multi-Agent Information Processing and Adaptive Control in Global Telecommunication and Computer Networks. — International Journal «Information Theories and Their Applications», 2003, № 10. Pp. 54–60.
13. Timofeev A.V. Intellectualization for Man-Machine Interface and Network Control in Multi-Agent Infotelecommunication Systems of New Generation. — Proceedings of 9th International Conference «Speech and Computer» (20–22 September, 2004), Saint-Petersburg, Russia. Pp. 694–700.
14. Timofeev A.V. Adaptive Control and Multi-Agent Interface for Infotelecommunication Systems of New Generation. — International Journal «Information Theories & Applications». Vol.11, 2004.
15. Амбарян Т., Тимофеев А. Модели квантовых и нейронных вычислений в задачах обработки информации. // Известия вузов. Приборостроение 2005, № 7. С. 35–40.
16. Тимофеев А.В., Димитриченко Д.П. Многокритериальная оценка сетевых топологических структур для моделирования и проектирования GRID-систем — Труды СПИИРАН. Вып. № 10. СПб.: Наука, 2008. С. 72–77.

17. Тимофеев А. В. Фрактальное моделирование и многокритериальная оптимизация компьютерных сетей. — International Book Series Information Science & Computing. Intelligent Engineering, vol 3/2009, № 11. Pp.79–83.
 18. Косовская Т.М., Тимофеев А.В. Логико-когнитивные методы распознавания и анализа сложных изображений и сцен. — International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE, VOLUME 6/2012, с. 324–335.
 19. Тимофеев А.В., Браницкий А.А. Исследование и моделирование нейросетевого метода обнаружения и классификации сетевых атак — International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE, VOLUME 6/2012, с. 257–265.
-

ИНФОРМАЦИОННЫЕ И МУЛЬТИ-АГЕНТНЫЕ ТЕХНОЛОГИИ УПРАВЛЕНИЯ КВАДРОКОПТЕРАМИ ДЛЯ МОНИТОРИНГА МОРСКОЙ ФАУНЫ

Данилов А.А.

*аспирант кафедры прикладной информатики и информатизации образования
ИМИКТ САФУ им. М.В.Ломоносова,
dluciv@math.spbu.ru*

Северный (Арктический) Федеральный университет им. М.В. Ломоносова

А.В. Тимофеев

*профессор кафедры информатики
Санкт-Петербургского государственного университета,
заведующий лабораторией информационных технологий
в управлении и робототехнике СПИИРАН,
tav@iias.spb.su*

*Санкт-Петербургский государственный университет,
Санкт-Петербургский институт информатики и автоматизации РАН*

Аннотация: Рассматриваются задачи и методы управления полётом квадрокоптеров с использованием современных информационных технологий. Обсуждаются возможности и перспективы групповой (мульти-агентной) навигации беспилотных летательных аппаратов на примере мониторинга морской фауны.

Введение

Актуальность данной работы связана с тем, что в случаях военных конфликтов, природных и техногенных катастроф, разведки и добычи полезных ископаемых в удалённых районах с суровым климатом возникает необходимость применения беспилотных летательных аппаратов (БПЛА) ввиду специфики ситуаций и сред, опасных для пилота обычных летательных аппаратов. Замена обычных летательных аппаратов на беспилотные позволит улучшить условия труда людей в опасных зонах, сохранить здоровье и жизнь людям в военных конфликтах, а также позволит получить экономический эффект [1–8].

В связи с бурным развитием информационных технологий применение математических моделей и методов математического моделирования для слежения и мониторинга с использованием беспилотных летательных аппаратов (БПЛА) становится всё более важной проблемой. Применение БПЛА позволит найти новые пути решения проблемы мониторинга и слежения за объектами морской фауны.

Для решения данной проблемы в первую очередь следует решить ряд соответствующих задач:

- разработать аппаратно-программный комплекс для решения задачи мониторинга объектов морской фауны;
- апробировать комплекс в акватории Белого моря.

Математическая и концептуальная модель квадрокоптера

Процесс разработки аппаратно-программного комплекса разбивается на следующие этапы:

- формулировка требований к разрабатываемой системе, которые в последствии будут задавать множество ограничений и допущений для концептуальной модели;
- математическая формализация концептуальной модели;
- постановка задачи и разработка численных методов её решения;
- создание программного комплекса для проведения численного эксперимента;
- создание аппаратного комплекса и интеграция его с разработанным программным комплексом;
- техническая проверка и апробация.

Разработанный БПЛА будет представлять собой квадрокоптер. Квадрокоптер – это летательный аппарат с четырьмя роторами, вращающимися диагонально в противоположных направлениях. В совокупности с программным комплексом разрабатываемый БПЛА должен обладать дистанционной системой навигации и управления полётом и встроенной системой стабилизации [1–3].

Для построения математической модели в первую очередь необходимо рассмотреть данный объект как на физическую систему. Для этого вводятся две системы координат (далее СК):

- неподвижная СК, $OXYZ$ оси OX и OY расположены в горизонтальной плоскости, ось OZ направлена вверх;
- подвижная СК, $O'X'Y'Z'$, в центре СК располагается центр масс квадрокоптера, дуги квадрокоптера сонаправлены плоскости $O'X'Y'$.

Подвижная система координат получается из неподвижной системы путем параллельного переноса на радиус-вектор и поворота на углы крена, тангажа и рыскания.

Базовая математическая модель, в основе которой лежит закон Ньютона, имеет вид:

$$m\vec{a} = mg\vec{e}_z + R(\psi, \theta, \varphi) \sum_{i=1}^4 \vec{F}_i,$$

где $R(\psi, \theta, \varphi)$ — матрица поворота подвижной СК относительно неподвижной, ψ, θ, φ — углы крена, тангажа и рыскания, \vec{F}_i — сила тяги каждого из 4-х двигателей.

Архитектура программно-аппаратного комплекса

Базовая математическая модель строится путем разложения векторных составляющих на координатную плоскость $OXYZ$ на основе дифференциальной зависимости. Схема аппаратной части изображена на рисунке 1.

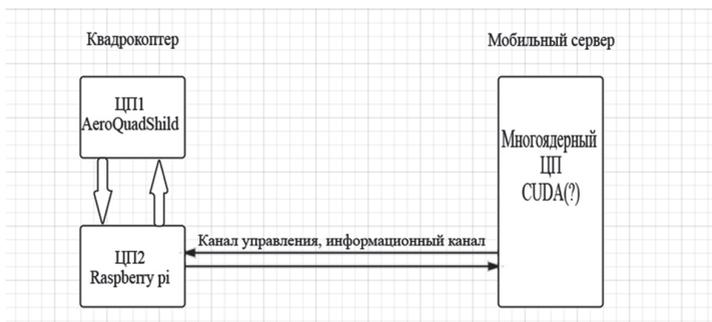


Рис. 1. Схема аппаратной части

Данная схема демонстрирует взаимосвязь трех центральных процессоров, каждый из которых выполняет свой набор функций.

Первые два центральных процессора (ЦП) находятся непосредственно на борту летательного аппарата.

Центральный процессор ЦП1 предназначен для стабилизации, базового управления полетом, предоставления интерфейса доступа к датчикам для второго ЦП.

Второй центральный процессор ЦП2 производит фото-видео съемку, обеспечивает канал связи с мобильным сервером, выполняет более сложные команды по управлению полетом (например, реализует систему приоритетов команд).

В схеме аппаратной части присутствует мобильный сервер с многоядерной архитектурой. Его задачи отличаются повышенной трудоёмкостью, а именно: обработка и распознавание объектов на фотоснимках, расчет траектории полета, навигация и телеметрия.

Данное разделение предлагается для получения оптимального разделения нагрузки на все процессоры и оптимизации времени реагирования на команды.

Данную схему планируется применить при построении программно-аппаратного комплекса для мониторинга и слежения за объектами морской фауны.

Заключение

В последнее время большое значение уделяется групповому (коллективному) использованию БПЛА и, в частности, квадрокоптеров [1–3]. Это по-

звolyет существенно расширить их функциональные возможности и надёжность. Однако навигация и управление полётом группы БПЛА представляет собой очень сложную задачу. Для её эффективного решения необходимы интеллектуальные, нейросетевые и мульти-агентные технологии навигации, управления полётом и функциональной диагностики. Эти новые подходы и технологии рассмотрены, например, в роботах [3–8].

Литература

1. Тимофеев А.В. Адаптивные робототехнические комплексы. Л.: Машиностроение, 1988.
2. Тимофеев А.В., Юсупов Р.М. Интеллектуализация систем автоматического управления. — Известия АН. Техническая кибернетика, 1994, № 5.
3. Тимофеев А.В., Юсупов Р.М. Принципы построения интегрированных систем мульти-агентной навигации и интеллектуального управления мехатронными роботами. — International Journal «Information Technologies and Knowledge», Vol. 5, Number 3, 2011, pp. 237–244.
4. Тимофеев А.В. Мульти-агентные системы и нейросетевые технологии в робототехнике и информатике. — Известия Кабардино-Балкарского научного центра РАН, № 1 (39), 2011, с. 249–252.
5. Тимофеев А.В. Методы высококачественного управления, интеллектуализации и функциональной диагностики автоматических систем. Часть I, Часть II. – Мехатроника, автоматизация, управление, 2003, № 5, 2004, № 2.
6. Cai Z., He H., Timofeev A. Navigation and Control of Mobile Robots in Unknown Environment: A Survey // Proceedings of 10-th International Conference on Integrated Navigation Systems (June 27–29, St-Petersburg). 2003. Vol. 1. P. 158–166.
7. Зотов Ю.К., Тимофеев А.В., Шишкин Д.С. Информационные технологии навигации и управления полетом малоразмерных летательных роботов корабельного базирования. — Информационно-измерительные и управляющие системы, № 8, т. 6, 2008.
8. Тимофеев А.В. Методы нейросетевого и мульти-агентного управления в робототехнике и мехатронике. // Нелинейная теория управления и её приложения. Динамика, управление, оптимизация. — М.: Физматлит. 2003, с. 101–126.]

ИНТЕЛЛЕКТУАЛИЗАЦИЯ ИНТЕГРИРОВАННЫХ СИСТЕМ НАВИГАЦИИ, УПРАВЛЕНИЯ ДВИЖЕНИЕМ И ФУНКЦИОНАЛЬНОЙ ДИАГНОСТИКИ РОБОТОВ

А.В. Тимофеев

зав. лабораторией информационных технологий в управлении и робототехнике СПИИРАН, профессор кафедры информатики Санкт-Петербургского государственного университета, tav@iias.spb.su

Санкт-Петербургский институт информатики и автоматизации РАН, Санкт-Петербургский государственный университет

Аннотация: Обсуждаются проблемы интеграции систем мульти-агентной (групповой) навигации, интеллектуального управления движением и функциональной диагностики роботов в динамической среде с препятствиями. Определяется понятия глобальной управляемости и стабилизируемости программных движений роботов как мехатронных агентов. Дается сравнительный анализ пяти поколений локальных систем навигации и управления движением роботов и стратегий централизованного, децентрализованного и мульти-агентного управления информационными потоками в сложных робототехнических системах. Описываются методы функциональной диагностики и алгоритмы дефектоустойчивого управления движением роботов в реальном времени при наличии разного рода дефектов и возмущений. Даются оценки времени правильного функционирования роботов, степени их неисправности и допустимых границ (допусков) различных классов дефектов и возмущений. Приводятся примеры использования предлагаемых методов для мехатронных роботов и мульти-агентных систем различного типа и назначения (манипуляционные и мобильные роботы, космические роботы, беспилотные летающие роботы и т. п.).

Введение

В последние годы в России и за рубежом значительное внимание уделялось исследованию проблем интеграции и интеллектуализации локальных систем автономной навигации и управления роботами и разработке глобальных стратегий групповой навигации и управления движением сложных робототехнических систем. В связи с бурным развитием систем навигации и управления движением мехатронных роботов как подвижных агентов с элементами искусственного интеллекта и необходимостью организации их коллективной работы возникла потребность в создании общих принципов построения и внедрения мульти-агентных робототехнических систем (МАРС). Такие МАРС объединяют группу агентов-роботов (например, мобильных

роботов или беспилотных летательных аппаратов) для достижения общей (глобальной) цели в динамической среде с препятствиями или запретными зонами и возможным противодействием [1–5].

Фундаментальные и поисковые исследования в области адаптивного и интеллектуального управления роботами и робототехническими системами (РТС) активно проводились в России в ряде научных школ (МГТУ, СПИ-ИРАН, СПбГУ, МГУ, МИРЭА, НИИ МВС, ЦНИИ РТК, ИПМ РАН и др.), начиная с 60-х годов XX века. Важная роль в становлении и развитии этих исследований принадлежит академикам И.М. Макарову, Д.Е. Охоцимскому, Е.П. Попову, Г.С. Поспелову и Ф.Л. Черноушко, членам-корреспондентам РАН В.А. Якубовичу, В.А. Лопоте, Г.Г. Сербрякову, Е.Д. Теряеву, Р.М. Юсупову и И.А. Каляеву и профессорам Е.И. Юревичу, В.М. Лохину, А.В. Тимофееву и др.

Проблемы и методы адаптивной навигации и интеллектуального управления движением роботов, а также задачи группового управления РТС обсуждались на ряде Международных и Российских конференций, проведённых в последние десятилетия в России, в том числе на 2-х Международных конференциях «Мехатроника и робототехника», 22-х конференциях «Экстремальная робототехника», 7-и конференциях «Мехатроника, автоматизация, управление», 6-и конференциях «Управление и информационные технологии» и 3-х Мультиконференциях по проблемам управления.

Параллельно в России и за рубежом проводились поисковые исследования и опытно-конструкторские разработки по созданию интегрированных систем навигации и управления движением объектов и роботов различного типа и назначения для экстремальных сред и частично неопределённых условий эксплуатации. Достижения специалистов из России и зарубежный опыт в этой области обсуждались на 20-ти Международных конференциях по интегрированным навигационным системам, организованных ЦНИИ «Электроприбор», и на 33-х научных сессиях Международной академии навигации и управления движением, президентом которой является академик В.Г. Пешехонов.

Первый международный проект по исследованию MAPC на тему «Multi-Agent Robot Systems for Industrial Applications in the Transport Domain» был выполнен в 1997–1999 годах по Европейской программе COPERNICUS. Координатором этого проекта был Prof., Dr.–Ing. U.Rembold (University of Karlsruhe, Institute for Process Control and Robotics, Germany), а его участниками — Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (Russia), Unite de Recherche INRIA Rhone—Alpes (France), Technical University of Budapest (Hungary), Technical University of Poznan (Poland), Belorussian State University, (Belarus), Ufa State Aviation Technical University (Russia) и корпорации Daimler-Benz AG (Germany) и Beta Computer Automation (Germany).

Развиваемые в настоящем докладе новые информационные и телекоммуникационные технологии и интегрированные интеллектуальные системы

навигации и управления движением для МАРС в значительной степени базируются на оригинальных научных результатах поисковых и ориентированных фундаментальных исследований, полученных автором доклада в последние годы при поддержке грантов Российского фонда фундаментальных исследований (РФФИ), программ Президиума РАН и государственных заказов Министерства образования и науки РФ (см., например, [5–14]).

Важным средством интеллектуализации интегрированных систем навигации и управления движением роботов и РТС является встроенные средства функциональной диагностики в реальном масштабе времени. [5, 21, 22]. Поэтому в докладе обсуждаются методы функциональной диагностики и алгоритмы дефекутоустойчивого управления вместе с оценками допустимых границ для разных классов дефектов и возмущений.

Глобальная управляемость и оптимизация программных движений РТС

Рассмотрим некоторую робототехническую систему (РТС) R , состоящую из n роботов r_i , $i=1, 2, \dots, n$. Обычно роботы и РТС функционируют в динамической среде E с препятствиями или запретными зонами $O_j \in E$, $j=1, 2, \dots, q$. В роли динамических препятствий для одних роботов r_i из РТС могут выступать другие роботы r_j , $j \neq i$.

РТС будем называть гомогенной, если она состоит из однотипных роботов. Сложную РТС будем называть гетерогенной, если её структура включает в себя несколько разнотипных роботов или гомогенных РТС.

Архитектура гомогенных и гетерогенных РТС по существу является сетевой и территориально распределённой, т.е. включает в себя множество роботов, взаимодействующих между собой с помощью физических, сенсорных (информационных), управляющих и коммуникационных каналов прямой и обратной связи. Поэтому возникает потребность в разработке новых принципов сетевой организации, управления движением, навигации, обработки и передачи информации в сложных РТС в динамической окружающей среде с препятствиями или запретными зонами. Необходимость в сетевом управлении и групповой навигации возникла прежде всего в глобальных инфотелекоммуникационных сетях (например, в Internet), в робототехнике и гибких автоматизированных производственных системах, а также в сложных автоматизированных системах вооружений (например, в сетцентрических и мульти-агентных системах оборонного назначения).

Глобальная (общая) цель управления РТС заключается в том, чтобы синтезировать такой закон группового управления локальными технологическими операциями роботов, который обеспечивает перевод РТС R и окружающую её среду E из заданного начального состояния в желаемое конечное (целевое) состояние с учётом заданных ограничений на вектор сетевого управления и вектор состояний.

РТС будем называть глобально управляемой в динамической среде, если существует закон группового управления (вообще говоря, зависящий от текущего состояния РТС и среды), обеспечивающий выполнение граничных условий с учетом имеющихся ограничений. Соответствующее этому групповому управлению движение будем называть программным движением (ПД) в динамической среде $E(t)$ с препятствиями или запретными зонами.

Среди множества групповых управлений и ПД РТС можно выделить наилучшее (оптимальное) управление и соответствующее ему оптимальное ПД.

Следует отметить, что критерии глобальной управляемости роботов с нелинейной динамикой, а также методы аналитического синтеза и оптимизации ПД мехатронных роботов с учётом заданных ограничений при наличии препятствий были предложены и подробно описаны в работах [1, 5, 12, 19, 20].

Стабилизация, декомпозиция и синхронизация программных движений роботов

Предположим, что ПД РТС и соответствующий ему закон программного группового управления существуют. Тогда ПД называется практически (или асимптотически) устойчивым по отношению к начальным возмущениям (или, возможно, к другим возмущениям), если существует закон группового управления роботами такой, что в замкнутой этим управлением РТС и окружающей её динамической среде обеспечивается осуществление ПД с заданной точностью.

Закон группового управления РТС, обеспечивающий выполнение целевого условия, будем называть стабилизирующим. Этот закон управления является декомпозирующим, если целевые условия выполняются независимо для каждого робота из РТС. Это означает, что локальное управление каждым роботом r_i осуществляется независимо от локального управления другими роботами r_j , $j \neq i$, с компенсацией возможных перекрёстных динамических связей между роботами РТС [1, 5].

В некоторых случаях (например, на конвейерах) от закона группового управления требуется обеспечить полную или частичную синхронизацию движений различных роботов из РТС.

На практике возникают также задачи группового управления роботами, связанные с достижением консенсуса (consensus), когда каждый робот из РТС стремится, чтобы его ПД было близко к ПД своих соседей, или рандеву (rendezvous), когда все роботы и РТС должны попасть в желаемое состояние в заданный момент времени.

Следует отметить, что для локальных систем навигации и управления движением мехатронных роботов с нелинейной динамикой законы стабилизирующего, модального (спектрального) и декомпозирующего управления, а также алгоритмы планирования поведения и маршрутизации безопасных маршрутов движения роботов в среде с препятствиями были предложены и детально описаны в работах [1–7, 19, 20–24].

Пять поколений локальных систем навигации и управления движением роботов

В работах [1–5] была предложена классификация локальных систем навигации и управления движением (СНУД) роботов на пять поколений:

- 1) программные СНУД;
- 2) адаптивные СНУД;
- 3) интеллектуальные СНУД;
- 4) нейросетевые СНУД;
- 5) креативные СНУД.

Программные СНУД роботов основаны на классических принципах программного или оптимального управления движением, если модель динамики роботов или РТС полностью известна, а среда (или связанные с ней возмущения) известны и неизменны.

Адаптивные СНУД базируются на современных принципах робастного или адаптивного управления, когда модель динамики РТС и среды (или связанных с ней возмущений) полностью или частично неизвестны. В этом случае программные СНУД дополняются специальными средствами адаптации или идентификации факторов неопределённости или нестационарности, к числу которых можно отнести неизвестные возмущения или препятствия [1–4].

Интеллектуальные СНУД дополняются не только средствами адаптации, но и некоторыми элементами (алгоритмами) искусственного интеллекта (например, способностью распознавать речевые команды или идентифицировать неизвестные препятствия). Как правило, эти элементы искусственного интеллекта описываются логическими, алгебраическими и нечёткими алгоритмами. Поэтому они могут быть программно реализованы на традиционных микропроцессорах и компьютерах [1–3].

Нейросетевые СНУД основываются на обучении, самоорганизации и распараллеливании процессов обработки информации, навигации и управления на нейронных сетях или нейрокомпьютерах [6, 27]. В этом заключается их основное отличие и преимущество по сравнению с СНУД роботов первых трёх поколений.

Креативные СНУД ещё слабо изучены. Они могут базироваться на когнитивных и мульти-агентных технологиях, системах виртуальной или дополненной реальности, процессах моделирования сознания и т. п. [24, 25].

Глобальные стратегии навигации и сетевого управления движением РТС

Важно отметить, что роботы редко используются изолированно, т. е. в полуавтономном или автономном режимах эксплуатации. Обычно они входят в состав РТС и предназначены для группового (коллективного) вы-

полнения некоторой сложной общей (глобальной) задачи, которую каждый робот самостоятельно (автономно) решить не может [1].

Сложная гетерогенная РТС может включать в себя десятки или сотни роботов или гомогенных РТС, функционирующих в динамической среде с препятствиями. В связи с этим важное значение приобретают проблемы системного анализа и сетевого синтеза интегрированных систем групповой навигации, управления движением и обработки сенсорной информации. Однако проектирование и создание интегрированных систем невозможно без организации и координации взаимодействия роботов в РТС с помощью коммуникационной сети [1–6].

Математической моделью такой, вообще говоря, динамической коммуникационной сети РТС может служить коммуникационный граф вида [10, 24, 25].

В процессе решения общей (глобальной) задачи структура и параметры РТС могут изменяться (например, некоторые роботы или каналы связи между ними могут выйти из строя). Тогда будут изменяться структура (топология узлов и каналов связи) или параметры (веса) коммуникационного графа. Этот граф характеризует информационное и коммуникационное взаимодействие между роботами в РТС. Поэтому его структура тесно связана со стратегией групповой навигации и управления ПД роботов и РТС, обеспечивающей достижение общей (глобальной) цели управления в динамической среде с препятствиями.

Первоначально (начиная с 70-х годов XX века) стратегии группового управления роботами в РТС разделялись на два класса [1, 16, 17]:

- стратегии централизованного (глобального) управления РТС из общего командного центра;
- стратегии децентрализованного (распределённого) управления, основанные на локальном управлении каждым роботом из РТС и их взаимодействии.

Преимущества стратегии централизованного (глобального) управления РТС заключаются в простоте иерархической организации и программно-аппаратной реализации систем группового (в том числе оптимального) управления на центральном компьютере. Однако её недостатки проявляются в низкой живучести (надёжности) и возможности искажений и временных задержек при передаче команд программного управления от верхнего уровня иерархии к нижнему.

Преимущества стратегии децентрализованного (локального распределённого) управления роботами из РТС заключаются в высокой локальной производительности (малом времени принятия локальных решений) и параллелизме локального управления роботами, а также в повышенной живучести (надёжности) РТС. В тоже время ей свойственны такие недостатки, как повышенные требования к надёжности узлов и каналов информационной связи между роботами и невысокое качество группового управления

РТС, т. к. даже из локальной оптимальности управления отдельными роботами, вообще говоря, не следует глобальная оптимальность управления РТС в целом.

Сравнительный анализ преимуществ и недостатков традиционных стратегий централизованного и децентрализованного управления территориально распределённых РТС привёл к необходимости разработки (начиная с 90-х годов XX века) новой гибридной (смешанной) стратегии мульти-агентной навигации и группового (коллективного) управления РТС на современном уровне развития робототехники, мехатроники и инфотелекоммуникационных технологий [3, 5, 9–11]. Этот уровень развития характеризуется всё более широким внедрением роботов с интеллектуальным и нейросетевым управлением, глобальных систем навигации типа ГЛОНАСС и (или) GPS и инфотелекоммуникационных систем типа Internet и Grid [24, 25].

Функциональная диагностика роботов и дефектоустойчивое управление

Проблемы технической диагностики играют важную роль в процессе проектирования, испытания и эксплуатации автономных роботов и сложных РТС. Среди этих проблем большое значение имеют задачи функциональной диагностики, когда контроль дефектов и диагностика неисправностей осуществляются в реальном времени непосредственно в процессе эксплуатации роботов и РТС.

Функционирование роботов и РТС в значительной степени зависит не только от возможных неисправностей и дефектов, но и от используемых алгоритмов управления [5, 21, 22].

Цель навигации и управления движением роботов и РТС в таких системах заключается в построении желаемого (программного) движения и синтезе алгоритмов управления, обеспечивающих фактическое осуществление этого программного движения (ПД) при наличии известных или неизвестных препятствий, возмущений и дефектов из определённого класса, заданных своими границами (допусками) [5–21]. Для контроля гарантированного достижения этой цели необходимы функциональное диагностирование в процессе навигации и управления движением, а также, по возможности, обнаружение, локализация и компенсация неисправностей (дефектов).

Функциональная диагностика роботов и РТС основывается на адекватных моделях динамики объекта управления и моделях возможных неисправностей (дефектов) в замкнутой системе. Такие модели можно построить в аналитической форме для линейных и нелинейных обратимых моделей динамики роботов, описываемых дифференциальными уравнениями динамики, разрешимыми относительно управления на некотором подпространстве [1, 5, 20].

Важную роль играют методы функционального диагностирования роботов и алгоритмы дефектоустойчивого управления их движением. В ра-

ботах [5, 21, 22] показано, что эти алгоритмы являются робастными или адаптивными по отношению к различным типам возможных дефектов и неконтролируемым возмущениям. Даются оценки допустимых границ (допусков) различных дефектов и возмущений [21, 22].

Предлагаемые методы основываются на прямых и обратных (на подпространстве) моделях динамики мехатронных роботов [5, 21, 22]. В докладе даётся классификация моделей дефектов и приводятся оценки основных показателей правильного функционирования и степени неисправности [21, 22]. Большое внимание уделяется синтезу и анализу программного, стабилизирующего, робастного и адаптивного управления движением мехатронных роботов, обеспечивающих инвариантность переходных процессов (динамической ошибки) в различных (узких и широких) классах возможных дефектов. Показано, что использование алгоритмов идентификации неизвестных параметров роботов и внешних возмущений с конечным временем адаптации [1–5] позволяет не только обнаружить и локализовать дефекты из широкого класса неопределённости, но и автоматически их компенсировать.

Принципы интеграции систем навигации, управления движением и диагностики в МАРС

В современной робототехнике, а также в информационных технологиях навигации и теории управления движением сложных мехатронных и аэрокосмических систем существует ряд проблем, связанных с проектированием, созданием и интеграцией систем адаптивной (локальной) и мульти-агентной (групповой) навигации, функциональной диагностики и интеллектуального управления движением мехатронных роботов в динамической среде с известными или неизвестными препятствиями. Некоторые из этих проблем, а также перспективные подходы и новые методы их решения рассмотрены в работах [1–25]. При этом важную роль играют принципы проектирования интегрированных систем мульти-агентной навигации, интеллектуального управления и функциональной диагностики применительно к мобильным роботам и РТС.

Мобильный робот рассматривается как интеллектуальный подвижный мехатронный агент, имеющий на борту [1, 24]:

- сенсорную систему датчиков внутренней и внешней информации;
- интеллектуальную систему навигации, управления движением и функциональной диагностики;
- двигательную систему с традиционным (например, гусеничным или четырёхколёсным) или нетрадиционным (например, с одноосным двухколёсным или ползающим, т. е. перемещающимся по поверхностям произвольной ориентации) шасси;
- коммуникационную систему для информационного взаимодействия с человеком-оператором и другими роботами-агентами.

Важными задачами при создании мобильных мехатронных роботов как подвижных агентов является интеллектуализация и интеграция бортовых интегрированных систем навигации, управления движением и функциональной диагностики, обеспечивающая возможность автоматического планирования и оптимизации движений для обхода известных или неизвестных препятствий, адаптации к динамическим факторам неопределённости (например, к возмущениям и неизвестным подвижным препятствиям) и к возможным дефектам или отказам, а также распознавания образов (например, стерео-изображений препятствий), моделирования и анализа сложных 3D-сцен и диагностики состояний мобильного робота-агента в реальном масштабе времени.

Новые актуальные проблемы возникают при групповом (коллективном) использовании мобильных мехатронных роботов-агентов в составе МАРС для решения общей (глобальной) задачи в динамической среде с препятствиями. Группа (коллектив) мобильных мехатронных роботов-агентов, объединённых общей (глобальной) целью и функционирующая в едином информационном и коммуникационном пространстве на базе соответствующих стандартов, команд и форматов передачи потоков данных, называется мобильной МАРС.

Ключевыми проблемами при создании интегрированных систем навигации и управления движением мобильных МАРС являются [1–6, 21–25]:

- декомпозиция общей (глобальной) задачи, возлагаемой на группу (коллектив) мобильных роботов, на ряд локальных задач для каждого робота-агента,
- мульти-агентное (групповое) планирование маршрутов движения и навигация мобильных роботов-агентов в динамической среде с известными или неизвестными препятствиями,
- функциональная диагностика и отказоустойчивое управление движением мобильных роботов-агентов,
- интеллектуализация и интеграция систем навигации, управления движением и диагностики на базе бортовых средств распознавания сложных образов (мульти-изображений, речевых команд и т.п.), мобильных навигаторов (спутниковые системы ГЛОНАСС и GPS и т.п.) и функциональной диагностики в составе мобильных МАРС.

Важными задачами являются также организация интеллектуального человеко-машинного интерфейса и комплексирование перспективных решений этих проблем для создания интегрированных систем мульти-агентной навигации, интеллектуального управления движением и функциональной диагностики мобильных роботов как агентов мехатронных МАРС, способных функционировать в реальном времени в динамической среде с препятствиями.

Заключение

Актуальность проведённых исследований связана с повышенным интересом во всём мире к интеллектуальным мехатронным роботам как под-

важным агентам в МАРС ввиду их способности автоматически функционировать под контролем человека как в традиционных отраслях производства (машиностроение, приборостроение, микроэлектроника и т.п.), так и в нетрадиционных областях и экстремальных средах (космическая и подводная робототехника, атомная энергетика, ликвидация техногенных аварий и террористических угроз и т.п.). Важно отметить, что круг задач, решаемых мехатронными роботами-агентами и мобильными МАРС, непрерывно расширяется и усложняется (необходимость эффективно работать в экстремальных средах в условиях неопределённости или противодействия при наличии препятствий или запретных зон и т.п.).

Большую научную значимость и актуальность имеют поисковые исследования в области создания интегрированных систем мульти-агентной (групповой) навигации, интеллектуального управления движением и функциональной диагностики мехатронных роботов-агентов и мобильных МАРС в динамических (изменяющихся) средах с препятствиями. Сегодня этим инновационным исследованиям и опытно-конструкторским разработкам во всём мире уделяется приоритетное внимание.

Работа выполнена при частичной поддержке грантов РФФИ №09-08-00767-а, №12-08-01167-а и РФФИ-ГФЕН Китая №10-08-91159-ГФЕН-а и Программы №14 «Проблемы создания национальной научной распределенной информационно-вычислительной среды на основе GRID-технологий, облачных вычислений и современных телекоммуникационных сетей» Президиума РАН.

Литература

1. Тимофеев А.В. Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях. — СПб.: Анатолия, 2012, 280 с.
2. Тимофеев А.В., Сырцев А.В. Модели и методы маршрутизации потоков данных в телекоммуникационных системах с изменяющейся динамикой. — М.: Новые технологии, 2005, 82 с.
3. Тимофеев А.В., Сгурев В.В., Йотсов В.С., Лютикова Л.В. Развитие и применение многозначных логик и сетевых потоков в интеллектуальных системах. — Труды СПИИРАН, вып. 2, 2004. С. 72–84.
4. Тимофеев А.В. Оптимизационный синтез и минимизация сложности генно-нейронных сетей по целочисленным базам данных. — Нейрокомпьютеры: разработка и применение, 2002, №5–6. С. 34–39.
5. Timofeev A.V. Polynomial Neural Network with Self-Organizing Architecture. — International Journal on Optical Memory and Neural Networks, 2004, N 2.
6. Timofeev A.V. Parallelism and Self-Organization in Polynomial Neural Networks for Image Recognition. — Pattern Recognition and Image Analysis, 2005, vol. 15, No. 1, pp. 97–100.
7. Тимофеев А.В., Сырцев А.В. Мультиагентная и нейросетевая маршрутизация потоков данных в телекоммуникационных сетях // Труды 10-й международной кон-

- ференции «Knowledge—Dialogue—Solution» (16–26 июня. 2003, Варна) 2003. С. 187–190.
8. Тимофеев А.В. Модели мульти-агентного диалога и информационного управления в глобальных телекоммуникационных сетях // Труды 10-й международной конференции «Knowledge—Dialogue—Solution» (16–26 июня, 2003, Варна). 2003. С. 180–186.
 9. *Timofeev Adil*. Adaptive Routing and Multi-Agent Control for Information Flows in IP-Networks. — Proceedings of Xith International Conference «Knowledge—Dialogue—Solution» (KDS-2005), June 20–30, Varna, Bulgaria, 2005. Pp. 442–445.
 10. *Syrzhev A.V., Timofeev A.V.* Neural and Multi-Agent Routing in Telecommunicational Networks. — International Journal «Information Theories and Their Applications», 2003, № 2.
 11. *Timofeev A.V.* Models for Multi-Agent Dialogue and Informational Control in Global Telecommunicational Networks. — International Journal «Information Theories and Their Applications», 2003, № 1.
 12. *Timofeev A.V.* Multi-Agent Information Processing and Adaptive Control in Global Telecommunication and Computer Networks. — International Journal «Information Theories and Their Applications», 2003, № 10. Pp. 54–60.
 13. *Timofeev A.V.* Intellectualization for Man-Machine Interface and Network Control in Multi-Agent Infotelecommunication Systems of New Generation. — Proceedings of 9th International Conference «Speech and Computer» (20–22 September, 2004), Saint-Petersburg, Russia. Pp. 694–700.
 14. *Timofeev A.V.* Adaptive Control and Multi-Agent Interface for Infotelecommunication Systems of New Generation. — International Journal «Information Theories & Applications». Vol.11, 2004.
 15. *Амбарян Т., Тимофеев А.* Модели квантовых и нейронных вычислений в задачах обработки информации. // Известия вузов. Приборостроение 2005, № 7. С. 35–40.
 16. *Тимофеев А.В., Дмитриченко Д.П.* Многокритериальная оценка сетевых топологических структур для моделирования и проектирования GRID-систем — Труды СПИИРАН. Вып. № 10. СПб.: Наука, 2008. С. 72–77.
 17. *Тимофеев А.В.* Фрактальное моделирование и многокритериальная оптимизация компьютерных сетей. — International Book Series Information Science & Computing. Intelligent Engineering, vol 3/2009, № 11. Pp.79–83.
 18. *Косовская Т.М., Тимофеев А.В.* Логико-когнитивные методы распознавания и анализа сложных изображений и сцен. — International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE, VOLUME 6/2012, с. 324–335.
 19. *Тимофеев А.В., Браницкий А.А.* Исследование и моделирование нейросетевого метода обнаружения и классификации сетевых атак — International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE, VOLUME 6/2012, с. 257–265.
 20. *Тимофеев А.В.* Адаптивное управление и интеллектуальный анализ информационных потоков в компьютерных сетях. — СПб.: Анатолия, 2012, 280 с.
-

Автоматное программирование, машинное обучение и биоинформатика



**Шалыто
Анатолий Абрамович**

Д.Т.Н.
профессор,
заведующий кафедрой технологии программирования
СПбГУ ИТМО

ПОСТРОЕНИЕ АВТОМАТОВ УПРАВЛЕНИЯ ОБЪЕКТАМИ СО СЛОЖНЫМ ПОВЕДЕНИЕМ ПО ТЕСТАМ С УЧЕТОМ НЕПРЕРЫВНЫХ ВОЗДЕЙСТВИЙ

И.П. Бужинский

*студент кафедры «Компьютерные технологии» НИУ ИТМО,
buzhinsky@rain.ifmo.ru*

В.И. Ульянов

*студент кафедры «Компьютерные технологии» НИУ ИТМО,
ulyantsev@rain.ifmo.ru*

Аннотация: Предлагается улучшение метода автоматизированного построения конечных управляющих автоматов. Рассматриваемые автоматные модели характеризуются наличием непрерывных (вещественных) выходных воздействий. Улучшение опробовано на задаче построения автомата управления моделью беспилотного самолета.

Введение

Автоматное программирование [1] — парадигма, в рамках которой программы проектируются в виде систем автоматизированных объектов управления, каждый из которых состоит из объекта управления и управляющего конечного автомата. Автоматное программирование применяется для задач управления объектами со сложным поведением — объектами, которые могут демонстрировать различное поведение при одинаковых входных воздействиях. В настоящем исследовании рассматривается задача построения автоматов управления объектами с непрерывными (вещественными) управляющими параметрами по обучающим примерам, или тестам.

В работе [2] для создания автомата с дискретными выходными воздействиями, управляющего моделью беспилотного самолета, был использован генетический алгоритм. При этом функция приспособленности автомата вычислялась на основе моделирования в авиасимуляторе, а полный цикл построения автомата занимал месяц работы персонального компьютера. В работе [3] был также применен генетический алгоритм, но функция приспособленности автомата вычислялась на основе тестов, что позволило сократить время построения автомата до нескольких часов. Другой отличительной особенностью работы [3] является использование непрерывных выходных воздействий. Метод построения автоматов, предложенный в [3], был позже ускорен [4] за счет использования муравьиного алгоритма [5].

Целью настоящего исследования является качественное улучшение метода построения автоматов, предложенного в работе [3]. Предлагается новый способ представления автоматов, позволяющий использовать для задания

функции выходов вещественные переменные — произвольные функции от параметров объекта управления.

Постановка задачи

Рассматриваемая задача заключается в построении конечного управляющего автомата по набору из N тестов, которые описывают желаемое поведение объекта управления, обладающего сложным поведением. В качестве объекта управления в настоящей работе, как и в работах [2–4], рассматривается модель беспилотного самолета. Для моделирования работы автоматов используется авиасимулятор *FlightGear* [6].

Тест состоит из последовательности P -элементных *кортежей входных воздействий* $\text{in}[i] = (\text{in}[i][t])_{t=1}^{\text{len}[i]}$ и последовательности C -элементных *кортежей выходных воздействий* $\text{out}[i] = (\text{out}[i][t])_{t=1}^{\text{len}[i]}$, где $i = 1 \dots N$ — номер теста, а $\text{len}[i]$ — его длина. Входные воздействия описывают состояние объекта управления в различные моменты времени (интервал между ними в настоящей работе равен 0,1 с). По кортежам входных воздействий вычисляются значения булевых переменных — *предикатов*. Кортежи выходных воздействий представляют собой «эталонные» значения управляющих параметров — положений органов управления самолета. Будем считать, что значения всех управляющих параметров можно задать вещественными числами. Пример теста, записанного с помощью авиасимулятора *FlightGear*, приведен в Таблице 1.

Воздействие	Комментарий	$t=1$...	$t=10$...	$t=235$
$\text{in}[i][t][1]$	Угол крена (°)	-0,076	...	-0,076	...	-0,076
$\text{in}[i][t][2]$	Угол курса (°)	198,03	...	198,11	...	205,64
$\text{in}[i][t][3]$	Скорость (узлы)	251,42	...	252,29	...	289,40
$\text{out}[i][t][1]$	Положение элеронов (число от -1 до 1)	0,000	...	0,032	...	-0,003
$\text{out}[i][t][2]$	Положение руля высоты (число от -1 до 1)	-0,035	...	-0,039	...	-0,011

Таблица 1. Пример теста ($P=3$, $C=2$, $\text{len}[i]=235$)

Базовые положения исследования

Для построения автомата на основе поисковой оптимизации используется незначительно модифицированная *функция приспособленности* (ФП) из работы [3], отражающая близость поведения автомата к поведению, показанному в тестах:

$$f = 1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{1}{\text{len}[i]} \sum_{t=1}^{\text{len}[i]} \frac{1}{C} \sum_{j=1}^C \left(\frac{\text{ans}[i][t][j] - \text{out}[i][t][j]}{c_j^{\max} - c_j^{\min}} \right)^2} - P_{\tau}$$

Здесь c_j^{\min} и c_j^{\max} — границы значений j -го управляющего параметра, $\text{ans}[i]$ — последовательность значений управляющих параметров, выработанная автоматом в ответ на $\text{in}[i]$, а P_τ — дополнительный штраф, возрастающий с числом изменений состояния автомата в процессе его работы на тестах. Особью алгоритма поисковой оптимизации является «каркас» автомата — автомат, выходные воздействия которого не определены. Для их определения используется алгоритм, максимизирующий ФП на заданном «каркасе».

В работе [3] для представления автомата использовались полные таблицы переходов. Каждому переходу был сопоставлен кортеж выходных воздействий. За такт работы автомата совершалось несколько переходов, при этом кортежи выходных воздействий на всех выполнившихся переходах суммировались. В настоящей работе предлагается совместно с методом построения автоматов, предложенным в [3], использовать новый способ представления автоматов, имеющих непрерывные выходные воздействия.

Для представления функции переходов используются сокращенные таблицы [2]: в каждом состоянии автомата хранится маска значимости предикатов (Таблица 2), а событиями на переходах автомата являются всевозможные комбинации значений предикатов, помеченных единицей в маске значимости. На каждом такте автомат совершает только один переход, что упрощает как его визуализацию, так и понимание роли отдельных его состояний.

Предикат	Значение маски
p_1	1
p_2	0
p_3	1

Таблица 2. Пример маски значимости предикатов для одного из состояний автомата

Ключевым компонентом для представления функции выходов являются *вещественные переменные*, которые, как и предикаты, представляют собой функции от кортежей входных воздействий. Для каждого управляющего параметра $j = 1 \dots C$ задается набор $\{v_{j,i}\}_{i=1}^{n_j}$ из n_j вещественных переменных. В каждом состоянии автомата хранится маска значимости вещественных переменных (Таблица 3): только помеченные единицей в маске переменные влияют на формирование соответствующих им выходных воздействий. Все маски, используемые для представления функций переходов и выходов, являются частью особи алгоритма поисковой оптимизации.

Переменная	Значение маски	Переменная	Значение маски
$v_{1,1}$	1	$v_{2,1}$	0
$v_{1,2}$	1	$v_{2,2}$	1
$v_{1,3}$	0	$v_{2,3}$	1

Таблица 3. Пример маски значимости вещественных переменных для одного из состояний автомата ($C = 2$, $n_1 = n_2 = 3$)

Пусть s — текущее состояние автомата. Тогда изменение Δu_j выходного воздействия для j -го управляющего параметра по отношению к предыдущему такту работы автомата вычисляется как линейная комбинация значений вещественных переменных:

$$\Delta u_j = \sum_{i=1}^{n_i} r_{s,i,j} v_{j,i},$$

где $r_{s,i,j}$ — коэффициенты, вычисляемые алгоритмом расстановки выходных воздействий [3]. Эти коэффициенты заведомо равны нулю для незначимых в состоянии s переменных.

Пример автомата с предлагаемым способом представления приведен на Рис. 1. В каждом состоянии автомата значимыми являются один предикат и две переменные, при этом всего заданы три предиката p_1, p_2, p_3 и три переменные v_1, v_2, v_3 для единственного управляющего параметра u .

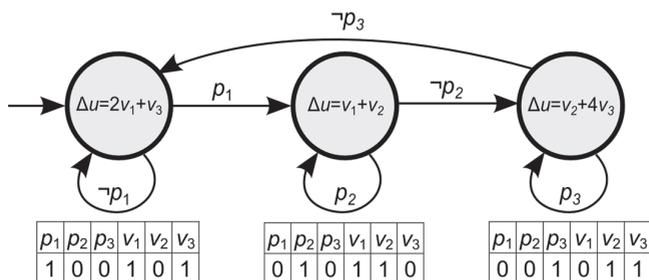


Рис. 1. Пример автомата с предлагаемым способом представления. Внутри состояний приведены законы управления параметром u , внизу — маски значимости переменных и предикатов, соответствующие различным состояниям

Экспериментальное исследование

Сравнение предлагаемого метода представления автоматов с ранее разработанным проведено на трех наборах тестов, описывающих выполнение фигур пилотажа: «мертвой петли», «бочки» и разворота в горизонтальной плоскости на 180° .

Для использования способа представления автоматов из работы [3] требуется задать набор предикатов, а для предлагаемого способа помимо этого необходимо определить набор вещественных переменных. Подбор набора переменных осуществляется вручную и является итеративным процессом: если качество выполнения фигуры пилотажа построенным автоматом является неудовлетворительным, набор переменных корректируется. Задача подбора предикатов не удалось полноценно решить для выполнения разво-

рота при использовании ранее разработанного способа: подавляющая часть построенных автоматов не смогла его выполнить, несмотря на высокое значение ФП.

Поиск автоматов проводился при помощи муравьиного алгоритма [5]. Для каждого набора тестов, способа представления автоматов и числа состояний автомата от трех до пяти были проведены 50 запусков алгоритма. Критерием остановки запуска была стагнация в течение 5000 вычислений ФП, при этом продолжительность запусков муравьиного алгоритма для различных конфигураций составляла от 9000 до 14000 вычислений ФП. В Таблице 4 приведены средние значения ФП, достигнутые по окончании запусков алгоритма. Для каждого набора тестов в левой колонке приведены значения ФП для ранее разработанного метода, в правой — для предлагаемого.

Число состояний	«Мертвая петля»		«Бочка»		Разворот	
	0,9814	0,9855	0,9832	0,9851	0,9891	0,9893
3	0,9814	0,9855	0,9832	0,9851	0,9891	0,9893
4	0,9834	0,9866	0,9854	0,9862	0,9900	0,9899
5	0,9838	0,9873	0,9856	0,9865	0,9900	0,9901

Таблица 4. Средние значения ФП, достигнутые на различных наборах тестов при использовании двух способов представления автомата

Как видно из таблицы, использование предлагаемого способа представления автомата, как правило, позволяет получить автоматы с более высоким значением ФП. Качество сгенерированных автоматов оценивалось при моделировании в авиасимуляторе. Исследования показали, что автоматы с предложенным способом представления, построенные для выполнения «мертвой петли», лучше управляют креном, а автоматы, построенные для «бочки», лучше управляют и креном, и тангажом. На тестах, описывающих разворот, с использованием обоих способов представления удалось построить автоматы с высокими значениями ФП, однако выполнить разворот смогли только автоматы с предлагаемым способом представления.

Для проведения вычислительных экспериментов использовался персональный компьютер с четырехъядерным процессором *Intel Core 2 Quad Q9400*. Время построения автоматов занимало порядка пяти минут при использовании предыдущего и порядка десяти минут при использовании нового способа представления.

Заключение

Предложен новый способ представления автоматов с непрерывными выходными воздействиями, улучшающий метод построения автоматов [3]. По сравнению с ранее разработанным способом он показывает лучшие ре-

зультаты по качеству построенных автоматов, а также делает логику работы автоматов более наглядной.

Литература

1. *Поликарпова Н.И., Шалыто А.А.* Автоматное программирование. СПб.: Питер, 2010.
 2. *Поликарпова Н.И., Точилин В.Н., Шалыто А.А.* Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования // Известия РАН. Теория и системы управления. 2010. №2. С. 100–117.
 3. *Александров А.В., Казаков С.В., Сергушичев А.А., Царев Ф.Н., Шалыто А.А.* Применение генетического программирования на основе обучающих примеров для генерации конечных автоматов, управляющих объектом со сложным поведением // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики, 2011, №2 (72). С. 3–11.
 4. *Бужинский И.П., Ульянов В.И.* Применение муравьиных алгоритмов для построения автоматов управления системами со сложным поведением на основе обучающих примеров / Сборник докладов XV Международной конференции по мягким вычислениям и измерениям (SCM'2012). СПб: СПбГЭТУ «ЛЭТИ», 2012. Т. 1, С. 250–253
 5. *Chivilikhin D., Ulyantsev V.* Learning Finite-State Machines with Ant Colony Optimization // Lecture Notes in Computer Science, 2012, Volume 7461/2012, pp. 268–275
 6. FlightGear [Электронный ресурс]. Режим доступа <http://www.flightgear.org/> свободный. Яз. англ. (дата обращения 20.04.13).
-

МЕТОД ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ НА ОСНОВЕ МУРАВЬИНОГО АЛГОРИТМА

Чивилихин Д.С.

*магистрант кафедры компьютерных технологий НИУ ИТМО,
chivilikhin.daniil@gmail.com*

Ульянцев В.И.

*магистрант кафедры компьютерных технологий НИУ ИТМО,
ulyantsev@rain.ifmo.ru*

Аннотация: Предлагается новый метод построения конечных автоматов, основанный на сведении этой задачи к поиску на графе и применении муравьиного алгоритма нового типа для поиска решений в этом графе. Результаты экспериментального исследования предложенного метода позволяют говорить о его высокой эффективности по сравнению с генетическим алгоритмом.

Введение

В парадигме автоматного программирования [1] выделяется система управления, представляемая в виде конечного автомата, и объект управления. В некоторых случаях автомат управления может быть построен вручную, однако для большинства объектов со сложным поведением этот процесс является весьма трудоемким. Для автоматизированного построения конечных автоматов в основном применяются эволюционные алгоритмы [2–4].

В данной работе предлагается новый алгоритм построения конечных автоматов, основанный на муравьином алгоритме [5]. Приводятся результаты экспериментального исследования эффективности разработанного метода на задаче об «Умном муравье» [6].

Постановка задачи

Конечный автомат — это шестерка $\langle S, s_0, \Sigma, \Delta, \delta, \lambda \rangle$, где S — множество состояний, $s_0 \in S$ — начальное состояние, Σ — множество входных событий, Δ — множество выходных воздействий. Функция $\delta: S \times \Sigma \rightarrow S$ называется функцией переходов, а функция $\lambda: S \times \Sigma \rightarrow \Delta$ — функцией выходов. За стартовое состояние s_0 в настоящей работе всегда принимается первое состояние. Пример диаграммы состояний конечного автомата приведен на рис. 1.

Пусть задано число состояний автомата N , множество входных событий Σ , множество выходных воздействий Δ и вещественная функция приспособленности (ФП) f , определенная на множестве всех автоматов с параметрами (N, Σ, Δ) . Значения ФП автомата пропорциональны близости его поведе-

ния к желаемому. Требуется найти автомат x такой, что значение ФП этого автомата $f(x)$ не меньше заданного барьерного значения.

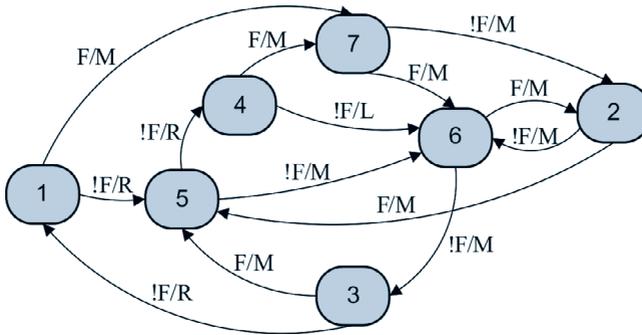


Рис. 1. Пример диаграммы состояний конечного автомата из семи состояний с $\Sigma = \{F, !F\}$, $\Delta = \{L, M, R\}$

Представление пространства поиска в виде графа

Задача поиска оптимального автомата сводится к поиску оптимальной вершины в ориентированном графе G , вершинами которого являются конечные автоматы, а ребрами — *мутации* конечных автоматов. Мутация конечного автомата – небольшое изменение в его структуре. В предлагаемом алгоритме используется два типа мутаций автоматов: для случайного перехода меняется либо состояние, в которое он ведет, либо записанное на нем выходное воздействие.

Опишем граф G , в котором производится поиск оптимальной вершины. Две вершины в полностью построенном графе G соединены ребром, если соответствующие вершинам автоматы могут быть получены друг из друга с помощью одной операции мутации. Таким образом, применяя определенные мутации, можно получить из любого автомата любой другой автомат. Кроме того, на каждом ребре (u, v) графа задается значение эвристической информации, вычисляемое по формуле:

$$\eta_{uv} = \max(\eta_{\min}, f(v) - f(u)),$$

где η_{\min} — небольшая положительная константа, например, 0,001.

Муравьиные алгоритмы

В муравьиных алгоритмах [5] решения строятся набором агентов-муравьев, называемых *муравьиной колонией*. Каждый муравей перехо-

дит по ребрам графа от вершины к вершине, пользуясь неким вероятностным алгоритмом. При этом вершины графа обычно соответствуют компонентам решения задачи, а полное решение представляется последовательностью или множеством вершин. На каждом ребре графа задаются две величины, *эвристическая информация* и *значение феромона*. Значения эвристической информации фиксируются перед началом работы алгоритма и не изменяются во время его работы. Значения феромона модифицируются муравьями в процессе работы алгоритма и реализуют долговременную память колонии. Можно выделить следующие основные этапы работы любого муравьиного алгоритма, которые повторяются, пока не будет найдено подходящее решение или не выполнится одно из условий останова.

1. **Построение решений муравьями.** Муравьи переходят от одной вершины графа к другой, руководствуясь при выборе пути эвристической информацией и значениями феромона на ребрах графа.
2. **Обновление значений феромона.** Значение феромона на каждом ребре графа может увеличиться за счет откладывания феромона муравьями или уменьшиться за счет испарения — равномерного уменьшения значений феромона.

Предлагаемый алгоритм построения конечных автоматов

Общая схема предлагаемого муравьиного алгоритма в целом совпадает с изложенной выше схемой работы классических муравьиных алгоритмов. Ключевой особенностью нового муравьиного алгоритма является то, что вершинами графа в новом алгоритме являются не компоненты решений, а полные решения — конечные автоматы.

В начале работы алгоритма строится случайный конечный автомат с заданным числом состояний. Далее, к случайному автомату применяется $(1+1)$ -эволюционная стратегия в течение небольшого числа вычислений ФП. Эволюционная стратегия хранит одно решение и выполняет одну случайную мутацию этого решения на каждом шаге, заменяя старое решение новым в случае увеличения или сохранения значения ФП. Автомат, построенный с помощью эволюционной стратегии, добавляется в граф и становится его первой вершиной.

Построение решений муравьями

Все муравьи начинают поиск из вершины, соответствующей лучшему из найденных решений. На каждом шаге муравей, находясь в вершине u , соответствующей автомату A , выполняет одно из следующих действий.

1. **Построение новых решений.** Муравей выполняет N_{mut} случайных мутаций автомата A . Результатом выполнения каждой мутации является некий автомат A_{mut} . Далее, если в графе G нет вершины t , ассоциированной с автоматом A_{mut} , такая вершина создается и добавляется в граф. Наконец, в граф добавляется ребро (u, t) . После выполнения всех мутаций муравей переходит в вершину, соответствующую лучшему из построенных на этом шаге автоматов.
2. **Выбор из существующих решений.** Следующая вершина выбирается из множества вершин N_u , инцидентных текущей вершине u . Вершина $v \in N_u$ выбирается с вероятностью:

$$p_{uv} = \frac{\tau_{uv}^\alpha \cdot \tau_{uv}^\beta}{\sum_{w \in N_u} \tau_{uw}^\alpha \cdot \tau_{uw}^\beta},$$

где $\alpha, \beta \in [0, 1]$ — параметры, определяющие значимость феромона и эвристической информации при выборе пути.

Выбор следующей вершины на каждом шаге муравья осуществляется с помощью построения новых решений с вероятностью p_{new} и путем выбора из существующих решений с вероятностью $1 - p_{\text{new}}$. Муравьи запускаются параллельно — каждый муравей делает один шаг и передает ход следующему. На одной итерации муравьиной колонии каждый муравей строит решения до тех пор, пока не исчерпает максимальное число ходов n_{stag} , на которых не произошло увеличения лучшего найденного им значения ФП. Колония может совершить не более N_{stag} итераций без увеличения лучшего значения ФП. После этого алгоритм перезапускается.

Обновление феромона

На каждом ребре графа, кроме значения феромона τ_{uv} , хранится число τ_{uv}^{\max} — наибольшее из всех значений феромона, когда-либо отложенных на ребре (u, v) . Из пути каждого муравья выделяется участок, ведущий от начала к лучшей из вершин пути, и на ребрах этого участка обновляются значения τ_{uv}^{\max} :

$$\tau_{uv}^{\max} = \max(\tau_{uv}^{\max}, f(\bar{x})),$$

где \bar{x} — вершина пути муравья с наибольшим на пути значением ФП. Затем значения феромона на каждом ребре графа обновляются по формуле:

$$\tau_{uv} = \max(\tau_{\min}, \rho \tau_{uv} + \tau_{uv}^{\max}),$$

где $\rho \in [0, 1]$ — скорость испарения феромона, а τ_{\min} — минимальное значение феромона.

Эксперименты: задача об «Умном муравье»

В задаче об «Умном муравье» [6] задано тороидальное поле размером 32×32 клетки. На поле вдоль некоторой ломаной распределены «яблоки». В данной работе рассматриваются две конфигурации задачи – поле Джона Мура и поле Санта Фе (рис. 2). Оба поля содержат по 89 клеток с яблоками. Черные клетки содержат яблоки, серые клетки обозначают пустые клетки оптимального пути, а белые клетки пусты. Муравей изначально располагается в левой верхней клетке и «смотрит» на восток. Находясь в некоторой клетке, он может определить, есть ли в следующей клетке яблоко. На каждом шаге муравей может повернуть налево, повернуть направо или перейти вперед на одну клетку. Если в клетке, в которую переходит муравей, находится яблоко, муравей его «съедает».

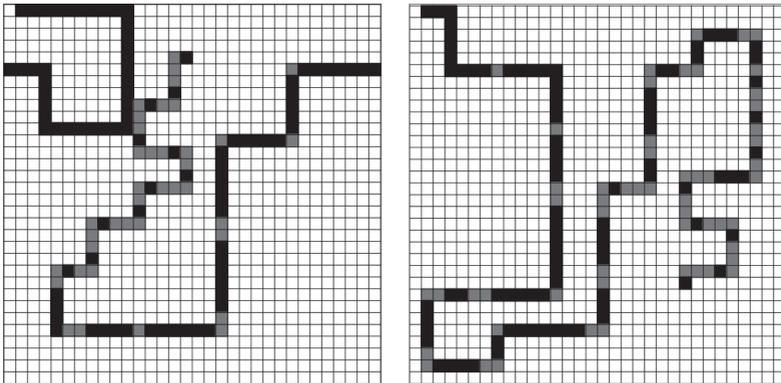


Рис. 2. Поле Джона Мура (слева) и поле Санта Фе (справа)

В описанной задаче целью является построение конечного автомата для управления муравьем, который позволит ему съесть все яблоки за отведенное число шагов s_{\max} . Используемая ФП имеет вид:

$$f = n_{\text{food}} + \frac{s_{\max} - s_{\text{last}} - 1}{s_{\max}},$$

где n_{food} — число съеденных яблок, а s_{last} — номер шага, на котором было съедено последнее яблоко. Переходы автоматов в этой задаче могут быть

помечены двумя входными событиями: F (следующая клетка содержит яблоко) и $!F$ (следующая клетка пуста), а также выходными воздействиями L (повернуть налево), R (повернуть направо) и M (сделать шаг вперед).

На рис. 3 приведены графики, позволяющие сравнить предложенный алгоритм с генетическим алгоритмом [4] для поля Джона Мура при $s_{\max} = 200$. Из рис. 3 видно, что для рассмотренной задачи предложенный алгоритм эффективнее генетического алгоритма. Более того, для построения автоматов из семи состояний, решающих поставленную задачу, предложенному алгоритму требуется в среднем 31×10^6 вычислений ФП, в то время как генетическому алгоритму требуется приблизительно в 60 раз больше — 1800×10^6 вычислений ФП. Диаграмма состояний одного из построенных автоматов изображена на рис. 1. Этот автомат позволяет муравью съесть всю еду за 189 шагов.

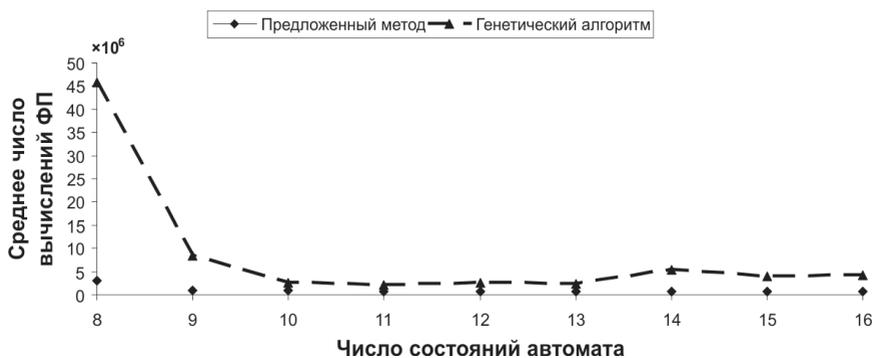


Рис. 3. Поле Джона Мура: зависимость среднего числа вычислений ФП от числа состояний автомата при $s_{\max} = 200$

Пометки об использовании переходов автомата

При разработке алгоритма и проведении экспериментальных исследований был предложен подход, не имеющий прямого отношения к муравьиному алгоритму, однако позволяющий увеличить его производительность. Предлагается при вычислении значения ФП автомата пометать переходы, которые при этом совершались. Если переход автомата не использовался при вычислении ФП, то поведение автомата при изменении такого перехода не может измениться, следовательно, значение ФП можно не вычислять заново.

На рис. 5 приведены графики зависимости среднего числа вычислений ФП от числа состояний автомата с использованием и без использования по-

меток на переходах для поля Санта Фе при $s_{\max} = 600$. Как видно из графиков, выигрыш от использования пометок увеличивается с ростом числа состояний целевого автомата. Это объясняется тем, что вероятность совершить мутацию, не меняющую поведение автомата, также увеличивается с ростом числа его состояний.

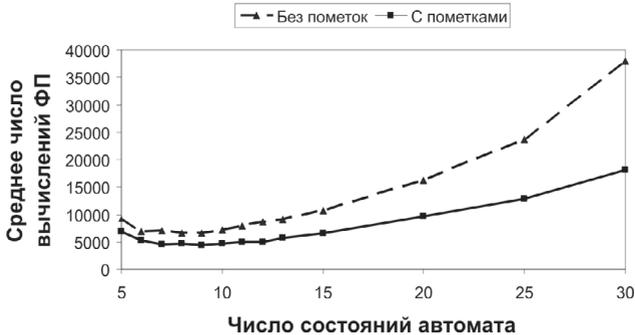


Рис. 5. Влияние пометок об использовании переходов на производительность предложенного алгоритма

Заключение

Был разработан метод построения конечных автоматов, основанный на муравьином алгоритме. Новый метод сравнивался с генетическим алгоритмом на примере задачи об «Умном муравье». В результате было установлено, что предложенному методу требуется меньше вычислений функции приспособленности для получения идеальных решений, что позволяет говорить о его высокой эффективности. Также был предложен подход к уменьшению числа вычислений ФП путем введения пометок об использовании переходов автомата при вычислении ФП. Этот подход может также быть применен для увеличения эффективности эволюционных алгоритмов построения конечных автоматов.

Литература

1. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. — 2011. — СПб: Питер. — 176 с.
2. Lucas S., Reynolds J. Learning Finite State Transducers: Evolution versus Heuristic State Merging // IEEE Transactions on Evolutionary Computation. — 2007. — Vol. 11. — № 3. — P. 308–325.
3. Chellapilla K., Czarnecki D. A preliminary investigation into evolving modular finite state machines // Proceedings of the 1999 Congress on Evolutionary Computation. — 1999. — Vol. 2. — P. 1349–1356.

4. Царев Ф.Н., Шальто А.А. О построении автоматов с минимальным числом состояний для задачи об «Умном муравье» // Сборник докладов X международной конференции по мягким вычислениям и измерениям. СПбГЭТУ «ЛЭТИ». — 2007. — Т. 2. — С. 88–91.
 5. *Dorigo M.* Optimization, Learning and Natural Algorithms. PhD thesis. Polytechnico di Milano, Italy. — 1992.
 6. *Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A.* Evolution as a theme in artificial life: The Genesys/Tracker system. Technical report. — 1990.
-

ЭВОЛЮЦИОННЫЕ СТРАТЕГИИ С АДАПТИВНЫМ ПАРАМЕТРОМ НА ОСНОВЕ СВОЙСТВ ЛАНДШАФТА ФУНКЦИИ ПРИСПОСОБЛЕННОСТИ

Чивилихин Д.С.

*магистрант кафедры компьютерных технологий НИУ ИТМО,
chivilikhin.daniil@gmail.com*

Аннотация: Предлагается метод адаптивного выбора параметра $(1, \lambda)$ -эволюционных стратегий, основанный на свойствах ландшафта функции приспособленности. Метод базируется на запоминании информации о совершавшихся в процессе работы эволюционного алгоритма мутациях. Эта информация используется для выбора значения параметра стратегии. Приводится экспериментальное сравнение адаптивной стратегии с традиционными стратегиями на задаче с сильно неровным ландшафтом функции приспособленности.

Введение

В области эволюционных вычислений вводится понятие *ландшафта функции приспособленности* (fitness landscape). В простейшем случае двух-параметрического пространства поиска ландшафт функции приспособленности (ФП) можно представить как поверхность, задающую график ФП, где абсцисса и ордината обозначают значения параметров особи, а аппликата соответствует значению ФП этой особи. Изучение свойств ландшафта ФП может проводиться как для оценки сложности решения задачи эволюционным алгоритмом, так и для настройки параметров эволюционных алгоритмов.

Эволюционные стратегии (ЭС) являются простейшим видом эволюционных алгоритмов. Адаптивные эволюционные алгоритмы используют информацию о текущем состоянии поиска для выбора значений одного или нескольких своих параметров. Примером такой информации может быть информация о ландшафте ФП. В существующих адаптивных эволюционных алгоритмах [1, 2] обычно адаптивно выбирается параметр p_m — вероятность мутировать ген особи эволюционного алгоритма. При этом, в том числе, используются свойства ландшафта ФП.

В данной работе предложен новый способ представления ландшафта ФП в процессе работы эволюционных алгоритмов и новый метод настройки $(1, \lambda)$ -эволюционных стратегий. Предлагается настраивать параметр λ , оставляя параметр p_m неизменным.

Новая адаптивная $(1, \lambda)$ -ЭС сравнивается с ЭС с фиксированным λ на примере построения конечных автоматов для задачи об «Умном муравье». Ландшафт ФП в этой задаче является сильно неровным, то есть решения с малыми значениями ФП зачастую находятся в непосредственной близости

сти от решений с большими значениями ФП. Эксперименты показали, что в большинстве случаев новая адаптивная стратегия эффективнее стратегий с фиксированным значением λ при небольших размерах целевых автоматов. Сравнение нового алгоритма с другими адаптивными эволюционными алгоритмами выходит за рамки данной работы, и будет являться частью дальнейших исследований.

(1, λ)-эволюционная стратегия

(1, λ)-эволюционная стратегия является одним из простейших видов эволюционных стратегий. Популяция (текущее множество решений) в этом алгоритме состоит из единственного решения. На каждой итерации производится λ случайных мутаций — небольших изменений — текущего решения. Далее вычисляется значение ФП всех λ измененных решений и решение с наибольшим (в случае задачи максимизации) значением ФП становится новым текущим решением.

Построение конечных автоматов для задачи об «Умном муравье»

Конечный автомат — это шестерка $\langle S, s_0 \in S, \Sigma, \Delta, \delta, \lambda \rangle$, где S — множество состояний, $s_0 \in S$ — начальное состояние, Σ — множество входных событий, Δ — множество выходных воздействий. Функция $\delta: S \times \Sigma \rightarrow S$ называется функцией переходов, а функция $\lambda: S \times \Sigma \rightarrow \Delta$ — функцией выходов.

В парадигме автоматного программирования [3] предлагается использовать конечные автоматы в качестве ключевых компонентов программных систем, задающих логику работы программы при поступлении внешних входных событий. В таких системах выделяют систему управления, представленную в виде конечного автомата, и объект управления. Одним из простейших примеров применения автоматного программирования является задача об «Умном муравье» [4]. В этой задаче требуется построить автомат, который оптимальным образом управляет некоторым агентом в игре. Игра происходит на поле, которое представляет собой тор размера 32×32 клетки. На поле вдоль некоторой строго заданной ломаной расположены «яблоки». Примеры полей изображены на рис. 1.

Агентом в игре является «муравей», который в начале игры расположен в левой верхней клетке поля и «смотрит» на восток. Муравей видит на одну клетку вперед — он может определить, есть в следующей клетке еда (событие F) или нет (событие $\neg F$). На каждом шаге муравей может повернуть налево (действие L), повернуть направо (действие R) или перейти на одну клетку вперед (действие M). Если в клетке, в которую перешел муравей, находилось яблоко, муравей его съедает. Целью игры является построение

конечного автомата, управляющий муравьем так, чтобы он мог съесть всю еду не более, чем за s_{max} шагов. Функция приспособленности в этой задаче учитывает как число съеденных яблок n_{food} , так и номер шага s_{last} , на котором было съедено последнее яблоко:

$$f = n_{food} + \frac{s_{max} - s_{last} - 1}{s_{max}}$$

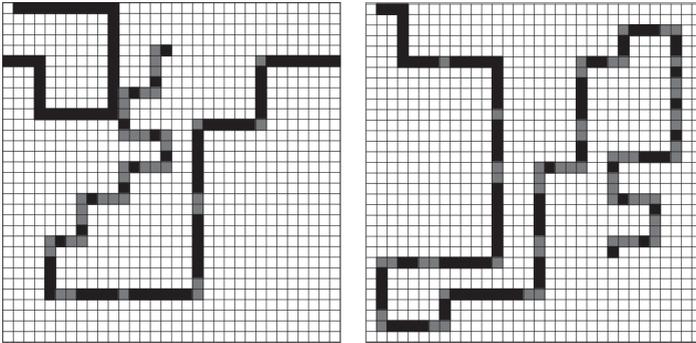


Рис. 1. Поле Джона Мура (слева) и поле Санта Фе (справа). Черные клетки обозначают яблоки, белые клетки пусты, а серые клетки изображают ломаную

Пример автомата из пяти состояний, позволяющего муравью съесть все яблоки за 394 ходов, приведен на рис. 2.

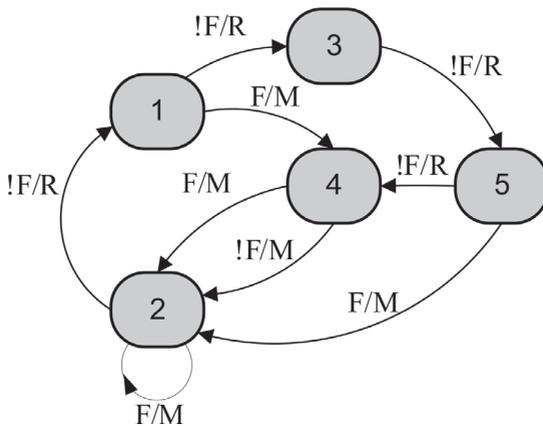


Рис. 2. Диаграмма состояний автомата из пяти состояний, позволяющего муравью съесть всю еду на поле Санта Фе за 394 шага

Мотивация: ландшафт функции приспособленности в задаче об «Умном муравье»

Мотивацией к данной работе послужил анализ ландшафта ФП для задачи об «Умном муравье». Сначала рассматривается оптимальный конечный автомат из семи состояний, решающий задачу для поля Джона Мура за 189 шагов. Значение ФП этого автомата равно 89,05. Были рассмотрены все соседи этого автомата, то есть все автоматы, полученные из него применением одной операции мутации. Гистограмма значений ФП этих автоматов изображена на рис. 3.

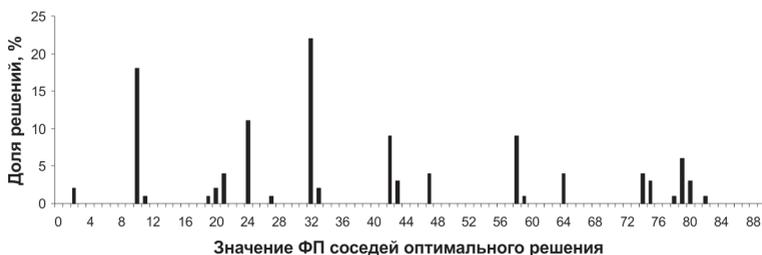


Рис. 3. Гистограмма значений ФП соседей оптимального решения задачи об «Умном муравье» для поля Джона Мура при $s_{\max} = 200$

Из гистограммы на рис. 3. видно, что ни один из соседей оптимального решения не имеет значения ФП выше, чем 82. Фактически это означает, что эволюционные алгоритмы, старающиеся максимизировать значения ФП своих особей, могут найти оптимальное решение лишь случайно.

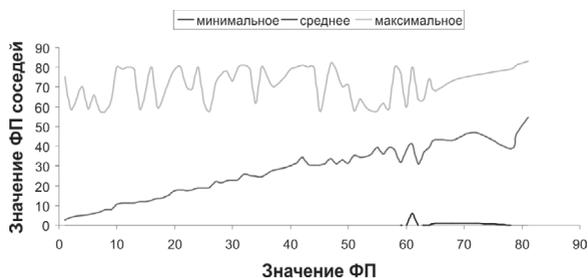


Рис. 4. Минимальное, среднее и максимальное значения ФП соседей в зависимости от значения ФП автоматов из семи состояний

Во втором эксперименте было рассмотрено 10^4 случайных автоматов из семи состояний. Для каждого из них были построены все соседние автоматы и записано значение ФП исходного автомата, а также минимальное, среднее и максимальное значение ФП соседей. График зависимости среднего,

минимального и максимального значения ФП соседей от значения ФП исходного автомата приведен на рис. 4. Эти графики показывают, что с ростом значения ФП автомата значения ФП его соседей в среднем также возрастают. Однако вместе с тем видно, что разброс значений ФП соседей для всех значений ФП автоматов велик, что негативно влияет на производительность эволюционных алгоритмов при решении данной задачи.

Предлагаемый алгоритм

Основой предлагаемого алгоритма является построение двух распределений: распределения $M_s(f_0)$ числа успешных мутаций особей по значениям ФП и распределения $M(f_0)$ числа мутаций особей по их значениям ФП. Рассмотрим состояние эволюционного алгоритма, в котором значение ФП текущего решения равно f_c , а значение ФП решения, полученного из текущего с помощью одной операции мутации, равно f_m . Если $f_m > f_c$, то значения $M_s(f_c)$ и $M(f_c)$ увеличиваются на единицу, в противном случае увеличивается только значение $M(f_c)$.

В предлагаемой адаптивной модификации $(1, \lambda)$ -ЭС значения параметра λ выбираются пропорционально отношению $M_s(f_c)$ и $M(f_c)$, где f_c — значение ФП текущего решения. На каждой итерации значение λ выбирается по формуле:

$$\lambda(M_s(f_c), M(f_c)) = \begin{cases} \frac{M_s(f_c)}{M(f_c)} \cdot N_{\text{size}}, & \text{если } M(f_c) \geq M_{\text{min}} \\ \lambda_{\text{default}}, & \text{иначе} \end{cases},$$

где N_{size} — число соседей текущего решения, M_{min} — минимальное число мутаций решения, при котором значение λ вычисляется по распределениям и λ_{default} — значение λ по умолчанию. Выбор такого правила обусловлен тем, что в случае недостатка статистической информации о мутациях решений с определенным значением ФП, использование этой статистической информации для выбора значения λ ничем не обосновано.

Результаты

Тестирование предложенного алгоритма проводилось на примере построения конечных автоматов для задачи об «Умном муравье» с полем Санта Фе. Результаты запусков адаптивной ЭС сравнивались с результатами запусков ЭС с фиксированными значениями $\lambda \in [1, 10]$. Выбор значений параметров M_{min} и λ_{default} был осуществлен перебором для задачи построения автоматов из пяти состояний с полем Санта Фе при $s_{\text{max}} = 600$. Выбранные значения параметров затем использовались во всех экспериментах.

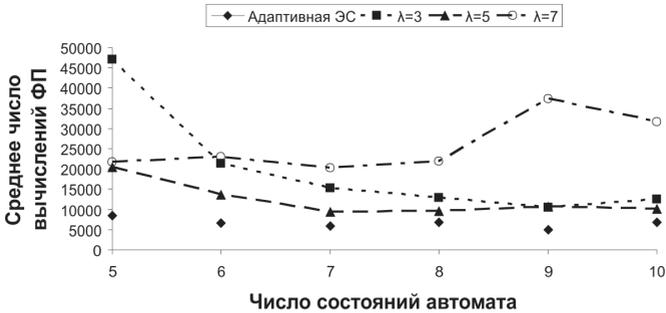


Рис. 5. Зависимость среднего числа вычислений ФП от числа состояний автомата для поля Санта Фе при $s_{\max} = 600$

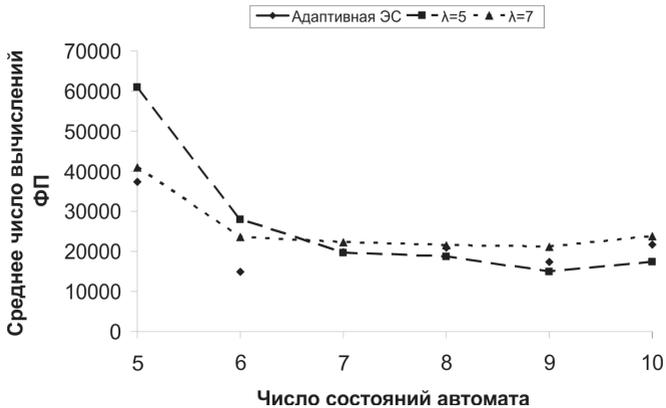


Рис. 6. Зависимость среднего числа вычислений ФП от числа состояний автомата для поля Санта Фе при $s_{\max} = 400$

В каждом из экспериментов строились автоматы с фиксированным числом состояний. Эксперимент для каждого числа состояний автомата $N \in [5, 10]$ был повторен 100 раз. Каждый эксперимент продолжался вплоть до получения идеального решения. На рис. 5 и рис. 6. представлены графики зависимости среднего числа вычислений ФП от числа состояний автомата для адаптивной ЭС и ЭС с некоторыми значениями λ . Графики для остальных значений λ не приведены, так как результаты для этих значений либо слабо отличаются от приведенных, либо существенно хуже них. Из рис. 5. видно, что в случае $s_{\max} = 600$ ЭС с адаптивным параметром позволяет найти решение в среднем быстрее, чем ЭС с фиксированным параметром. Графики на рис. 6. показывают, что при $s_{\max} = 400$ адаптивная стратегия не-

сколько эффективнее для небольших автоматов из пяти и шести состояний, а в других случаях более эффективной оказалась ЭС с $\lambda=5$.

Заключение

Предложен подход к адаптивной настройке эволюционных стратегий, основанный на основе свойств ландшафта функции приспособленности. Подход апробирован на примере адаптивной $(1, \lambda)$ -ЭС для построения автоматов в задаче об «Умном муравье». Предложенный алгоритм в большинстве случаев является более эффективным, чем ЭС с фиксированным параметром λ .

В качестве развития данной работы предполагается проведение сравнения предложенного подхода с другими методами адаптивной настройки эволюционных алгоритмов. Также будет изучена возможность оптимального выбора параметров предложенного адаптивного алгоритма.

Литература

1. *Beyer H-G, Schwefel H-P.* Evolution strategies — A comprehensive introduction // *Natural Computing: an international journal.* — 2002. — Vol. 1(1). — P. 3–52.
 2. *Govêa M.M. Jr., Araújo A.F. R.* Diversity-Based Adaptive Evolutionary Algorithms // *New Achievements in Evolutionary Computation.* — 2010. InTech. [Электронный ресурс]. — Режим доступа: <http://www.intechopen.com/books/new-achievements-in-evolutionary-computation/diversity-based-adaptive-evolutionary-algorithms>, свободный. Яз. англ. (дата обращения 22.04.2013)
 3. *Поликарпова Н.И., Шальто А.А.* Автоматное программирование. —2011. — СПб: Питер. — 176 с.
 4. *Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A.* Evolution as a theme in artificial life: The Genesys/Tracker system. Technical report. — 1990.
-

ПОСТРОЕНИЕ УПРАВЛЯЮЩИХ КОНЕЧНЫХ АВТОМАТОВ ПО СЦЕНАРИЯМ РАБОТЫ НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧИ УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ¹

Ульянцев В.И.

*магистрант кафедры «Компьютерные технологии» НИУ ИТМО,
ulyantsev@rain.ifmo.ru*

Научный руководитель —

А.А. Шалыто

*д. т. н., проф., зав. каф. «Технологии программирования» НИУ ИТМО,
shalyto@mail.ifmo.ru*

Аннотация: Настоящая работа является продолжением исследований в области поисковой программной инженерии. Предлагается алгоритм построения управляющих конечных автоматов по сценариям работы, основанный на решении задачи удовлетворения ограничений. Данный алгоритм предоставляет возможность построения автоматов, удовлетворяющих не только требованию непротиворечивости, но и требованию полноты.

Введение

Парадигма автоматного программирования [1] для реализации сущности со сложным поведением подразумевает выделение системы управления и объекта управления. На начальном этапе проектирования программы выделяются события, входные переменные и выходные воздействия. После этого проектирование программы может идти разными путями. Один из них состоит в составлении сценариев работы программы, по которым далее эвристически или автоматизировано строится управляющий автомат.

К управляющим автоматам зачастую предъявляются два требования:

- обязательное требование непротиворечивости — не должно быть двух переходов, исходящих из одного состояния управляющего автомата и одновременно выполнимых при некоторой комбинации события и входных переменных;
- необязательное требование полноты — любой комбинации события и входных переменных должен соответствовать переход в каждом состоянии.

Ранее авторами был предложен метод [2] построения автоматных программ, удовлетворяющих требованию непротиворечивости, но не удовлет-

¹ Исследование поддержано федеральной целевой программой «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы, соглашение 14.В37.21.0397.

воряющих требованию полноты. В настоящей работе предлагается метод, не обладающий данным недостатком, основанный на решении задачи удовлетворения ограничений (constraint satisfaction problem — CSP).

Постановка задачи

Управляющим конечным автоматом называется детерминированный конечный автомат, каждый переход которого помечен *событием*, последовательностью *выходных воздействий* и *охранным условием*, представляющим собой логическую формулу от *входных переменных*.

Автомат получает события от так называемых *поставщиков событий* (в их роли могут выступать внешняя среда, интерфейс пользователя и т. д.) и генерирует выходные воздействия для *объекта управления*. При поступлении события автомат выполняет тот соответствующий ему переход, для которого охранное условие оказывается истинным. При выполнении перехода генерируются выходные воздействия, которыми он помечен, и автомат переходит в соответствующее состояние. Отметим, что состояния такого автомата не делятся на допускающие и недопускающие. Пример управляющего автомата приведен на рис. 1.

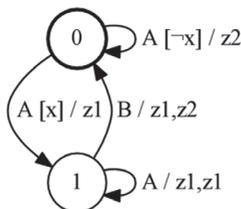


Рис. 1. Пример управляющего автомата

Для данного автомата множество входных событий равно $\{A, B\}$, охраняющие условия зависят от единственной логической входной переменной x , множество выходных воздействий равно $\{z1, z2\}$. Далее состояние автомата с номером 0 будем считать начальным.

В качестве исходных данных для построения управляющего конечного автомата используется множество *сценариев работы*. Сценарием работы будем называть последовательность $T_1 \dots T_n$ троек $T_i = \langle e_i, f_i, A_i \rangle$, где e_i — входное событие, f_i — булева формула от входных переменных, задающая охранное условие, A_i — последовательность выходных воздействий. В дальнейшем тройки T_i будем называть *элементами сценария*.

Будем говорить, что автомат, находясь в состоянии *state*, *удовлетворяет элементу сценария* T_i , если из *state* исходит переход, помеченный событием e_i , последовательностью выходных воздействий A_i и охраняющим условием,

тождественно равным f_i как булева формула. Автомат удовлетворяет сценарию работы $T_1 \dots T_n$, если он удовлетворяет каждому элементу данного сценария, находясь при этом в состояниях пути, образованного соответствующими переходами.

Решается задача построения управляющего конечного автомата, удовлетворяющего требованию полноты, с заданным числом состояний C по заданному множеству сценариев работы S_c , которым автомат должен удовлетворять.

Алгоритм построения управляющих автоматов

Предлагаемый алгоритм включает в себя пять основных этапов.

1. Построение дерева сценариев (рис. 2).
2. Построение графа совместимости вершин дерева сценариев.
3. Построение набора ограничений на целочисленные переменные, задающего требования к «раскраске» построенного графа и выражающей непротиворечивость и полноту системы переходов искомого автомата.
4. Использование стороннего пакета *Choco* [3], решающего задачу удовлетворения построенный ограничениям.
5. Построение автомата по найденной выполняющей подстановке.

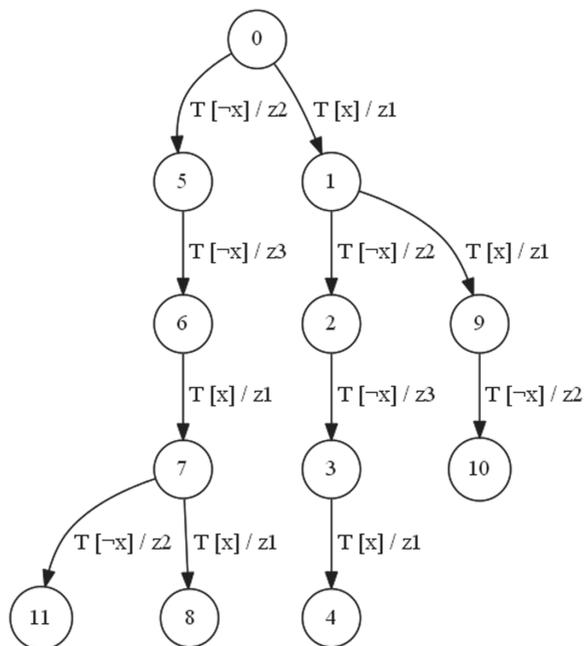


Рис. 2. Пример дерева сценариев

Опишем подробно этап построения набора ограничений, другие этапы описаны в [2]. Будем в дальнейшем множество всех условий переходов, встречающихся в сценариях работы S_c обозначать символом F . Множество входных событий e будем обозначать как E . Под F_e будем подразумевать множество условий переходов, помеченных входным событием e .

Для построения набора ограничений создадим, и будем использовать следующие целочисленные переменные.

1. Переменные x_v соответствуют цвету каждой вершины дерева сценариев v и принимают значения от 0 до $C-1$. Напомним, что вершины одного цвета будут объединены в одно состояние результирующего автомата.
2. Переменные $y_{i,e,f}$ являются вспомогательными для построения ограничений, задающих непротиворечивость (детерминированность) искомого управляющего автомата, и хранят в себе информацию о его переходах. Данные переменные являются вспомогательными, так как наличие в результирующем автомате переходов не определяется исходя из их значений. Используются переменные для каждого состояния i результирующего автомата (значение от 0 до $C-1$), каждого события e , каждого условия перехода f из множества F_e , встречающегося в сценариях. Каждая переменная принимает значения от 0 до $C-1$ и соответствует номеру состояния, в которое ведет переход искомого автомата из состояния i по событию e и условию перехода f .
3. Переменные $z_{i,e,f}$ используются для задания требования полноты искомого автомата и принимают значения 0 или 1, то есть по своей сути являются логическими. Данные переменные задаются для каждого состояния i результирующего автомата (значение от 0 до $C-1$), каждого события e , каждого условия перехода f из F_e , встречающегося в сценариях. Переменная $z_{i,e,f}$ равна 1, если существует вершина v в дереве сценариев, цвет которой равен i ($x_v = i$), и из нее ведет ребро, помеченное событием e и условием перехода f . В противном случае значение переменной равно 0. Таким образом, данные переменные хранят информацию о структуре переходов результирующего автомата, получаемого в результате объединения вершин дерева сценариев.

Составим набор ограничений на указанные переменные, задающий требования полноты и непротиворечивости искомого автомата.

1. $x_0 = 0$ — ограничение, задающее соответствие корня дерева сценариев начальному состоянию искомого автомата. В настоящем методе начальным состоянием автомата считается состояние с номером 0.
2. $x_v \neq x_u$ (для каждой несовместимой пары вершин дерева сценариев u и v , то есть соединенных ребром в графе совместимости) — ограничения, задающие непротиворечивость искомого автомата. Они гарантируют отсутствие различающих последовательностей, ведущих из одного состояния автомата. Число ограничений данного вида равно

числу ребер графа совместимости, то есть в худшем случае таких ограничений может быть $O(n^2)$, где n — число вершин дерева сценариев.

3. $(x_n = i) \Rightarrow (x_n = y_{i,e,f})$ (для каждого цвета i и каждого ребра дерева сценариев v_n , помеченного событием e и условием перехода f) — ограничения, задающие детерминированность искомого автомата. А именно, если вершине v присвоен цвет i , то цвет вершины u совпадает со значением переменной $y_{i,e,f}$, хранящей номер состояния автомата, в которое ведет переход из состояния i , помеченный событием e и условием перехода f . Число данных ограничений равно $C \cdot (n - 1)$.
4. $z_{i,e,f} = 1 \Leftrightarrow (x_{v_1} = i \vee \dots \vee x_{v_n} = i)$ (для каждого цвета i , входного события e , условия перехода f из F_e , встречающегося в заданных сценариях, и вершин $v_1 \dots v_n$ дерева сценариев, из которых ведет ребро, помеченное событием e и условием перехода f) — ограничения, необходимые для правильного задания значений переменных $z_{i,e,f}$. Количество данных ограничений оценивается как $O(C \cdot |E| \cdot |F|)$, где как $|E|$ обозначено число событий, а как $|F|$ — число различных условий перехода.

5. $\left(\sum_{f \in F_e} (z_{i,e,f} \cdot c(f)) = 0 \right) \vee \left(\sum_{f \in F_e} (z_{i,e,f} \cdot c(f)) = 2^m \right)$ (для каждого цвета i и каждого события e) — ограничения, задающие требование полноты искомого автомата. Здесь как $c(f)$ обозначена функция, которая возвращает число выполняющих подстановок для булевой формулы f . При подсчете $c(f)$ считается, что булева формула зависит от всех m переменных, содержащихся в сценариях (например, $c(\text{true}) = 2^m$, а $c(x_1 \vee \neg x_2) = 2^{m-2}$). Сумма $\sum_{f \in F_e} (z_{i,e,f} \cdot c(f))$ равна числу комбинаций значений входных переменных $values$, для которых существует переход из состояния i , помеченный событием e и условием перехода f таким, что выполняется $f(values)$. Условие полноты искомого автомата выражается тем, что или для любого значения входных переменных найдется переход, или ни для одного из значений переменных перехода не существует. Заметим, что условие того, что для любого значения входных переменных найдется переход, можно выразить как $\sum_{f \in F_e} (z_{i,e,f} \cdot c(f)) = 2^m$, так как считается, что требование непротиворечивости уже выполнено, то есть все формулы, для которых выполняется $z_{i,e,f} = 1$, попарно не имеют общих выполняющих подстановок.

Приведенные ограничения пяти типов составляют набор, задающий требования непротиворечивости и полноты искомого автомата, удовлетворяющего сценариям S_c и содержащего C управляющих состояний.

Экспериментальное исследование

Было проведено экспериментальное исследование, сравнивающее ранее разработанный и предложенный методы. Рассматривались задачи построения автоматов с 4, 6, 8 и 10 состояниями по сценариям работы различной суммарной длины. Процент автоматов, удовлетворяющих требованию полноты, построенных при помощи ранее разработанного метода, уменьшался с ростом числа состояний — для 10 состояний такие автоматы находились менее, чем в половине случаев. Предлагаемый же алгоритм гарантированно находил управляющие автоматы, удовлетворяющие требованию полноты.

Заключение

Разработан метод автоматизированного построения управляющих конечных автоматов по сценариям работы. Этот метод основан на сведении указанной задачи к задаче выполнимости ограничений. Результаты экспериментов показали наличие большого числа входных данных, для которых ранее предложенный метод находил управляющий автомат, не удовлетворяющий требованию полноты, в то время как разработанный метод справлялся с задачей, пусть и с меньшей производительностью.

Литература

1. *Поликарпова Н.И., Шалыто А.А.* Автоматное программирование. СПб: Питер, 2009.
 2. *Ulyantsev V., Tsarev F.* Extended Finite-State Machine Induction using SAT-Solver / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011, Honolulu, HI, USA, 18–21 December 2011. IEEE Computer Society, 2011. Vol. 2. P. 346–349
 3. Choco, java library for constraint satisfaction problems (CSP) and constraint programming (CP). <http://www.emn.fr/z-info/choco-solver/>
-

РАЗРАБОТКА АЛГОРИТМОВ ДЛЯ УПОРЯДОЧИВАНИЯ СТРУКТУР БЕЛКОВ

Соколов А.А.

студент НИУ ИТМО, ansokolmail@gmail.com

Буздалов М.В.

аспирант НИУ ИТМО, mbuzdalov@gmail.com

Аннотация: Данная работа посвящена моделированию движения белковых молекул. Исследование белковых взаимодействий является актуальной и малоизученной темой. Предлагается метод построения гладких конформационных движений белков.

Введение

Современная биоинформатика — молодая и бурно развивающаяся наука. Белки в клетках живых организмов выполняют множество разнообразных функций [1]. Основным предметом изучения протеомики являются белки и их взаимодействия. Одной из интереснейших задач протеомики является построение траекторий движения молекул.

Описание задачи

Для того чтобы получить представление о структуре белка, нужно использовать один из методов исследования химических объектов. Наиболее распространенные методы: спектроскопия ядерного магнитного резонанса и рентгеноструктурный анализ. В интернете доступны электронные банки белков, например, Protein Data Bank [2]. В указанном банке белков имеется 89 740 записей (по состоянию на 16 апреля 2013 г.), и их число постоянно растет.

Однако, эти отдельные снимки белка не позволяют получить полного представления о том, как молекула движется в пространстве. Возникает задача компьютерного построения траектории движения белка на основе некоторого числа конформаций, полученных из банков данных.

Метод построения графа гладких конформационных движений

Входными данными для описываемого метода является N конформаций исследуемого белка, полученных из банка белков. Назовем указанные конформации *базовыми*. Результатом работы метода является новый набор конформаций исследуемого белка и граф, описывающий, в каком порядке белок может принимать конформации из этого набора.

Метод состоит из следующих этапов:

- Вычисление конформационных движений между всеми парами конформаций и их стоимости.
- Удаление заведомо неверных конформационных движений.
- Для каждой базовой конформации: определение несовместимых конформационных движений из заданной конформации.
- Построение графа возможных конформационных движений.

Далее каждый этап будет изложен подробнее. В данном разделе считается, что базовые конформации исследуемого белка пронумерованы числами от одного до N .

Вычисление конформационных движений между всеми парами конформаций и их стоимости. Для всех пар конформаций (S, T) с номерами $1 \leq S < T \leq N$ и некоторого, заранее заданного числа K с использованием метода минимизации переноса масс [3, 4] вычисляется стоимость конформационного движения $L(S, T) = L(T, S)$ и цепочка $C(S, T)$ из $K-2$ промежуточных конформаций. Результатом работы является неориентированный граф G , вершинами которого служат базовые конформации, и между каждой парой различных вершин есть ребро, весом которого является стоимость конформационного движения между соответствующими конформациями.

В дополнение к этому, для каждой базовой конформации A выбирается один из ее твердотельных поворотов в пространстве $R(A)$, и для всех базовых конформаций $B \neq A$ строится цепочка промежуточных конформаций $F(A, B)$, эквивалентная цепочке $C(\min(A, B), \max(A, B))$ с точностью до порядка следования конформаций (от начала к концу или от конца к началу) и до твердотельного вращения конформаций. Направление этой цепочки выбирается таким, что первой конформацией является конформация A , а ее ориентация в пространстве совпадает с $R(A)$. Кроме того, все остальные конформации выравниваются относительно предыдущей путем минимизации RMSD.

Удаление заведомо неверных конформационных движений. По причине использования приближенных стохастических алгоритмов оптимизации конформационного движения может получиться так, что для некоторой тройки конформаций A, B, C выполняется неравенство $L(A, B) + L(B, C) \ll L(A, C)$. Это означает, что оптимизация конформационного движения от A до C проведена неудовлетворительным образом. Если $2 \times (L(A, B) + L(B, C)) < L(A, C)$, то ребро из A в C удаляется из графа.

Определение несовместимых конформационных движений из заданной конформации. Если белок при движении из конформации A в конформацию B проходит некоторую промежуточную конформацию C , то это движение из общих соображений должно происходить без резкого изменения направлений движения молекул, составляющих белок.

Из этого следует, что если белок пришел в конформацию C из некоторой конформации A , то он может продолжить движение не во все конфор-

мации — например, он не может вернуться обратно в конформацию A или двигаться в конформацию B , похожую на A , так как это повлечет изменение направления движений молекул на угол, близкий к 180° .

Определение несовместимых конформационных движений производится следующим образом. Пусть A — заданная базовая конформация. Для каждой базовой конформации $B \neq A$ строится список $D(A, B)$ разностей координат атомов между $R(A)$ и второй конформацией из цепочки $F(A, B)$. Тогда для атома с номером q величина $M_q(A, B) = D_q(A, B) / L(A, B)$ будет являться приближенным значением производной координат атома по времени при начале движения белка из конформации A в конформацию B . Далее, для каждой пары базовых конформаций (B, C) , $B \neq A$, $C \neq A$, $B \neq C$ вычисляется величина $Q(B, C) = \max_q (M_q(A, B) + M_q(A, C))$. Чем ближе к нулю эта величина, тем с большей долей уверенности можно утверждать, что движение из A в C может быть продолжением движения из B в A . Выберем пороговое значение ϵ и определим возможность движения из B в C через A следующим образом: $\theta(A, B, C) = 1$, если $Q(B, C) < \epsilon$, и 0 в противном случае.

Построение графа возможных конформационных движений. Строится ориентированный граф G_2 с $O(N^2)$ вершинами. В данном графе имеются вершины следующих типов:

- для каждой базовой конформации A — вершина SA , соответствующая началу движения белка из конформации A ;
- для каждой базовой конформации A — вершина TA , соответствующая окончанию движения белка в конформации A ;
- для каждой пары базовых конформаций A, B , $A \neq B$ — вершина SAB , соответствующая началу движения белка из A в B ;
- для каждой пары базовых конформаций A, B , $A \neq B$ — вершина TAB , соответствующая окончанию движения белка из A в B .

Приведем алгоритм построения множества ребер графа G_2 :

- Для всех упорядоченных пар базовых конформаций A, B , $A \neq B$:
 - добавляется ребро с весом 0 из S_A в S_{AB} ;
 - добавляется ребро с весом 0 из T_{BA} в T_A ;
 - если существует ребро E из A в B в графе G_1 , то добавляется ребро с весом, равным весу E , из S_{AB} в T_{AB} .
- Для всех упорядоченных троек базовых конформаций A, B, C , $B \neq A$, $C \neq A$, $B \neq C$:
 - $\theta(A, B, C) = 1$, то добавляется ребро с весом 0 из T_{BA} в S_{AC} .

Таким образом, в графе G_2 имеется $O(N^2)$ вершин и $O(N^3)$ ребер.

Свойством графа G_2 является то, что любой путь из S_A в T_B взаимно однозначно соответствует такому пути в графе G_1 , в котором любые два ребра подряд не являются несовместимыми, при этом веса этих путей равны. Используя граф G_2 , можно искать гладкие пути между конформациями.

Заключение

Построение возможных белковых траекторий позволяет получить представление о том, как движется белок в пространстве, и понять, какие конформации этого белка возможны. Исследование белков и их взаимодействий позволит ускорить разработку лекарственных средств.

Литература

1. *Финкельштейн А.В. Птицын О.Б.* Физика Белка. Курс лекций. 2005, 456 с.
 2. RCSB Protein Data Bank <http://www.rcsb.org>
 3. *Krass A., Nikolenko S., Stepanov E. and Porozov Yu.* Modeling of Protein Conformational Movements Based on Principle of Average Action. ISCB-ECCB 2011, Vienna, 2011, July 15–19.
 4. *Порозов Ю.Б.* Моделирование конформационной подвижности белков XIX Национальный конгресс «Человек и Лекарство 2012». Симпозиум «Биоинформатика и компьютерное конструирование лекарств». Москва, 23–27 апреля 2012.
-

ГИБРИДНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ ВЫПОЛНЕНИЯ ЗАДАНИЙ, СВЯЗАННЫХ ПОРЯДКОМ ПРЕДШЕСТВОВАНИЯ, В МНОГОПРОЦЕССОРНОЙ СРЕДЕ

Кобер Д.А.

аспирант кафедры исследования операций СПбГУ, dmitry.kober@gmail.com

Аннотация: В данной работе рассматривается подход к решению задачи составления расписания выполнения заданий, на множестве которых введен частичный порядок предшествования, основанный на применении островной модели генетического алгоритма с динамической настройкой параметров локального поиска. Помимо дочерних генетических алгоритмов основной модели, в данной работе рассматривается алгоритм искусственной иммунной системы, выступающей в роли отдельного контроллера развития островной популяции.

Введение

Планирование — процесс распределения вычислительных работ между процессорами, с целью уменьшения общего времени выполнения заданий. Важным аспектом подобных процессов является решение задач составления расписаний, которые, в общем случае, являются NP-трудными. В данной работе рассматривается задача составления расписания заданий, на множестве которых введен частичный порядок, заданный ориентированным графом без циклов. Задания представляются вершинами графа, в то время, как зависимости между заданиями — его дугами. Наличие частичного порядка существенно усложняет задачу и требует внесения множества корректив в алгоритмы построения расписания. В данной работе рассматривается гибридный алгоритм, включающий идеи как эволюционных процессов [1], так и механизмов работы искусственных иммунных систем [5].

Постановка задачи

Пусть m — число заданий, подлежащих выполнению; n — число процессоров, работающих параллельно; $\tau_{i,j}$ — время начала выполнения задания i на процессоре j , $1 \leq i \leq m$, $1 \leq j \leq n$; t_i — время выполнения задания i , $1 \leq i \leq m$. В единицу времени процессор может выполнять не более одного задания. Прерывания выполнения заданий запрещены. На множестве заданий введен частичный порядок: задание i предшествует заданию j ($i < j$) тогда и только тогда, когда $\tau_{i,k} + t_i \leq \tau_{j,l}$, где $1 \leq i, j \leq m$; $1 \leq k, l \leq n$.

Необходимо составить расписание минимальной длины.

Структура данных

Основой разработанного алгоритма является островная модель генетического алгоритма, особи которой представлены следующей структурой данных. Пусть $T: T_N \rightarrow I$ — взаимно однозначное отображение имен заданий, подлежащих выполнению (T_N), на множество индексов I , $P: P_N \rightarrow K$ — взаимно однозначное отображение множества имен процессоров (P_N) на множество индексов K . Рассматривается ассоциативный массив длины m , индексы которого принадлежат множеству I , а значения — множеству K . Исходя из приведенных выше отображений, данный массив позволяет однозначно соотнести задание и процессор, на котором это задание будет выполняться.

Для оценки приспособленности особей выбрана широко распространенная целевая функция C_{\max} , показывающая длину расписания, полученного путем восстановления расписания из хромосомы особи.

Островная модель генетического алгоритма

Генетические алгоритмы имеют вероятностный характер. Поэтому, при разных запусках, популяция может сходиться к разным допустимым решениям, которые в той или иной степени близки к оптимуму. Кроме того, нередки случаи, когда одни статически заданные параметры работы отдельного алгоритма приводят к длительной сходимости, в то время как другие позволили бы быстро получить субоптимальное, или даже оптимальное решение. Для обхода этих и многих других проблем классической схемы генетического алгоритма была разработана островная модель [2].

В данной версии гибридного алгоритма островная модель базируется на пяти параллельно развивающихся популяциях, четыре из которых основаны на «локальных» генетических алгоритмах, а одна — на алгоритме искусственной иммунной системы. Координация действий локальных алгоритмов возлагается на алгоритм-контроллер, посредством которого производится миграция особей из популяции в популяцию и контроль жизненного цикла развития популяций в целом.

Работа алгоритма-контроллера начинается с генерации случайных особей для начальных популяций. Как только все начальные популяции сгенерированы, алгоритм-контроллер запускает локальные алгоритмы и переходит в режим ожидания завершения их выполнения.

Когда все локальные алгоритмы завершили выполнение, начинается основная работа алгоритма-контроллера. Из каждой результирующей популяции выбирается третья часть лучших ее представителей (но не менее одного) посредством алгоритма селекции методом имитации отжига [3]. Выбранные особи образуют множество мигрирующих особей. Далее алгоритм-контроллер проводит по три итерации кроссовера между случайно выбранными парами особей этого множества, если его мощность больше единицы. Кроссовер вну-

три локального генетического алгоритма проводится согласно установленной стратегии развития популяции, тем самым приводя особей к одному классу. Кроссовер же внутри алгоритма-контроллера выполняет операции над особями разных классов, что приводит к получению «нестандартных комбинаций».

Завершает обработку популяций локальных алгоритмов фаза миграции, осуществляемая путем применения алгоритмов заменяющего и вероятностного обновлений.

После обновления популяций локальные алгоритмы перезапускаются, и описанный цикл повторяется.

Критериями остановки алгоритма-контроллера являются: проведение максимально возможного количества итераций и достижение максимально возможного количества неулучшающих итераций. Оба эти параметра являются предустановленными для алгоритма-контроллера.

Локальные генетические алгоритмы

Обычно, в начале работы алгоритма построения расписания в разрезе данной задачи, задания, не имеющие предков относительно графа частичного порядка, считаются независимыми, и построение расписания начинается с них. Такой метод в данной работе условно называется методом «сверху-вниз». Построенное этим методом расписание выглядит так:

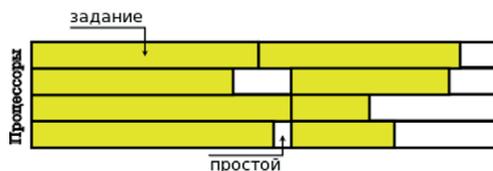


Рис. 1. Расписание, полученное методом «сверху-вниз»

Два из четырех локальных генетических алгоритма развития популяций используют именно этот метод.

Другой метод — «снизу-вверх» — строит расписание, начиная с листовых вершин графа частичного порядка. Для этого все зависимости обращаются, образуя новый граф частичного порядка, относительно которого и работает механизм построения расписания. Расписание, полученное таким способом, имеет в некотором смысле обратный вид:

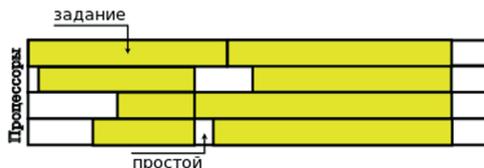


Рис. 2. Расписание, полученное методом «снизу-вверх»

Этот метод используют другие два локальных генетических алгоритма.

Таким образом, комбинируя особей популяций, развивающихся по этим стратегиям, делается попытка получить более «компактную укладку» заданий в расписании.

Важной особенностью локальных генетических алгоритмов, применяемых в данной работе, является наличие фазы локального поиска с динамической настройкой параметров, где под параметрами понимаются коэффициенты усиления и расширения поиска [4], базирующиеся на введенной ранее структуре данных, описывающей особь. Введение данной фазы позволило получать более приспособленных особей в пределах каждой отдельной итерации генетического алгоритма.

Пятый алгоритм развития популяции особей основан на идеях искусственной иммунной системы.

Искусственная иммунная система

Искусственные иммунные системы (ИИС) – это адаптивные алгоритмы и системы алгоритмов, работа которых основана на идеях, взятых из наблюдений за принципами функционирования естественных иммунных систем [5]. При гибридизации ИИС с другими алгоритмами, чаще всего рассматривается лишь часть всего процесса работы естественной иммунной системы, наиболее подходящая под специфику конкретной задачи. В данной работе используются такие модели ИИС, как отрицательный отбор, отвечающий за регуляцию жизненного цикла популяции, и клональная селекция, выполняющая функции алгоритма развития популяции и механизма локального поиска [6].

Иммунная система развития популяции

Для запуска алгоритма иммунной системы необходима начальная популяция, которая, как правило, генерируется случайным образом. Работу иммунной системы можно условно разделить на два состояния: пассивное и активное. В пассивном состоянии иммунная система «просматривает» решения с целью обнаружения *патогенов* — решений, значения целевой функции которых превышает среднее значение целевой функции по популяции. При выявлении патогенов алгоритм иммунной системы переключается в активную фазу, где основным алгоритмом является клональная селекция [6]. В целом, клональная селекция похожа на генетический алгоритм без кроссовера и предельно высокой вероятностью мутации, но есть существенные отличия: генетический алгоритм имеет тенденцию к концентрации особей популяции в окрестности наилучшего, возможно, локального оптимума, в то время, как клональный алгоритм позволяет получать разнообразное множество локальных решений, в том числе и глобальный опти-

мум. Кроме того, важной отличительной особенностью механизма работы клонального алгоритма является использование значения аффинности антитела при подсчете степени мутации, которую нужно применить к конкретному антителу для улучшения значения целевой функции. При обнаружении семейства патогенов решения, им соответствующие, клонируются, и клоны подвергаются мутации, причем степень мутации md зависит от величины невязки значения целевой функции каждого отдельного антитела и средней оценки популяции:

$$md = \frac{f_c - f_p}{f_w - f_p} \cdot \frac{n}{3},$$

где f_w — значение целевой функции «наиболее активного» патогена из рассматриваемого множества, n — размерность антитела, f_p — среднее значение целевой функции по популяции, f_c — аффинность данного антитела. После мутации из полученного множества клонов выбираются наилучшие относительно аффинности. Для каждого изначального патогена, находящегося в популяции, ищется мутировавший клон, значение аффинности которого выше текущего. Если такой клон найден — происходит замещение. В противном случае патоген остается в популяции¹.

Иммунный локальный поиск

Для увеличения средней аффинности популяции применяется алгоритм иммунного локального поиска, который является в некотором смысле обратным к алгоритму, описанному выше. Если центральной частью предыдущего алгоритма являлись процессы обнаружения антител, имеющих наименьшую аффинность, и взаимодействия с ними, в данном алгоритме делается упор на поиск наиболее приспособленных антител. Из популяции выбирается группа решений, имеющих наилучшие значения целевой функции относительно всей популяции, и производится процедура клональной селекции, применяемая к каждому антителу этой группы. Кроме того, после создания популяции мутировавших клонов, заменяются не «родительские» антитела, а патогены, имеющие аффинность ниже, чем у представителей множества мутировавших клонов.

Заключение

Результаты работы описанного гибридного алгоритма сравнивались с результатами работы жадного алгоритма (алгоритм критического пути),

¹ По аналогии с процессами, протекающими в живом организме, иммунная система не всегда справляется с вирусом, в результате чего организм заболевает.

списочного алгоритма (дающего оптимальное решение при наличии двух процессоров) и метода ветвей и границ построения расписаний. Кроме того, для тестирования алгоритма на задачах исключительно большого размера, где выявление оптимума посредством метода ветвей и границ является практически невыполнимой операцией, был разработан модуль генерации задач с заранее заданной длиной расписаний. Во всех тестовых случаях решения, полученные разработанным алгоритмом, были оптимальными или близкими к оптимуму, а использование параллельной модели позволило значительно ускорить выполнение алгоритма. Так же были выявлены ограничения алгоритма иммунной системы, что дало развитие новой ветви исследования, направленной на построение многоуровневой параллельной модели.

Литература

1. *И.Л. Каширина*. Введение в эволюционное моделирование // Федеральное агентство по образованию Воронежский государственный университет, Воронеж. — 2007.
 2. *Darrell Whitley, Soraya Rana, and Robert B. Heckendorn*. The island model genetic algorithm: On separability, population size and convergence // *Journal of Computing and Information Technology*. — 7:33–47, 1998.
 3. *Rakesh Kumar and Jyotishree*. Effect of annealing selection operators in genetic algorithms on benchmark test functions // *International Journal of Computer Applications*. — (0975–8887), 40 – No.3, 2012.
 4. *Guang-Yuan Liu, Yi He, Yonghui Fang, and Yuhui Qiu*. A novel adaptive search strategy of intensification and diversification in tabu search // *Neural Networks and Signal Processing*. — Proceedings of the 2003 International Conference on volume 1, pages 428–431 Vol.1, 2003.
 5. *Steven A. Hofmeyr and S. Forrest*. Architecture for an artificial immune system. — 2000.
 6. *В.И. Литвиненко*. Применение алгоритма клонального отбора для решения систем алгебраических уравнений // *Математичні машини і системи*. — 2006.
-

ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ ПРИСПОСОБЛЕННОСТИ ДЛЯ ТЕСТИРОВАНИЯ РЕШЕНИЙ ОЛИМПИАДНЫХ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ

Буздалова А.С.

*студентка кафедры компьютерных технологий НИУ ИТМО,
abuzdalova@gmail.com*

Буздалов М.В.

*ассистент кафедры компьютерных технологий НИУ ИТМО,
mbuzdalov@gmail.com*

Аннотация: Рассматривается задача автоматической генерации тестов для олимпиадных задач по программированию с помощью эволюционных алгоритмов. Для повышения эффективности генерации предлагается использовать вспомогательные функции приспособленности. Рассматриваются различные способы их использования, такие как многокритериальная оптимизация со случайным выбором критериев и ранее разработанный метод EA+RL, применимый как к однокритериальным, так и к многокритериальным алгоритмам. Результаты экспериментов подтверждают, что использование вспомогательных функций приспособленности позволяет получать достаточно качественные тесты за меньшее число поколений эволюционного алгоритма.

Введение

Данная работа посвящена генерации тестов, применяющихся для оценки решений задач по олимпиадному программированию [1, 2]. Как правило, в состав задания по олимпиадному программированию входит формулировка задачи, которую необходимо решить, и форматы входных и выходных данных. Тесты представляют собой наборы входных данных, для которых известны соответствующие выходные данные — правильные ответы на поставленную задачу. Для того чтобы решение успешно прошло тест, необходимо, чтобы оно позволило получить правильный ответ в пределах максимального установленного времени и фиксированного объема используемой памяти.

Целью данной работы является автоматическая генерация тестов производительности. Будем называть *успешным* запуском алгоритма генерации запуск, в результате которого получен тест, время работы решения на котором превышает максимальное установленное время.

Для генерации тестов используются эволюционные алгоритмы [3]. Этот подход был предложен в работе одного из авторов данной статьи [4]. Задачей эволюционного алгоритма является максимизация *целевой*

функции приспособленности (в данном случае времени работы решения олимпиадной задачи). Алгоритм оперирует с *особями* — кандидатами в решение задачи оптимизации (определенным образом закодированными тестами). Особи подвергаются *мутации* и *скрещиванию*. Таким образом, образуются потомки. Наиболее приспособленные особи из числа потомков и родителей отбираются для формирования следующего *поколения*. Описанный процесс повторяется до тех пор, пока не будет получена достаточно приспособленная особь или же не будет превышено максимальное число поколений.

Несмотря на то, что время работы решения является целевой функцией приспособленности, использование ее в эволюционном алгоритме не является эффективным подходом по ряду причин: время работы решения платформозависимо, зашумлено и квантовано [4]. Для повышения эффективности эволюционного алгоритма предлагается использовать вспомогательные функции приспособленности [5–8], коррелирующие с целевой функцией, но при этом лишенные ее недостатков. Рассматривается как случайный выбор вспомогательных функций приспособленности, предложенный в работе [8], так и выбор наиболее эффективных функций приспособленности, осуществляемый с помощью метода EA+RL, предложенного ранее авторами данной работы [5, 6].

Цель работы

Целью работы является исследование эффективности различных подходов к генерации тестов, основанных на использовании эволюционных алгоритмов. Особое внимание уделяется применению вспомогательных функций приспособленности. Предлагается генерировать тесты производительности для двух примеров решений олимпиадной задачи. В следующих разделах описаны олимпиадная задача, ее решения, а также соответствующие решениям вспомогательные функции приспособленности.

Олимпиадная задача по программированию

В качестве примера, рассмотрим задачу «Ships. Version 2», расположенную на сайте Timus Online Judge [9] под номером 1394. Задача формулируется следующим образом: даны N кораблей, каждый длины s_i , и M гаваней, каждая длины h_j . Необходимо распределить корабли по гаваням таким образом, чтобы суммарная длина всех кораблей, размещенных в j -ой гавани, не превосходила значения h_j . Гарантируется, что корректное размещение всегда существует. Вводятся следующие ограничения:

- $N \leq 99, 2 \leq M \leq 9, 1 \leq s_i \leq 100$;
- $\sum s_i = \sum h_j$;

- ограничение на время работы решения 1 с;
- ограничение на объем используемой памяти 64 Мб.

Описанная задача является частным случаем задачи о мультирюкзаке, которая, в свою очередь, является NP-трудной в строгом смысле [10] (другими словами, не известно ни одного решения этой задачи, время работы которого было бы полиномиальным относительно любого набора входных значений). По этой причине является крайне маловероятным, что каждый возможный экземпляр задачи может быть решен в рамках заданных ограничений на время и память. Однако в случае наиболее удачных решений весьма сложно составить такой тест, на котором решение превысит заданный предел времени работы.

Вспомогательные функции приспособленности

Как было сказано выше, целевой функцией приспособленности является время работы решения на тесте, однако выращивать особи, руководствуясь значениями этой функции, неэффективно. Предлагается использовать вспомогательные функции приспособленности. Для вычисления значений этих функций в код решения встраиваются счетчики, коррелирующие со временем выполнения кода. Например, если в алгоритме решения часто вызывается некоторая рекурсивная функция, можно ввести счетчик числа вызовов этой функции. Значения, принимаемые таким счетчиком, не зависят от платформы, на которой запущено решение, а также не подвержены зашумлению. Ниже представлены псевдокоды двух решений задачи, описанной в предыдущем разделе, с включенными в них счетчиками. Далее будем называть вспомогательные функции приспособленности так же, как соответствующие счетчики.

В *Решении 1* используются счетчики I, P, R:

```
Read the input data
I := 0, P := 0, R := 0
while(solution not found)
  Randomly shuffle ships
  Call the recursive DP-based ship arranging procedure
    For each call to this procedure, R := R + 1
    In each innermost loop, P := P + 1
  if (solution is found)
    Write the answer
  else
    I := I + 1
  end if
end while.
```

В *Решении 2* используются счетчики Iterations, Length, Tuple:

```
Read the input data
iterations := 0, length := 0, last := 0
while (solution not found) do
  Randomly shuffle ships and havens
  last := 0
  Call the recursive ship arranging procedure
  For each call to this procedure, last:= last + 1
  if (solution is found) then
    Write the answer
  else
    iterations := iterations + 1
    length := length + last
    last := 0
  end if
end while
tuple := 100000000 * iterations + last.
```

Главное отличие между двумя решениями заключается в реализации рекурсивной функции распределения кораблей. Особенностью первого решения является то, что для реализации этой функции используется подход, основанный на динамическом программировании. К сожалению, привести более детальное описание не представляется возможным, так как правила олимпиадных соревнований по программированию не позволяют публиковать решения участников. Однако из результатов экспериментов будет видно, что, с точки зрения сложности генерации тестов, решения различаются.

Описание предлагаемого подхода

В данном разделе описаны различные подходы к генерации тестов, основанные на применении эволюционных алгоритмов, которые могут использоваться как для максимизации одной функции приспособленности (*однокритериальные*), так и для максимизации нескольких функций приспособленности (*многокритериальные*). Функция приспособленности может использоваться постоянно (*фиксированная*) или выбираться из заранее подготовленного набора в течение работы эволюционного алгоритма (*динамическая*). Применяемые в работе эволюционные алгоритмы, а также стратегии выбора динамической функции приспособленности.

Эволюционные алгоритмы

Используются как однокритериальные, так и многокритериальные эволюционные алгоритмы. Во всех используемых алгоритмах особи кодируются

ся одинаковым образом, а также применяются одни и те же операторы мутации и скрещивания, описанные в работе [11]. Размер поколений составляет 200 особей. Ниже описаны используемые алгоритмы.

Однокритериальный генетический алгоритм (GA). В данном алгоритме для формирования следующего поколения применяется турнирный отбор с размером турнира 2 и с вероятностью выбора лучшей особи, равной 0,9. Операторы мутации и скрещивания применяются с вероятностью 1,0. Используется элитизм, отбирается пять лучших особей [3]. Если на протяжении 10 000 поколений лучшее полученное значение функции приспособленности не изменяется, то текущее поколение очищается и заполняется случайными новыми особями.

Многокритериальный генетический алгоритм (NSGA-II). Для одновременной оптимизации нескольких функций приспособленности используется ускоренная вариация алгоритма NSGA-II [12]. Применяется стратегия выбора особей, основанная на доминировании по Парето.

Стратегии выбора вспомогательных функций приспособленности

Стратегия, предложенная М.Т. Jensen [8]. Рассмотрим две стратегии выбора вспомогательных функций приспособленности. Первая из них была предложена в работе [8]. Согласно этой стратегии, вспомогательная функция приспособленности выбирается случайным образом и используется в течение фиксированного числа поколений. Затем выбирается следующая функция приспособленности, и так далее, пока все вспомогательные функции приспособленности не будут использованы.

Метод EA+RL. В качестве второй стратегии выбора использовался предложенный авторами данной статьи метод EA+RL [5, 6]. В контексте данной работы он называется GA+RL и NSGA-II+RL, в соответствии с теми эволюционными алгоритмами, совместно с которыми был применен. Функция приспособленности выбирается с помощью обучения с подкреплением [13] из множества, включающего как целевую функцию приспособленности, так и вспомогательные. На выбор влияет вознаграждение, которое зависит от роста целевой функции приспособленности. Целью алгоритма обучения с подкреплением является максимизация суммарного вознаграждения. Таким образом, выбираются функции приспособленности, способствующие росту целевой функции приспособленности.

В качестве алгоритма обучения используется алгоритм *Delayed Q-learning* $m=0$, $\varepsilon=0.001$, $\gamma=0.1$ [14]. Алгоритм перезапускается каждые 50 поколений с целью предотвращения стагнации.

Результаты эксперимента

Для каждого из рассмотренных решений были сгенерированы тесты. Использовались все рассмотренные алгоритмы с каждой из предложенных

функций. Каждый алгоритм был запущен 100 раз, затем результаты были усреднены. Алгоритм останавливался, если был получен тест, на котором время выполнения решения превышало ограничение по времени, или же если было достигнуто максимальное число поколений, равное 10 000.

Результаты экспериментов приведены в Таблицах 1, 2. Используются следующие обозначения: ФП — функция приспособленности, используемая для оценки особей; T — целевая функция приспособленности (время выполнения решения на тесте); σ — среднеквадратичное отклонение числа поколений, необходимого для генерации эффективного теста. Алгоритм тем эффективнее, чем меньше среднее число понадобившихся для генерации поколений.

Рассмотрим результаты генерации тестов для Решения 1, приведенные в Таблице 1. Как и ожидалось, использование времени выполнения решения в качестве единственной функции приспособленности (алгоритм №4) неэффективно.

В случае фиксированной ФП, многокритериальная оптимизация (№5–7) превосходит по производительности однокритериальную (№1–4) вне зависимости от используемой вспомогательной функции приспособленности. В случае динамической ФП многокритериальная оптимизация также достаточно эффективна, хотя и уступает производительности многокритериального алгоритма в случае фиксированной ФП. Однако использование динамической ФП в общем случае может быть более предпочтительным подходом, так как заранее неизвестно, какая из фиксированных ФП эффективна. Пример случая, когда не все вспомогательные ФП эффективны, соответствует Решению 2.

№	Алгоритм	ФП	Успешные запуски, %	Число поколений	
				среднее	σ
Фиксированная ФП					
1	GA	I	99	2999	1986
2	GA	R	93	3153	3742
3	GA	P	54	12621	12770
4	GA	T	0	—	—
5	NSGA-II	T, I	100	203	119
6	NSGA-II	T, R	100	440	381
7	NSGA-II	T, P	100	448	360
Динамически выбираемая ФП					
8	GA+RL	все	65	9636	9538
9	NSGA-II+RL	все	99	895	1215
10	NSGA-II+Jensen	все	100	882	786

Таблица 1. Результаты генерации тестов для Решения 1

Результаты генерации тестов для Решения 2 приведены в Таблице 2. В случае фиксированной ФП, только Turtle оказывается эффективной вспо-

могательной функцией (№ 1–5). Это невозможно узнать, не запустив алгоритмы с каждой из вспомогательных функций.

В то же время, все приведенные методы, использующие динамический выбор ФП, достаточно эффективны (№ 8–10). В случае использования динамической ФП достаточно одного запуска любого из приведенных алгоритмов. Механизм автоматического выбора наиболее эффективной функции приспособленности позволяет не осуществлять этот выбор вручную. Таким образом, использование динамически выбираемой вспомогательной функции приспособленности является достаточно универсальным и эффективным методом.

№	Алгоритм	ФП	Успешные запуски, %	Поколения	
				среднее	σ
Фиксированная ФП					
1	GA	Tuple	95	3815	3466
2	GA	Iterations	54	12669	12873
3	GA	Length	51	13755	14082
4	GA	T	0	-	-
5	NSGA-II	T, Tuple	95	2217	3136
6	NSGA-II	T, Iterations	45	15861	16723
7	NSGA-II	T, Length	20	41330	44768
Динамически выбираемая ФП					
8	GA+RL	все	80	5817	6160
9	NSGA-II+RL	все	72	6679	7764
10	NSGA-II+Jensen	все	75	6103	7076

Таблица 2. Результаты генерации тестов для Решения 2

Заключение

Было проведено сравнение некоторых подходов к генерации тестов для задач по олимпиадному программированию. Показано, что использование вспомогательных функций приспособленности позволяет существенно повысить производительность эволюционных алгоритмов, применяемых для решения рассмотренной задачи генерации. Наиболее универсальной следует считать стратегию динамического выбора вспомогательной функции приспособленности, так как для ее использования не требуется предварительных знаний от том, какая из вспомогательных функций приспособленности оптимальна. Результаты эксперимента показывают, что в среднем наиболее эффективно использовать данную стратегию совместно с алгоритмами многокритериальной оптимизации. Дальнейшие исследования предполагают разработку метода автоматического добавления в код тестируемого решения счетчиков, позволяющих вычислять значения вспомогательных функций приспособленности.

Литература

1. ACM International Collegiate Programming Contest. http://en.wikipedia.org/wiki/ACM_ICPC [дата просмотра: 23.04.2013].
 2. International Olympiad in Informatics. <http://www.ioinformatics.org> [дата просмотра: 23.04.2013].
 3. *Eiben A.E., Smith J.E.* Introduction to Evolutionary Computing. — Springer-Verlag, Berlin, Heidelberg, New York. — 2007.
 4. *Буздалов М.В.* Генерация тестов для олимпиадных задач по программированию с использованием генетических алгоритмов // Научно-технический вестник СПбГУ ИТМО. — 2011. — № 2(72). — С. 72–77.
 5. *Buzdalova A., Buzdalov M.* Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement Learning // Proceedings of the Eleventh International Conference on Machine Learning and Applications, ICMLA 2012. — Boca Raton: IEEE Computer Society, 2012. — Vol. 1. — P. 150–155.
 6. *Буздалова А.С., Буздалов М.В.* Метод повышения эффективности эволюционных алгоритмов с помощью обучения с подкреплением // Научно-технический вестник информационных технологий, механики и оптики — Санкт-Петербург, 2012. — № 5(81) — С. 115–119.
 7. *Knowles J.D., Watson R.A., Corne D.* Reducing Local Optima in Single-Objective Problems by Multi-objectivization // Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization EMO '01. — London, UK: Springer-Verlag. — 2001. — P. 269–283.
 8. *Jensen M.T.* Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation: Evolutionary Computation Combinatorial Optimization // Journal of Mathematical Modelling and Algorithms. — 2004. — № 4. — P. 323–347.
 9. Timus Online Judge. Архив задач с проверяющей системой. <http://acm.timus.ru> [дата просмотра 23.04.2013]
 10. *Pisinger D.* Algorithms for Knapsack Problems: PhD. thesis / University of Copenhagen. — 1995.
 11. *Buzdalova A.S., Buzdalov M.V.* Adaptive Selection of Helper-Objectives for Test Case Generation // Proceedings of the IEEE Congress on Evolutionary Computation. — 2013 [to be published].
 12. *R.G. L. D'Souza, K.C. Sekaran, and A. Kandasamy.* Improved NSGA-II based on a novel ranking scheme. Computing Research Repository, abs/1002.4005. — 2010.
 13. *Sutton R.S., Barto A.G.* Reinforcement Learning: An Introduction. — MIT Press, Cambridge, MA, 1998. — 322 p.
 14. *Strehl A.L., Li L., Wiewora E., Langford J., Littman M.L.* PAC Model-Free Reinforcement Learning // Proceedings of the 23rd International Conference on Machine learning, ICML'06. — 2006. — P. 881–888.
-

ПРИМЕНЕНИЕ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ О ВЫПОЛНИМОСТИ КВАНТИФИЦИРОВАННОЙ БУЛЕВОЙ ФУНКЦИИ ДЛЯ ПОСТРОЕНИЯ УПРАВЛЯЮЩИХ КОНЕЧНЫХ АВТОМАТОВ ПО СЦЕНАРИЯМ РАБОТЫ И ТЕМПОРАЛЬНЫМ СВОЙСТВАМ

Е.В. Панченко, В.И. Ульянов

Научный руководитель — кандидат наук, ассистент

Ф.Н. Царев

Аннотация: Данная статья рассматривает вопрос о применении управляющих автоматов при построении систем со сложным поведением. Предложен метод построения управляющего конечного автомата по заданному множеству сценариев работы и темпоральным свойствам, которые должны выполняться в результирующем автомате. Метод основан на сведении задачи к задаче выполнимости квантифицированной булевой функции. Описан алгоритм построения данной функции и основные составляющие полученной булевой формулы.

Ключевые слова: кванторы, булева функция, управляющие конечные автоматы, верификация.

Введение

Парадигма автоматного программирования используется для реализации систем со сложным поведением во многих промышленных отраслях [1]. В настоящее время существуют системы, насчитывающие более тысячи различных состояний, при записи их поведения в виде конечного автомата. Одним из используемых подходов, при попытке задать параметры системы на вход сторонним приложениям является передача сценариев работы системы, по которым эвристически строится необходимый конечный автомат.

Как правило, данный метод позволяет эффективно создавать автоматизированные системы по сценариям работы, описываемым, например, в функциональной спецификации продукта. Тем не менее, не все свойства системы можно описать с помощью сценариев работы.

В данной работе приведено решение, которое позволяет строить автоматы не только по сценариям работы, но также и по темпоральным свойствам, заданным с помощью подмножества языка LTL, с помощью сведения решения данной задачи к решению задачи выполнимости обобщенной булевой формулы.

Цель работы

В работе [2] было спроектировано решение, позволяющее решить задачу построения автомата по сценариям работы с помощью сведения ее к за-

даче SAT, задаче о выполнимости булевой формулы. После этого SAT-solver находил решение данной формулы, и на основании полученного примера алгоритм строил необходимый автомат.

Полученный автомат удовлетворял заданным сценариям работы, однако зачастую не соответствовал изначальным бизнес-требованиям, поскольку лист сценариев оказывался не полон. Целью настоящей работы является модификация метода построения управляющих автоматов, а именно добавление возможности использования LTL-формул в качестве дополнительного инструмента описания требуемой работы автомата.

Базовые положения исследования

На вход разрабатываемой программе подается список сценариев, а также набор темпоральных свойств работы системы, заданный с помощью синтаксиса языка линейной темпоральной логики LTL.

Сценарием работы является последовательность $T_1 \dots T_n$ троек $T_i = \langle e_i, f_i, A_i \rangle$, где e_i — входное событие, f_i — булева формула от входных переменных, задающая охранное условие, A_i — последовательность выходных воздействий. Автомат, находясь в состоянии S , удовлетворяет элементу сценария T_i , если из S исходит переход, помеченный событием e_i , последовательностью выходных воздействий A_i и охранным условием, тождественно равным f_i как булева формула. Автомат удовлетворяет сценарию работы $T_1 \dots T_n$, если он удовлетворяет каждому элементу данного сценария, находясь при этом в состояниях пути, образованного соответствующими переходами.

На формат входных LTL наложено ограничение в виде невозможности использовать характеристики состояний автомата, поскольку на этапе задания логики, состояний еще нет. Однако, это позволяет задавать свойства общего формата, что может быть полезно при создании систем с нуля. Было принято решение использовать в качестве переменных LTL формул входные и выходные воздействия. Таким образом, синтаксис входных данных включает в себя:

- Булевы связки ($!$, \wedge , \vee);
- Темпоральные операторы **X** (neXt) и **U** (Until);
- Выведенные темпоральные операторы **F**(Future), **G**(Globally in the future), **R**(Release);
- Предикаты:
 - $wasEvent(e)$ — переход совершен по событию e
 - $wasAction(z)$ — во время перехода было вызвано выходное воздействие z .

Пример входного темпорального свойства, используемого при верификации часов с будильником в работе [3]:

$$\mathbf{F}(!wasEvent(e) \wedge !wasAction(z)).$$

Разработанное программное средство в несколько шагов производит построение искомого конечного автомата, удовлетворяющего входным данным:

1. С помощью разработанного в работе [2] алгоритма по заданным сценариям работы создается булева формула, содержащая логические переменные $y_{a,b,e,f}$ (для каждой пары состояний результирующего автомата a и b , каждого события e , каждой формулы f , встречающейся в сценариях), соответствующие наличию перехода из состояния a в состояние b , помеченного событием e и формулой f в результирующем автомате.
2. С использованием подхода верификации моделей с ограничением на длину вычислений, Bounded Model Checking [3], входные темпоральные свойства «разворачиваются» в булеву функцию. Для этого используется понятие «обратного цикла» и ограничение его глубины поиска.

«Обратный цикл» — это цикл, образованный ребром, ведущим из какого-либо состояния пути в состояние, лежащее на данном пути ранее, как показано на Рис. 1.

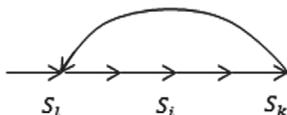


Рис. 1. Пример обратного (k, l) -цикла

Логика линейного предполагает, что некоторое утверждение будет выполняться для всех путей. Поэтому в работе [4] доказательство построено от противного, проверяя существование пути, на котором выполняется отрицание LTL формулы.

Однако, данный подход возможен только при верификации уже построенных автоматов. В нашем случае автомата еще только должен быть построен, поэтому используется квантор всеобщности, позволяя найти такое решение формулы, значения переменных $y_{a,b,e,f}$, чтобы темпоральные свойства выполнялись на всех путях построенного конечного автомата.

3. Для ограничения времени работы алгоритма устанавливается ограничение на длину цикла k . Таким образом, темпоральные свойства с помощью разложения на композицию темпоральных и булевых предикатов «разворачиваются» по циклу в формулу, размер которой линейно зависит от размера начальной формулы и константы k .
4. Полученные формулы объединяются в одну квантифицированную булеву функцию, проверяющую все бесконечные и конечные пути в радиусе k и содержащую кванторы существования и всеобщности.

5. Для решения полученной формулы используется программное средство *DepQBF*[5]. Полученные значения входных переменных с квантором существования используются для построения искомого управляющего конечного автомата.

Заключение

Разработан алгоритм автоматизированного построения управляющих автоматов по сценариям работы и темпоральным свойствам, основанный на сведении данных задач к проблеме о разрешимости квантифицированной булевой формулы. В настоящий момент производится сравнение качества и скорости работы с методами, реализованными в работах [3], [6].

Литература

1. *Поликарпова Н.И., Шальто А.А.* Автоматное программирование. СПб: Питер, 2009
2. *Ulyantsev V., Tsarev F.* Extended Finite-State Machine Induction using SAT-Solver / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011, Honolulu, HI, USA // IEEE Computer Society, 2011. — V. 2. — P. 346–349.
3. *Егоров К.В., Шальто А.А.* Совместное применение генетического программирования и верификация моделей для построения автоматов управления системами со сложным поведением
4. *Biere A., Cimatti A., Clarke E.M., Zhu Y., Strichman O.* Bounded Model Checking
5. *DepQBF QBF-solver.* <http://lonsing.github.io/depqbf/>
6. *Walkinshaw N., Bogdanov K.* Inferring Finite-State Models with Temporal Constraints

АЛГОРИТМ ВЫРАВНИВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДНК ДЛЯ МОДЕЛИ MAPREDUCE

Славнейшев Ф.В.

*магистрант кафедры КТ факультета ИТуп НИУ ИТМО,
slavnejshev@mail.ru*

Аннотация: Выравнивание последовательностей ДНК является важной и одновременно вычислительно сложной задачей биоинформатики. В данной работе представляется алгоритм выравнивания последовательностей ДНК для популярной модели распределенных вычислений MapReduce. Также проводится анализ уже существующих решений для данной модели.

Введение

Выравнивание последовательностей ДНК является одной из важнейших задач в биоинформатике. Решение этой задачи требуется при филогенетическом анализе, определении функций отдельных частей ДНК на основе сравнения, нахождения эволюционно консервативных участков.

Однако решение задачи выравнивания последовательной ДНК требует больших вычислительных ресурсов. Например, в случае ДНК человека и при длине чтений в 100 символов производится выравнивание как минимум 600 миллионов чтений с допущением трех несовпадений относительно образца ДНК длиной три миллиарда символов [1]. Решение этой задачи на обычном персональном компьютере может потребовать несколько недель или даже месяцев вычислений. Применение распределенных вычислений и использование кластеров машин позволяет сильно сократить требуемое для выравнивания время.

Модель MapReduce

Одной из самых популярных моделей распределенных вычислений является MapReduce. Использование данной модели позволяет решать задачи, связанные с большими объемами данных. Еще одним плюсом данной модели является ее хорошая масштабируемость.

Работа MapReduce состоит из двух шагов:

1. **map-шаг** — производится обработка входных данных и формирование промежуточных результатов, которые объединяются по некоторому ключу (идентификатору решаемой задачи) в группы;
2. **reduce-шаг** — происходит обработка каждой группы промежуточных результатов и формируется окончательный ответ для решаемой задачи.

Из описания модели видно, что необходимым условием ее эффективной работы является разбиение решаемой задачи на множество небольших подзадач. В случае выравнивания последовательной ДНК данное условие можно легко удовлетворить, т. к. выравнивание каждого чтения фактически не зависит от других. Информация об образце может быть предоставлена каждому узлу в сети целиком, либо образец может быть также разбит на части.

Обзор имеющихся решений

Рассмотрим наиболее известные алгоритмы выравнивания последовательностей ДНК для модели MapReduce:

1. CloudBurst [2] — простой и быстрый при работе с небольшими объемами данных алгоритм, основанный на методе «seed and extend». Выполняется всего один прогон работы MapReduce, в ходе которого находятся общие k -меры (части последовательности ДНК длиной символов), а затем производится выравнивание чтения в каждой найденной позиции. Алгоритм подходит для работы с короткими чтениями (36 пар оснований) и небольшим по длине образцом. Для более длинных чтений время работы и требуемая память жесткого диска сильно возрастают;
2. BlastReduce [3] — распределенная версия алгоритма BLAST. Главным отличием данного алгоритма от CloudBurst является процесс объединения общих k -меров для уменьшения числа выполняемых выравниваний. Выполняется три прогона MapReduce. Однако BlastReduce имеет примерно такие же ограничения на длину чтений и образца, что и CloudBurst;
3. Crossbow [4] — алгоритм нахождения однонуклеотидных полиморфизмов (SNP), на первом этапе которого выполняется выравнивание при помощи алгоритма Bowtie. Алгоритм Bowtie использует для выравнивания чтений созданный в результате препроцессинга образца BWT. На каждом узле запускается отдельная копия Bowtie, а в память узла загружается BWT, который может занимать несколько гигабайт оперативной памяти, что повышает требования к ЭВМ, на которых может быть запущен данный алгоритм. Еще одним минусом алгоритма является невозможность учета ошибок вставки и удаления, т. к. текущая версия Bowtie может работать лишь с ошибками замены.

Описание алгоритма

Основная идея алгоритма заключается в том, чтобы разбить каждое чтение на k -меры и определить их позиции в образце, т. е. найти общие k -меры. Затем по расположению отдельных частей можно найти наиболее вероят-

ную позицию исходного чтения относительно образца. После нахождения приблизительного положения чтения в образце можно произвести выравнивание оставшихся частей чтения при помощи существующих алгоритмов. Такой подход позволяет произвести выравнивание чтений, содержащих многочисленные несоответствия с образцом.

Рассмотрим более подробно, как при помощи модели MapReduce можно достичь требуемых результатов. Генерацию и нахождение позиций k -меров можно сделать за один прогон MapReduce. На map-шаге каждое чтение разбивается некоторым образом на k -меры, которые выдаются вместе с указанием позиции в чтении и идентификатора чтения в качестве промежуточных результатов. Стоит отметить, что позицию k -мера задает не только смещение от начала чтения до начала k -мера, но и указание того, из прямого или обратно-комплементарного варианта чтения взят этот k -мер. K -мер используется в качестве ключа. При этом на часть mapper'ов подается разбитый на части образец, который обрабатывается аналогичным образом. В итоге каждый reducer получает на вход k -мер и множество позиций, в которых он встречается. Если k -мер не встречается в чтениях или в образце, то reducer пропускает такой k -мер. В противном случае reducer в качестве ответа выдает множество позиций k -мера. Стоит отметить, что в отличие от алгоритмов CloudBurst и BlastReduce вместе с k -мерами не выводятся сами чтения и части образца. При больших длинах чтений это позволяет избежать генерации огромных объемов побочных данных.

На следующем этапе определяется положение каждого чтения по позициям его частей. На map-шаге обрабатываемое множество позиций разбивается на два — множество позиций в чтениях и множество позиций в образце. Затем для каждого встретившегося ID чтения выводится множество позиций в образце, дополненное позицией в чтении. В качестве ключа используется ID чтения. На reduce-шаге для каждого чтения имеется некоторое множество позиций в нем, на которых лежат общие k -меры. При этом для каждого такого k -мера может быть несколько вариантов расположения в образце. Каждый k -мер покрывает некоторую часть образца. Задача reducer'a — объединить все покрытия в группы. Для этого для каждого покрытия вычисляется позиция, в которой находилось бы чтение, если бы все чтение до начала k -мера полностью совпадало с образцом. Производится объединение близких по этому параметру покрытий «жадным» алгоритмом.

На заключительном этапе работы алгоритма производится окончательное выравнивание чтений. Каждое имеющееся покрытие дополнено чтением. Для того, чтобы произвести выравнивание, нужно иметь ту часть образца, в которой по предположению находится чтение. На map-шаге образец разбивается на блоки, каждый из которых выводится с ключом, равным номеру блока. Все имеющиеся сгруппированные покрытия выводятся с ключом, сформированным по тому же правилу. На reduce-шаге имеется блок из образца и множество чтений с покрытиями, для каждого из которых произ-

водится выравнивание непокрытых частей чтения. Для этого используется алгоритм Нидлмана-Вунша. Если расстояние Левенштейна между чтением и образцом превышает указанный в параметрах алгоритма максимум, выравнивание отбрасывается. Если требуется для каждого чтения найти максимум одно выравнивание, возможен еще один дополнительный этап, на котором для каждого чтения находится лучшее выравнивание.

Текущие результаты

Алгоритм имеет определенные ограничения в использовании. При слишком малом значении длины k -меров на втором этапе генерируется огромное число покрытий, которые могут потребовать больших затрат памяти жесткого диска. При этом на значение длины налагается ограничение параметром (максимальное расстояние Левенштейна между чтением и образцом) следующего вида:

$$k \leq \frac{Len_{read}}{d+1},$$

где Len_{read} — длина чтения. Таким образом, для некоторых значений параметра d и при определенных длинах чтения и образца выравнивание данным алгоритмом осуществить затруднительно.

Производился тестовый запуск алгоритма на машине с четырехъядерным процессором Core i5 2.80 GHz и четырьмя гигабайтами оперативной памяти. В качестве образца использовалось ДНК *Escherichia coli*, размер которой примерно четыре миллиона пар оснований. Производилось выравнивание четырех миллионов чтений длиной 36 пар оснований каждое. Для каждого чтения находилось лучшее выравнивание, то есть производился и дополнительный четвертый этап. На каждом этапе запускалось два `mapper`'а и один `reducer`.

В результате удалось произвести выравнивание 95,431 % чтений за 8 минут при $d=1$ и $k=18$. При $d=2$ и $k=12$ удалось произвести выравнивание 97,512 % чтений менее чем за 15 минут. Планируется произвести сравнение с уже имеющимися алгоритмами выравнивания как при работе с короткими, так и с длинными чтениями.

Заключение

В данной работе были рассмотрены существующие распределенные алгоритмы выравнивания последовательностей ДНК для модели MapReduce. Также был представлен трехэтапный алгоритм выравнивания последовательностей ДНК, при построении которого были учтены проблемы уже имеющихся алгоритмов. Разработанный алгоритм был протестирован на реальных данных и показал высокую скорость выравнивания чтений.

Литература

1. *Liu C.-M., Lam T.-W., Wong T., Wu E., Yiu S.-M., Li Z., Luo R., Wang B., Yu C., Chu X., Zhao K., Li R.* Soap3: GPU-Based Compressed Indexing and Ultra-Fast Parallel Alignment of Short Reads. Third Workshop Massive Data Algorithmics, June 2011.
 2. *Schatz M.C.* CloudBurst: highly sensitive read mapping with MapReduce. [Электронный ресурс]. Режим доступа <http://bioinformatics.oxfordjournals.org/content/25/11/1363.full.pdf> свободный. Яз. англ. (дата обращения 19.04.13).
 3. *Schatz M.C.* BlastReduce: High Performance Short Read Mapping with MapReduce. [Электронный ресурс]. Режим доступа <http://www.cs.umd.edu/Grad/scholarlypapers/papers/MichaelSchatz.pdf> свободный. Яз. англ. (дата обращения 19.04.13).
 4. *Langmead B., Schatz M.C., Lin J., Pop M., Salzberg S.L.* Searching for SNPs with cloud computing. [Электронный ресурс]. Режим доступа <http://genomebiology.com/content/pdf/gb-2009-10-11-r134.pdf> свободный. Яз. англ. (дата обращения 19.04.13).
-

ОЦЕНКА ВРЕМЕНИ РАБОТЫ ЭВОЛЮЦИОННОГО АЛГОРИТМА RMHC ПОД УПРАВЛЕНИЕМ АЛГОРИТМА Q-LEARNING НА ЗАДАЧЕ ONEMAX С МЕШАЮЩИМ КРИТЕРИЕМ ОПТИМИЗАЦИИ

Буздалов М.В.

*ассистент кафедры компьютерных технологий НИУ ИТМО,
mbuzdalov@gmail.com*

Буздалова А.С.

*студентка кафедры компьютерных технологий НИУ ИТМО,
abuzdalova@gmail.com*

Аннотация: В данной работе доказывается, что эволюционный алгоритм с двумя критериями оптимизации — целевым и мешающим, управляемый обучением с подкреплением, использующим жадную стратегию исследования среды, находит оптимальное решение столь же эффективно (в асимптотическом смысле), что и эволюционный алгоритм, оптимизирующий только целевой критерий.

Введение

Метод EA+RL [1, 2], разработанный авторами настоящей статьи, успешно применяется в практических задачах. Метод основан на обучении с подкреплением [3] и предназначен для повышения эффективности эволюционных алгоритмов [4]. Было экспериментально подтверждено, что с помощью указанного метода можно эффективно решить ряд модельных задач, однако до настоящего времени для метода не было разработано теоретического обоснования. В данной работе описаны первые результаты по этому направлению.

Модельная задача оптимизации

Исследуется модификация модельной задачи оптимизации, известная как OneMax [4] и широко применяемая в теоретических исследованиях эволюционных алгоритмов. Пространство оптимизации в этой задаче — множество векторов длиной N , состоящее из нулей и единиц. Оптимизируемая величина (целевой критерий оптимизации) — число единиц в векторе. Над вектором определена операция мутации, инвертирующая один случайно выбранный бит.

Данная задача модифицируется путем добавления дополнительного критерия оптимизации — числа нулей в векторе. Целевой и дополнительный критерии оптимизации противоречат друг другу. Цель метода EA+RL

— выбирать такой критерий, чтобы при оптимизации по нему была достигнута цель оптимизации – максимизация целевого критерия. С точки зрения метода неизвестно, какой критерий «оптимальнее», так как на практике целевой критерий не всегда является оптимальным [1, 2].

Эволюционный алгоритм

В качестве алгоритма оптимизации для указанной модельной задачи используется алгоритм Random Mutation Hill Climbing, управляемый обучением с подкреплением. В качестве алгоритма обучения с подкреплением выбран алгоритм Q-Learning с жадной стратегией исследования среды [3]. Состоянием обучения с подкреплением является число единиц в текущей особи.

Схема алгоритма оптимизации такова:

- Инициализация:
 - X — текущая особь, изначально равна вектору из N нулей;
 - Q — матрица оценки выгодности переходов размером $N \times 2$, изначально заполнена нулями;
 - F_0 — функция приспособленности, равная числу нулей в векторе;
 - F_1 — функция приспособленности, равная числу единиц в векторе.
 - $M(X)$ — оператор мутации (возвращает вектор, равный вектору X с одним инвертированным битом);
 - $\alpha \in (0; 1)$, $\gamma \in (0; 1)$ — параметры алгоритма.
- Итерация:
 - $S = F_1(X)$;
 - Если $S = N$, заканчиваем работу;
 - $Y = M(X)$;
 - $X_0 = X$
 - Если $Q(S, 0) = Q(S, 1)$ то $I =$ равновероятно 0 или 1;
 - Если $Q(S, 0) > Q(S, 1)$, то $I = 0$, иначе $I = 1$;
 - Если $I = 0$, то $F = F_0$, иначе $F = F_1$;
 - Если $F(Y) \geq F(X)$, то $X = Y$;
 - $R = F_1(X) - F_1(X_0)$
 - $Q(S, I) = Q(S, I) \cdot (1 - \alpha) + \alpha \cdot (R + \gamma \cdot \max(Q(S, 0), Q(S, 1)))$

Лемма об обучении

Сформулируем следующую лемму:

Лемма об обучении. Пусть алгоритм впервые посетил состояние S , а затем вышел из него в какое-либо другое состояние. Тогда при всех следующих посещениях состояния S обучение с подкреплением выберет функцию F_1 .

Доказательство. При первом посещении состояния S выполняется равенство $Q(S, 0) = Q(S, 1) = 0$. Следовательно, равновероятно может быть вы-

брана как функция F_0 , так и функция F_1 . В зависимости от того, чему равнялся мутировавший бит — нулю или единице, алгоритм может как остаться в состоянии S , так и перейти в состояние $S+1$ или $S-1$. Рассмотрим эти случаи в зависимости от нового состояния:

1. $S, R=0$. Максимум из $Q(S,0)$, $Q(S,1)$ также равен нулю, поэтому значения Q изменены не будут, и этот случай не влияет на дальнейший выбор обучения с подкреплением.
2. $S+1, R=1$. Следовательно, была выбрана функция F_1 . Тогда $Q(S,1)$ будет равен $\alpha \cdot R$, что является положительной величиной, и $Q(S,1) > Q(S,0)$.
3. $S-1, R=-1$. Следовательно, была выбрана функция F_0 . Тогда $Q(S,0)$ будет равен $\alpha \cdot R$, что является отрицательной величиной, и $Q(S,1) > Q(S,0)$.

Таким образом, при первом выходе из состояния S будет выполняться $Q(S,1) > Q(S,0)$. При последующих переходах из состояния S всегда будет выбираться функция F_1 , причем будет выполняться неравенство $R \geq 0$. Если $Q(S,1) > 0$ до обновления, то оно будет положительным и после обновления. Если $Q(S,1) = 0$, то оно может стать положительным или остаться нулевым, но в этом случае $Q(S,0) < 0$. Таким образом, неравенство $Q(S,1) > Q(S,0)$ будет всегда выполняться, что приведет к тому, что алгоритм будет всегда выбирать функцию F_1 , **что и требовалось доказать**.

Моделирование алгоритма цепью Маркова

Для вычисления матожидания числа итераций алгоритма до достижения оптимума можно воспользоваться цепью Маркова. Перейдем к описанию цепи, моделирующей рассматриваемый алгоритм. Для различения состояния обучения с подкреплением в моделируемом алгоритме и состояния цепи Маркова последние будем называть «вершинами».

Первая группа вершин цепи A_i , $0 \leq i \leq N$ соответствует тому, что алгоритм пришел в состояние $S=i$ в первый раз. В этом случае алгоритм еще не знает, какой критерий следует выбрать. Однако, когда алгоритм покинет это состояние, он уже будет знать, что делать в случае $S=i$, согласно лемме об обучении.

Вторая группа вершин цепи B_i , $0 \leq i \leq N-2$ соответствует тому, что алгоритм вышел из вершины A_{i+1} , воспользовавшись функцией F_0 . Так как достичь состояния $i+1$ алгоритм может, только однажды побывав в состоянии i , то в этом случае алгоритм будет всегда выбирать функцию F_1 . При этом он может либо остаться в состоянии i , либо перейти в состояние $i+1$.

При попадании в состояние с номером $i+1$ из вершины B_i алгоритм также будет всегда выбирать функцию F_1 , так как он уже заведомо побывал в состоянии $i+1$. Для таких ситуаций требуется завести третью группу вершин цепи C_i , $1 \leq i \leq N-1$.

Опишем переходы между указанными вершинами.

1. Из вершины A_0 имеется переход по вероятности $\frac{1}{2}$ в саму себя и по вероятности $\frac{1}{2}$ в вершину A_1 . Действительно, любая мутация особи, состоящей из нулей, приводит к превращению нуля в единицу. С вероятностью $\frac{1}{2}$ будет выбрана функция F_0 , которая отклонит эту мутацию, в противном случае будет выбрана функция F_1 , которая эту мутацию примет.
2. Из вершины A_i , $1 \leq i \leq N-1$, имеется три перехода. С вероятностью $(N-i)/(2N)$ одновременно будет выбрана функция F_1 и будет инвертирован бит со значением 0, таким образом, состоится переход в A_{i+1} . С вероятностью $i/(2N)$ одновременно будет выбрана функция F_0 и будет инвертирован бит со значением 1, таким образом, состоится переход в B_{i-1} . Во всех остальных случаях, которые составляют вероятность $\frac{1}{2}$, алгоритм останется в вершине A_i .
3. Из вершины B_i , $0 \leq i \leq N-2$, имеется два перехода. С вероятностью $(N-i)/N$ будет инвертирован бит со значением 0 и состоится переход в вершину C_{i+1} . С оставшейся вероятностью алгоритм останется в вершине B_i .
4. Из вершины C_i , $1 \leq i \leq N-1$, также имеется два перехода — с вероятностью $(N-i)/N$ в вершину A_{i+1} , с оставшейся вероятностью в себя.

Сведение к линейной взвешенной цепи Маркова

Заметим, что вершины цепи Маркова из групп B и C имеют по одному входящему переходу и одному выходящему переходу, если не рассматривать переходы-петли. Кроме того, они принимают участие только в цепочке вершин $A_i \rightarrow B_{i-1} \rightarrow C_i \rightarrow A_{i+1}$. Целью построения исходной цепи Маркова является подсчет матожидания числа переходов из состояния A_0 в состояние A_N . Следовательно, мы можем упростить построенную цепь Маркова, исключив вершины из групп B и C , но взамен каждому переходу, помимо вероятности его осуществления, потребуется сопоставить его длину.

Длина перехода $A_i \rightarrow B_{i-1}$ равняется единице. Из вершины B_{i-1} можно выйти с вероятностью $(N-i+1)/N$, следовательно, матожидание числа итераций и, следовательно, длины перехода $B_{i-1} \rightarrow C_i$ будет равно $N/(N-i+1)$. Аналогично, длина перехода $C_i \rightarrow A_{i+1}$ будет равна $N/(N-i)$. Общая длина перехода $A_i \rightarrow A_{i+1}$ равна $1 + N/(N-i+1) + N/(N-i)$.

В полученной упрощенной цепи Маркова переходы ведут либо из состояния A_i в себя, либо из A_i в A_{i+1} . Таким образом, структура цепи стала линейной, что существенно упрощает расчет матожидания.

Опишем переходы в упрощенной цепи. Из вершины A_0 , как и ранее, имеется два перехода: один в себя с вероятностью $\frac{1}{2}$ и длиной 1, второй в A_1 с вероятностью $\frac{1}{2}$ и длиной 1. Из вершины A_i , $1 \leq i \leq N-1$, имеется три перехода:

- в вершину A_i с вероятностью $\frac{1}{2}$ и длиной 1;
- в вершину A_{i+1} с вероятностью $(N-i)/(2N)$ и длиной 1;
- в вершину A_{i+1} с вероятностью $i/(2N)$ и длиной $1 + N/(N-i+1) + N/(N-i)$.

Определим функцию $Z(i)$ как матожидание числа шагов, требуемых для того, чтобы перейти из состояния A_i в состояние A_{i+1} . Для $Z(0)$ верно соотношение $Z(0) = \frac{1}{2}(1 + Z(0)) + \frac{1}{2}$, следовательно, $Z(0) = 2$. Для $x > 0$ выполняется соотношение:

$$Z(i) = \frac{1}{2}(1 + Z(i)) + (N-i)/(2N) + i(1 + N/(N-i+1) + N/(N-i))/(2N),$$

что эквивалентно $Z(i) = 2 + i/(N-i+1) + i/(N-i)$.

Общее число итераций, таким образом, равняется

$$T_R(N) = \sum_{i=0}^{N-1} \left(2 + \frac{i}{N-i+1} + \frac{i}{N-i} \right). \quad (1)$$

Асимптотическая оценка

Для работы простого эволюционного алгоритма, Random Mutation Hill Climbing, на задаче OneMax (без введения дополнительного, мешающего критерия оптимизации) существует следующая оценка матожидания числа итераций до нахождения оптимального ответа:

$$T_0(N) = \sum_{i=0}^{N-1} \frac{N}{N-i} = \sum_{i=0}^{N-1} \left(1 + \frac{i}{N-i} \right) = \Theta(N \log N). \quad (2)$$

Отметим, что для каждого слагаемого из формулы (1) можно определить верхнюю и нижнюю асимптотические оценки с использованием соответствующего слагаемого из формулы (2):

$$1 + \frac{i}{N-i} < 2 + \frac{i}{N-i+1} + \frac{i}{N-i} < 2 \cdot \left(1 + \frac{i}{N-i} \right).$$

Отсюда следует, что $T_0(N) < T_R(N) < 2 \cdot T_0(N)$.

Заключение

В работе показано, что асимптотика числа итераций до завершения работы эволюционного алгоритма Random Mutation Hill Climbing под управлением обучения с подкреплением, использующего жадную стратегию,

на модельной задаче OneMax с мешающим критерием равна $\Theta(N \log N)$, а число итераций $T_R(N)$ не превышает $2 \cdot T_0(N)$, где $T_0(N)$ — число итераций того же эволюционного алгоритма без обучения с подкреплением на модельной задаче без мешающего критерия оптимизации.

Литература

1. *Buzdalova A., Buzdalov M.* Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement Learning // Proceedings of the Eleventh International Conference on Machine Learning and Applications, ICMLA 2012. — Boca Raton: IEEE Computer Society, 2012. — Vol. 1. — P. 150-155.
 2. *Буздалова А.С., Буздалов М.В.* Метод повышения эффективности эволюционных алгоритмов с помощью обучения с подкреплением // Научно-технический вестник информационных технологий, механики и оптики — Санкт-Петербург, 2012. — № 5(81) — С. 115–119.
 3. *Sutton R.S., Barto A.G.* Reinforcement Learning: An Introduction. — MIT Press, Cambridge, MA, 1998. — 322 p.
 4. *Eiben A.E., Smith J.E.* Introduction to Evolutionary Computing. — Springer-Verlag, Berlin, Heidelberg, New York, 2007. — 304 p.
-

ГЕНЕРАЦИЯ ТЕСТОВ ДЛЯ ОЛИМПИАДНЫХ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ С ПОМОЩЬЮ МНОГОКРИТЕРИАЛЬНЫХ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

Якорев В.О.

*студент кафедры компьютерных технологий НИУ ИТМО,
vadimyakorev@gmail.com*

Буздалов М.В.

*ассистент кафедры компьютерных технологий НИУ ИТМО,
mbuzdalov@gmail.com*

Аннотация: В данной работе рассматривается задача генерации тестов для олимпиадных задач по программированию эволюционных алгоритмов. Для повышения эффективности генерации предлагается использовать многокритериальные эволюционные алгоритмы. Рассматриваются различные алгоритмы, предлагаются методы адаптации алгоритмов под поставленную задачу. Результаты экспериментов подтверждают, что использование многокритериальных эволюционных алгоритмов позволяет получать достаточно качественные тесты за меньшее число поколений эволюционного алгоритма.

Введение

В данной работе рассматривается применение эволюционных методов многокритериальной оптимизации [1, 2] для построения тестов, необходимых для оценки корректности решений олимпиадных задач по программированию. В работе [3] описан метод генерации тестов для олимпиадных задач с помощью эволюционных, при этом в данной работе использовались однокритериальные методы оптимизации. В ходе данной работы проводится анализ эффективности применения эволюционных методов многокритериальной оптимизации совместно с этим способом. Также проводилась адаптация существующих алгоритмов для данной задачи с целью повышения эффективности.

Цель работы

Целью работы является применение многокритериальных эволюционных алгоритмов к генерации тестов для олимпиадной задачи «Ships. Version 2» [3]. Ожидается, что рассмотренные методы позволят повысить

эффективность построения тестов. Эффективность построения теста с помощью эволюционного алгоритма принимается, как число поколений, необходимое для генерации теста, на котором время работы тестируемого решения олимпиадной задачи превышает заданный временной промежуток. Также планируется проанализировать возможность оптимизации по всем параметрам одновременно, что позволит избежать выбора функции приспособленности.

Описание предлагаемого подхода

Для однокритериальной оптимизации задачи генерации тестов используется целевая функция t — время работы тестирующего решения [4]. Предлагается использовать не одну функцию, а вектора целевых функций, и проводить оптимизацию сразу по всем функциям из вектора. В данные вектора могут входить следующие функции:

- число итераций алгоритма решения i ;
- суммарное число вызовов рекурсивной функции на каждой итерации r ;
- упорядоченная пара $p = (i, r^{\wedge})$, где r^{\wedge} — число вызовов рекурсивной функции на последней итерации.

Описание эксперимента

В ходе эксперимента измерялась эффективность работы как обычных генетических алгоритмов, использующих различные функции приспособленности, так и многокритериальных эволюционных алгоритмов. Каждый алгоритм запускался по 50 раз. Вычислялись среднее значение и отклонение по количеству итерацию алгоритмов.

В качестве генетических операторов, использовались аналогичные операторам, описанным в [3]. В качестве многокритериальных алгоритмов применялись алгоритмы: Lexicographic ordering (с турнирным отбором и архивированием) [1], NSGA2 (с турнирным отбором и архивированием) [1]. Размер особи составлял 200, размер турнира равен единице.

Результаты

Результаты проведенного эксперимента представлены в Таблице 1. Из результатов видно, что однокритериальная оптимизация по целевой функции приспособленности t является неэффективной, в то время как многокритериальная оптимизация, приводит к более быстрому получению результата.

Алгоритм	Функция приспособленности	Число поколений	
		Среднее	Отклонение
GA	iterations + last iteration length	3112	3256
GA	iterations	10536	10873
GA	time	—	—
NSGA-II	time, iterations + last iteration length	1339	1382
NSGA-II	time + iterations	2987	2271
NSGA-II	all	1627	1365
Lexicographic ordering	time, iterations + last iteration length	1585	1505

Таблица 1. Результаты эксперимента

Заключение

Результаты показывают, что методы многокритериальной оптимизации позволяют значительно повысить эффективность генерации тестов для рассмотренной олимпиадной задачи, по сравнению с показателями, достигаемыми при использовании традиционной функции приспособленности. Также высокие результаты при оптимизации по все параметрам одновременно позволяют избежать необходимости выбора функции приспособленности.

Литература

1. *Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen.* Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, 2007.
2. *Kaisa Miettinen.* Nonlinear Multiobjective Optimization. Springer, 1999.
3. *Буздалов М.В.* Генерация тестов для олимпиадных задач по программированию с использованием генетических алгоритмов // Научно-технический вестник СПбГУ ИТМО. — 2011. — №2. (72) — С. 72–77.
4. *Eiben A.E., Smith J.E.* Introduction to Evolutionary Computing. — Springer-Verlag, Berlin, Heidelberg, New York, 2007. — 304 p.
5. *Strehl A.L., Li L., Wiewora E., Langford J., Littman M.L.* PAC Model-Free Reinforcement Learning // Proceedings of the 23rd International Conference on Machine Learning. ICML'06. — 2006. — P. 881–888.

СИНТЕЗ СИСТЕМ УПРАВЛЕНИЯ БАЛАНСИРУЮЩИМ РОБОТОМ НА ОСНОВЕ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Якорев В.О.

*студент кафедры компьютерных технологий НИУ ИТМО,
vadimyakorev@gmail.com*

Шальто А.А.

*д. т. н., профессор, заведующий кафедрой технологии программирования
НИУ ИТМО, anatoly.shalyto@gmail.com*

Аннотация: В данной работе рассматривается задача синтеза систем управления двухколесным балансирующим роботом с помощью методов искусственного интеллекта. Построены системы управления, основанные на конечных автоматах, построенных с помощью метода отжига, системы управления, основанные на ПИД-регуляторах, построенных с помощью генетических алгоритмов и метода отжига. Проведено сравнение полученных систем управления. Результаты экспериментов подтверждают, что методы параметрического и структурного синтеза систем управления достаточно перспективны в плане их применения для построения систем управления мобильными роботами.

Введение

В данной работе рассматривается проблема балансировки двухколесного балансирующего робота [1]. Как правило, для такого типа роботов в роли системы управления используется ПИД-регулятор [1, 2, 3]. При этом тема использования эволюционных алгоритмов для параметрического синтеза систем управления [4] подобных роботов остается малоизученной. Также малоизучена тема использования конечных автоматов [5] для управления подобным классом мобильных роботов.

Цель работы

Целями данной работы являются:

- построение систем управления мобильным роботом на основе автоматического синтеза конечных автоматов с помощью метода отжига [6];
- построение систем управления мобильным роботом на основе автоматического параметрического синтеза системы управления (ПИД-регулятор) с использованием генетических алгоритмов и метода отжига;

- анализ применимости и эффективности использования данных методов для построения систем управления мобильным роботом данного типа.

Описание предлагаемого подхода

Для построения систем управления, основанных на ПИД-регуляторах, используются генетический алгоритм и метода отжига для параметризации ПИД-регулятора. При этом каждое решение представляет из себя набор параметров ПИД-регулятора.

Для построения систем управления, основанных на конечных автоматах, используется метод отжига для структурного синтеза автомата. При этом каждое решение представляет из себя конечный автомат, представленный с помощью деревьев решений [7].

Функция приспособленности каждого решения зависит от результата моделирования процесса стабилизации модели балансирующего робота. Критерием оптимальности в данной работе выбран суммарный угол отклонения за время моделирования процесса стабилизации, который необходимо минимизировать.

Описание эксперимента

Для экспериментов была создана математическая модель двухколесного балансирующего робота, которая выступала в роли объекта управления.

Для сравнения рассмотренных методов в работе было сгенерировано десять наборов параметров (описание наборов параметров представлено в работе [8]), соответствующих десяти различным экземплярам роботов.

Далее для каждого робота были проведены следующие этапы эксперимента:

- параметрический синтез (на основе генетического алгоритма и метода отжига) и структурный синтез на основе конечных автоматов;
- для каждого метода синтеза выполнена генерация 1000 решений (экземпляров систем управления) для каждого экземпляра робота;
- тестирование каждого полученного решения на наборе тестов (тестирование проводилось на наборе из 100 тестов, каждый из которых представлял собой многократные отклонения в определенные моменты времени на определенных величинах угла);
- нахождение математического ожидания и дисперсии функции приспособленности для каждого метода для каждого экземпляра робота.

Результаты

Результаты проведенного эксперимента представлены в таблице 1.

	№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7	№ 8	№ 9	№ 10
<i>Параметрический синтез + генетический алгоритм</i>										
<i>M(F)</i>	128,9	54,52	65,66	95,56	30,63	92,47	92,21	139,62	88,43	95,65
<i>D(F)</i>	0,31	0,0033	0,12	0,012	0,41	0,013	0,014	0,0074	0,0011	0,011
<i>Параметрический синтез + метод отжига</i>										
<i>M(F)</i>	126,8	53,42	67,98	95,31	31,48	91,32	91,85	138,13	89,98	93,75
<i>D(F)</i>	0,22	0,038	0,022	0,014	0,021	0,15	0,019	0,047	0,028	0,045
<i>Структурный синтез</i>										
<i>M(F)</i>	136,4	59,72	71,73	106,36	38,73	98,24	97,32	136,81	93,77	99,29
<i>D(F)</i>	3,31	2,87	1,55	2,89	2,32	1,043	1,84	1,73	2,78	1,33

Таблица 1. Результаты эксперимента

Заключение

В результате работы для каждого экземпляра мобильного робота были получены системы управления с помощью как параметрического, так и структурного синтеза, которые успешно справились с задачей стабилизации на всех предложенных тестах в рамках эксперимента. При этом в среднем с помощью параметрического синтеза получалось создавать более эффективные системы управления, для девяти из десяти изученных экземпляров роботов эффективность систем управления, основанных на ПИД-регуляторе и параметрическом синтезе, оказалась выше, чем у систем управления, основанных на автоматах. Несмотря на это, структурный синтез с использованием автоматов представляет собой перспективное направление построения систем управления мобильными роботами. Улучшения результатов можно достичь, используя более точный выбор дискретизации входных воздействий.

Полученные результаты позволяют утверждать, что методы параметрического и структурного синтеза систем управления достаточно перспективны в плане их применения для построения систем управления мобильными роботами.

Литература

1. Chi R. Balancing a Two-Wheeled Autonomous Robot. The University of Western Australia School of Mechanical Engineering, 2003.
2. MathWorks Simulink Team, Simulink Support Package for LEGO MINDSTORMS NXT hardware (R2012a). [Электронный ресурс]. — Режим доступа: <http://www.mathworks.com/matlabcentral/fileexchange/35206-simulink-support-package-for-lego-mindstorms-nxt-hardware-r2012a>, свободный. Яз. англ. (дата обращения 3.10.2011).
3. Энциклопедия АСУ ТП. [Электронный ресурс]. — Режим доступа: http://www.bookasutp.ru/Chapter5_1.aspx, свободный. Яз. русс. (дата обращения 3.10.2012).

4. *Бесекерский В.А., Попов Е.П.* Теория систем автоматического регулирования. Издание третье, исправленное. М.: Наука, Главная редакция физико-математической литературы, 1975.
 5. *Шальто А.А.* Технология автоматного программирования / Труды Всероссийской научной конференции «Методы и средства обработки информации». М.: МГУ, 2003.
 6. *Лопатин А.С.* Метод отжига в задачах оптимизации. [Электронный ресурс]. — Режим доступа: <http://www.math.spbu.ru/user/gran/students/cothesis.pdf>, свободный. Яз. русс. (дата обращения 03.10.2012).
 7. *Данилов В.Р., Шальто А.А.* Метод генетического программирования для генерации автоматов, представленных деревьями решений. <http://is.ifmo.ru/download/2008-03-07-danilov.pdf>, свободный. Яз. русс. (дата обращения 07.03.2012).
 8. *Яковлев В.О.* Оценка эффективности применения методов искусственного интеллекта для синтеза системы управления мобильным роботом. Бакалаврская работа. [Электронный ресурс]. — Режим доступа: <http://is.ifmo.ru/papers/2011-bachelor-yakorev/>, свободный. Яз. русс. (дата обращения 07.03.2012).
-

МЕТОД СБОРКИ КОНТИГОВ ГЕНОМНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ИЗ ПАРНЫХ ЧТЕНИЙ С ОШИБКАМИ ВСТАВКИ И УДАЛЕНИЯ НА ОСНОВЕ СОВМЕСТНОГО ПРИМЕНЕНИЯ ГРАФОВ ДЕ БРЁЙНА И ГРАФОВ ПЕРЕКРЫТИЙ¹

Александров А.В.

*студент кафедры компьютерных технологий НИУ ИТМО,
alant@rain.ifmo.ru*

Казаков С.В.

*студент кафедры компьютерных технологий НИУ ИТМО,
svkazakov@rain.ifmo.ru*

Сергушичев А.А.

*студент кафедры компьютерных технологий НИУ ИТМО,
alserg@rain.ifmo.ru*

Научный руководитель:

Царев Ф.Н.

*к. т. н., ассистент кафедры программной инженерии
и верификации программ НИУ ИТМО*

Аннотация: Сборка гыштп генома является одной из важных задач современной биологии и медицины. Большинство из современных программ для сборки, ориентированы на работу с чтениями, в которых содержатся ошибки замены. В данной работе предлагается алгоритм, использующий небольшой объем оперативной памяти и позволяющий использовать чтения с ошибками вставки и удаления.

Введение

Многие современные задачи биологии и медицины требуют знания геномов живых организмов, который состоит из нескольких нуклеотидных последовательностей молекул дезоксирибонуклеиновой кислоты (ДНК). В связи с этим возникает необходимость в дешевом и быстром методе секвенирования, то есть определения последовательности нуклеотидов в образце ДНК.

Существующие секвенаторы – устройства для чтения ДНК — не позволяют считать за один раз всю молекулу ДНК. Вместо этого они позво-

¹ Исследование поддержано в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы» (Государственный контракт № 16.740.11.0495, соглашение № 14.В37.21.0562)

ляют читать фрагменты генома небольшой длины. Длина фрагмента может быть разной, она является важным параметром секвенирования — от нее напрямую зависит стоимость секвенирования и время, затрачиваемое на чтение одного фрагмента: чем больше длина считываемого фрагмента, тем выше стоимость чтения и тем дольше это чтение происходит. Современные высокопроизводительные секвенаторы основаны на разных подходах со своими достоинствами и недостатками. Одним из возможных недостатков, с которым необходимо справляться, является наличие ошибок вставки и удаления — в прочитанных фрагментах в некоторых местах возможно удаление или вставка символа, например, одной из распространенных ошибок является добавление или удаление одного или нескольких из серии одинаковых подряд идущих нуклеотидов. Часто для сборки используют парные чтения, для которых известно примерное расстояние между ними. Это позволяет получить больше информации о геноме при сравнительно небольших затратах.

Геном обычно представляют в виде нескольких строк из букв А, G, С и Т, соответствующих возможным нуклеотидам. Задачей сборки генома является восстановление последовательности ДНК (ее длина составляет от миллионов до миллиардов нуклеотидов у разных живых существ) на основании информации, полученной в результате секвенирования. Этот процесс делится, как правило, на следующие этапы:

1. Исправление ошибок в данных секвенирования.
2. Сборка *контигов* — максимальных непрерывных последовательностей нуклеотидов, которые удалось восстановить.
3. Построение *скэффолдов* — последовательностей контигов, разделенных промежутками, для длины которых найдены верхние и нижние оценки.

Одной из наиболее часто используемых при сборке генома математических моделей является так называемый граф де Брёйна [1]. На его использовании основаны следующие программные средства: *Velvet* [2], *Allpaths* [3], *AbySS* [4], *SOAPdenovo* [5], *EULER* [6].

Одним из недостатков, которым обладают перечисленные программные средства, является то, что эти программы были оптимизированы для работы с секвенаторами компании Illumina [7] и не поддерживают работу с большим числом ошибок вставки и удаления. Также эти методы требуют для своей работы большого объема оперативной памяти.

В настоящей работе предлагается метод, лишенный указанных недостатков. Построение скэффолдов в настоящей работе не рассматривается.

Предлагаемый метод

В работе предлагается модифицировать алгоритм, предложенный в работе [8].

Алгоритм исправления ошибок основан на анализе частот k -меров (строк длины k) и состоит из четырех шагов:

1. Разбиение k -меров на корзины согласно их префиксам. Фиксируется некоторая длина l . Для каждой из 4^l строк длины l выделяется множество k -меров, начинающихся с данного префикса.
2. Подсчет частот k -меров в одной корзине. Проходом по всем данным для каждого k -мера подсчитывается, сколько раз он встречается в наборе чтений, после чего все k -меры делятся на *надежные* (те, что встречаются много раз) и *ненадежные* (те, что встречаются мало раз).
3. Исправление ошибок в k -мерах одной корзины. Для каждого k -мера перебираются все возможные исправления, которые можно осуществить — на каждой позиции можно заменить, удалить или добавить символ. Максимальное число ошибок, исправляемых внутри одного k -мера, является параметром алгоритма. Внутри используемого при разбиении на корзины префикса ошибки не исправляются, благодаря чему исправленные k -меры лежат в той же корзине, что и исправляемые. Если ненадежный k -мер единственным образом превращается в надежный, найденное исправление записывается.
4. Применение исправлений. Для каждого чтения выписываются все k -меру, содержащиеся в нем. Те, к которым найдены исправления, заменяются на соответствующие надежные k -меры. Далее для каждой позиции по правилу консенсуса выбирается символ.

Разбиение на корзины позволяет проводить исправление ошибок и в большом объеме памяти.

После исправления ошибок происходит сборка контигов, выполняющаяся в два этапа: сборка квазиконтигов из парных чтений — фрагментов генома, из которых были получены парные чтения, и сборки контигов из квазиконтигов.

Сборка квазиконтигов осуществляется с использованием графа де Брёйна и состоит трех шагов:

1. Подсчет частот k -меров в исправленных чтениях. Выполняется так же, как и при исправлении ошибок.
2. Построение графа де Брёйна. В граф де Брёйна добавляются как вершины все k -меры, встречающиеся не меньше двух раз.
3. Поиск путей в графе де Брёйна, соответствующих фрагментам, из которых были получены парные чтения. Для каждой пары чтений в графе де Брёйна находится множество вершин и ребер, через которые проходит хотя бы один путь, соединяющий эти два чтения, подходящий по длине под априорные границы на расстояние между чтениями. В получившемся подграфе выделяется путь с наибольшим весом и проверяется, что все возможные пути, соединяющие парные чтения совпадают с этим путем с точностью до небольшого

числа ошибок замены, вставки или удаления. Если это так, то последовательность нуклеотидов на этом пути выводится как квазиконтиг.

Разбиение k -меров на корзине при подсчете частот и хранение в графе де Брёйна только ребер с их весами позволяет этому этапу также выполняться в небольшом объеме памяти.

Сборка контигов из квазиконтигов осуществляется с использованием графа перекрытий и метода *Overlap-Layout-Consensus*. Этот этап состоит из трех шагов:

1. Поиск перекрытий (в том числе неточных) между квазиконтигами. Выполняется с использованием суффиксного массива.
2. Построение и упрощение графа перекрытий – графа, в котором вершинами являются квазиконтиги, а ребрами – перекрытия между ними.
3. Вывод контигов. Для этого в упрощенном графе перекрытий находятся пути без ветвлений. Контиг получается консенсусом из множества квазиконтигов на этом пути с соответствующими сдвигами.

Экспериментальные исследования

Экспериментальные исследования проводились на библиотеке чтений бактерии *p_stutzeri*, геном которой имеет длину около 4 миллионов нуклеотидов. Данные получены секвенатором Ion Torrent. Всего использовалось три библиотеки, одна из которых содержала парные чтения со средним расстоянием между чтениями 2500 нуклеотидов. Общий объем исходных данных составлял 2,9 Гб.

После сборки было получено 434 контига суммарным размером в 4,6 Мбаз, максимальным контигом длиной 57 Кбаз и с N50 равным 17000. На тех же данных с помощью программы AiuSS было получено 50 тысяч контигов суммарной длиной в 6,3 Мбаз, максимальной длиной 7 Кбаз и N50 равным 660.

Заключение

В работе предложен метод сборки генома, использующий небольшой объем памяти, поддерживающий работу с чтениями, содержащими ошибки вставки и удаления.

Литература

1. *Pevzner P.A.* 1-Tuple DNA sequencing: computer analysis // *J. Biomol. Struct. Dyn.* 1989. Vol. 7, pp. 63–73.
2. *Zerbino D.R., Birney E.* Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research.* 2008. Vol. 18, pp. 821–829.

3. *Butler J., MacCallum I., Kleber M., Shlyakhter I.A., Belmonte M.K., Lander E.S., Nusbaum C., Jaffe D.B.* ALLPATHS: De novo assembly of wholegenome shotgun microreads // *Genome Research*. 2008. Vol. 18, pp. 810–820.
 4. *Simpson J.T., Wong K., Jackman S.D., Schein J.E., Jones S.J., Birol I.* ABySS: A parallel assembler for short read sequence data // *Genome Research*. 2009. Vol. 19, pp. 1117–1123.
 5. *Li R., Zhu H., Ruan J., Qian W., Fang X., Shi Z., Li Y., Li S., Shan G., Kristiansen K., et al.* De novo assembly of human genomes with massively parallel short read sequencing // *Genome Research*. 2010. Vol. 20, pp. 265–272.
 6. *Pevzner P.A., Tang H., Waterman M.S.* EULER: An Eulerian path approach to DNA fragment assembly // *Proc. Natl. Acad. Sci.* 2001. №98, pp. 9748–9753.
 7. Illumina, Inc. [Электронный ресурс]. — Режим доступа: <http://www.illumina.com/>, свободный. Яз. англ. (дата обращения 23.04.2013).
 8. *А.В. Александров, С.В. Казаков, С.В. Мельников, А.А. Сергушичев, Ф.Н. Царев, А.А. Шальто.* Метод сборки контигов геномных последовательностей на основе совместного применения графов де Брюина и графов перекрытий. // *Список-2012: Материалы всероссийской научной конференции по проблемам информатики.* — 2012. — С. 415–418.
-

Синтез элементов компьютерной архитектуры



**Леонов
Геннадий Алексеевич**

председатель оргкомитета конференции

д.ф.-м.н., профессор, чл.-корр. РАН
декан математико-механического факультета СПбГУ
заведующий кафедрой прикладной кибернетики СПбГУ

НЕЛИНЕЙНЫЙ АНАЛИЗ ЦИФРОВЫХ СИСТЕМ ФАЗОВОЙ АВТОПОДСТРОЙКИ ЧАСТОТЫ¹

Кудряшова Е.В.

кафедра прикладной кибернетики СПбГУ, kudryashova_lena@mail.ru

Кузнецова О.А.

кафедра прикладной кибернетики СПбГУ, o_a_kuznetsova@mail.ru

Селеджи С.М.

кафедра прикладной кибернетики СПбГУ, ssm@SS1563.spb.edu

Аннотация: Проблемы синхронизации систем возникают в различных областях науки и техники, в таких, например, как системы фазовой автоподстройки (ФАП). В работе рассматриваются бифуркационные эффекты, возникающие в одномерных дискретных динамических системах, описывающих фазовую автоподстройку частоты в простейших цифровых системах ФАП.

В работе представлены численные значения бифуркационных параметров такой системы, вычисленные с помощью специальных аналитических методов и современных математических пакетов длинных чисел. Кроме того, показано, что для рассмотренного неунимодального отображения наблюдается эффект сходимости бифуркационных значений параметра системы, аналогичный эффекту Фейгенбаума для унимодальных отображений.

Введение

В системах фазовой автоподстройки (ФАП), используемых для синхронизации и генерации сигналов в радиосвязи, телекоммуникациях и компьютерных архитектурах, возникают проблемы синхронизации частот [1–19]. Качественный анализ уравнений ФАП позволяет определить необходимые условия работы системы (при которых, например, имеются синхронизация частот и коррекция расфазировок) [20, 21]. В предложенной работе рассматривается дискретное одномерное неунимодальное отображение, описывающее работу цифровой ФАП.

В одной из первых работ, посвященных анализу цифровых ФАП [22], был рассмотрен алгоритм исследования периодических решений и показано, что даже в простой дискретной модели ФАП наблюдаются бифуркационные явления, приводящие к появлению новых устойчивых периодических решений и изменению их периода. В дальнейшем в работах [23] для

¹ Работа выполнена при финансовой поддержке проектов Совета по грантам при Президенте РФ, Минобрнауки РФ, РФФИ и Санкт-Петербургского государственного университета.

таких систем была рассмотрена модель перехода к хаосу через каскад удвоения периода. Объединение и развитие этих идей в работах Г.А. Леонова и С.М. Селеджи [20, 21] позволило построить бифуркационное дерево перехода к хаосу через каскад удвоения периода. Для этого аналитически были получены первые несколько бифуркационных значений параметров, в то время как расчет последующих бифуркационных значений и изучение хаоса потребовали применения компьютерного моделирования. Эти вычисления выявили эффект аналогичный эффекту Фейгенбаума для унимодальных отображений [20, 24, 25]. Позднее в работах [26–28] для получения более точных численных значений бифуркационных параметров рассматриваемой системы потребовалось применение качественной теории динамических систем, специальных аналитических методов и современных математических пакетов длинных чисел.

Вычисление бифуркационных параметров

Рассмотрим блок-схему простейшей дискретной ФАП с аналоговым входом (См. Рис. 1).

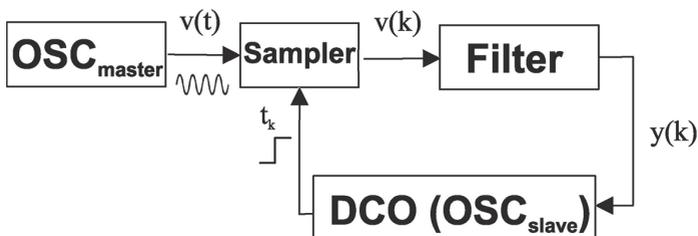


Рис. 1. Функциональная блок-схема ФАП

Здесь эталонный генератор $\text{OSC}_{\text{master}}$ вырабатывает синусоидальный сигнал:

$$v(t) = A \sin(\omega_{in} t + \theta_0), \quad (1)$$

где A — амплитуда, θ_0 — сдвиг по фазе, ω_{in} — частота,

$\phi(t) = \omega_{in} t + \theta_0$ — фаза входного сигнала,

$T_{in} = \frac{2\pi}{\omega_{in}}$ — период входного сигнала.

Сигнал $v(t)$ поступает на **Sampler** (дискретизатор) и преобразуется в дискретный сигнал $v(k)$ в моменты времени $\{t_k\}_{k=1,2,\dots}$, определяемые импульсами управляемого генератора **DCO**. Выход дискретизатора $v(k)$ пропуска-

ется через фильтр (**Filter**), и оставшийся на выходе фильтра сигнал $y(k)$ поступает на вход управляемого генератора **DCO**.

Случай совпадения начальных частот эталонного и подстраиваемого генераторов имеет большое значение в инженерной практике дискретных ФАП [20, 29]. Определяя значение сдвига по фазе в интервале $[-\pi, \pi]$ с учетом кратности 2π , можно в этом случае, согласно работам [20, 21], перейти к уравнению:

$$x_{n+1} = x_n - r \sin x_n, \quad n \in N, \quad (2)$$

где r — положительное число.

В работе [20] указана верхняя граница параметра r , при котором система (2) является отображением отрезка в себя, и аналитически получены первые три бифуркационных параметра системы $r_1 = 2$, $r_2 = \pi$, $r_3 = \sqrt{\pi^2 + 2}$. Было доказано, что при $r < r_1$ система является глобально асимптотически устойчивой, параметры r_1 и r_3 соответствуют бифуркации удвоения периода, а параметр r_2 соответствует бифуркации «расщепления» цикла: глобально устойчивый цикл периода 2 теряет устойчивость и рождаются два локально устойчивых цикла периода 2. В дальнейшем система переходит к хаосу через каскад бифуркаций удвоения периода.

Явление перехода к хаосу через бифуркации удвоения периода хорошо изучено для целого класса отображений отрезка в себя. В 1975 г. при исследовании бифуркаций удвоения периода логистического отображения

$$x_{n+1} = \lambda x_n (1 - x_n), \quad n \in N, \quad (2)$$

М. Фейгенбаум заметил, что для последовательных бифуркационных значений параметра λ справедлива сходимость:

$$\frac{\lambda_n - \lambda_{n-1}}{\lambda_{n+1} - \lambda_n} = \delta_n, \quad \lim_{n \rightarrow \infty} \delta_n = 4,6692\dots, \quad 16.1$$

и что аналогичный эффект имеет место для многих других отображений. В дальнейшем развитие теории ренорм-групп позволило доказать универсальность открытого Фейгенбаумом эффекта для одномерных однопараметрических унимодальных отображений отрезка в себя [30–34].

Отметим, что функция $f(x) = x - r \sin x$, представленная на Рис. 2, не является унимодальной или бимодальной. Здесь $f(x)$ является нечетной функцией и имеет два локальных экстремума $\pm x^m$, а последовательность $\{x_n\}$, начатая в точке x^m , не проходит через точки локальных экстремумов $f(x)$ и $f(f(x))$ — последовательность отображена на рисунке в виде линий последовательно соединяющих точки $(x, f(x))$, $(f(x), f(x))$, $(f(x), f(f(x)))$, $(f(f(x)), f(f(x)))$, ...

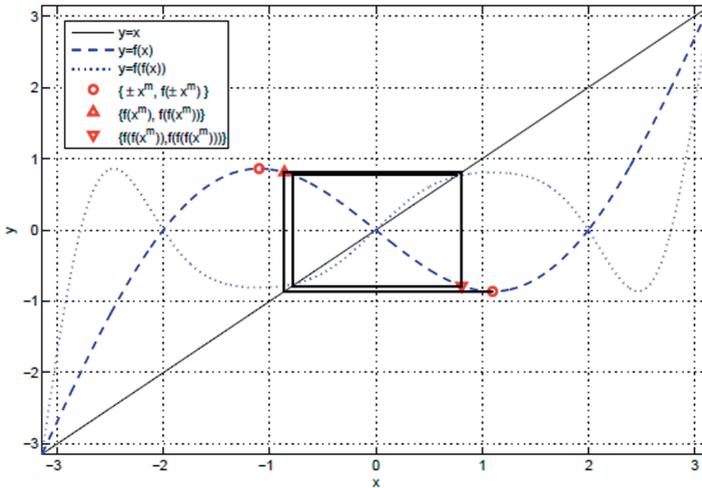


Рис. 2. Отображение $f(x) = x - r \sin(x)$, $r = 2.2$

Применение современных вычислительных пакетов, процедуры вычисления мультипликаторов и теории переходных процессов позволило получить численно первые 15 бифуркационных значений параметра r системы (2), которые представлены в Таблице 1. Для полученных бифуркационных значений r_j были также получены числа Фейгенбаума, которые представлены в последней колонке таблицы.

№	T	r_j	Δ_j
1	1/2	2	
2	1/2	Π	3.75973373258170937649
3	2/4	3.44522922330131157542194433174191398	4.624046596639769740149960075495
4	4/8	3.51289246475156628374774642737133234	4.660150061404217425922602823766
5	8/16	3.52752537124074929576590188097963033	4.6671810989023526532815779401952
6	16/32	3.53066537881483452997938899118568510	4.668778772473917134313391117376
7	32/64	3.53133816341597711569260339404602099	4.669109815433731376068465041681
8	64/128	3.53148226632466689433857302550273505	4.6691821019439780251939905083339
9	128/256	3.53151312936160242470979217662168692	4.6691974124177884207133643832507
10	256/512	3.53151973930661242566549959454182848	4.6692007126313128442197371622798
11	512/1024	3.53152115495569002125505077904545367	4.6692014168180717991991829239267
12	1024/2048	3.53152145814442932002148295601459890	4.669201567957076525629797957062
13	2048/4096	3.53152152307817542202214372051946438	4.6692016002863903548904303401188
14	4096/8192	3.53152153698499549756050390115891712	
15	8192/16384	3.53152153996341049568039036147220653	

Таблица 1. Бифуркационные значения и числа Фейгенбаума дискретной динамической системы

Заключение

Из полученных значений видно, что здесь для неунимодального отображения (2) наблюдается эффект сходимости бифуркационных значений параметра, аналогичный эффекту Фейгенбаума для унимодальных отображений.

Литература

1. *G.A. Leonov, N.V. Kuznetsov, S.M. Seledzhi*. Nonlinear Analysis and Design of Phase-Locked Loops (chapter in «Automation control – Theory and Practice»)// In-Tech. — 2009. — Pp. 89–114.
2. *Kuznetsov N.V.* Nonlinear Analysis of Phase Synchronization Systems: Phase-locked Loop and Costas Loop// Seminar series on Complex systems, networks, control and chaos. — City University of Hong Kong. — 2012. — Invited lecture.
3. *G.A. Leonov, N.V. Kuznetsov, M.V. Yuldashev, R.V. Yuldashev*. Analytical method for computation of phase-detector characteristic// IEEE Transactions on Circuits and Systems Part II. — Express Briefs. — Vol. 59. — Num. 10. — 2012. — Pp. 633–637.
4. *G.A. Leonov, N.V. Kuznetsov, M.V. Yuldashev, R.V. Yuldashev*. Differential equations of Costas loop// Doklady Mathematics. — 86(2). — 2012. — Pp. 723–728.
5. *G.A. Leonov, N.V. Kuznetsov, M.V. Yuldashev, R.V. Yuldashev*. Computation of Phase Detector Characteristics in Synchronization Systems// Doklady Mathematics. — 2011. — Vol. 84. — No. 1. — Pp. 586–590.
6. *N.V. Kuznetsov, G.A. Leonov, M.V. Yuldashev, R.V. Yuldashev*. Analytical methods for computation of phase-detector characteristics and PLL design// ISSCS 2011 – IEEE International Symposium on Signals, Circuits and Systems, Proceedings. — 2011. — Art. num. 5978639. — Pp.7–10.
7. *G.A. Leonov, S.M. Seledzhi, N.V. Kuznetsov, P. Neittaanmäki*. Asymptotic analysis of phase control system for clocks in multiprocessor arrays// ICINCO 2010 — Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics. — Vol. 3. — 2010. — Pp. 99–102.
8. *N.V. Kuznetsov, G.A. Leonov, P. Neittaanmäki, S.M. Seledzhi, M.V. Yuldashev, R.V. Yuldashev*. Nonlinear Analysis of Phase-locked loop// IFAC Proc. Volumes (IFAC-PapersOnline). — 4(1). — 2010. — Pp. 34–38.
9. *N.V. Kuznetsov, G.A. Leonov, S.M. Seledzhi*. Nonlinear analysis of the Costas loop and phase-locked loop with squarer// Proceedings of Eleventh IASTED International Conference Signal and Image Processing. — Vol. 654. — 2009. — Pp. 1–7. — ACTA Press.
10. *N. Kuznetsov, G. Leonov, S. Seledzhi*. Phase Locked Loops Design And Analysis// ICINCO 2008 — 5th International Conference on Informatics in Control, Automation and Robotics. — Proceedings Volume SPSMC. — 2008. — Pp. 114–118.
11. *N.V. Kuznetsov, G.A. Leonov, S.M. Seledzhi*. Analysis of phase-locked systems with discontinuous characteristics of the phase detectors// IFAC Proceedings Volumes (IFAC-PapersOnline). — 1(1). — 2006. — Pp. 107–112.
12. *Kuznetsov N.V., Leonov G.A., Neittaanmäki P., Seledzhi S.M., Yuldashev M.V., Yuldashev R.V.* Simulation of phase-locked loops in phase-frequency domain// International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, — 2012. — Pp. 351-356. — Art. no. 6459692.

13. *Kuznetsov N.V., Leonov G.A., Neittaanmäki P., Seledzhi S.M., Yuldashev M.V., Yuldashev R.V.* Nonlinear mathematical models of Costas Loop for general waveform of input signal// IEEE 4th International Conference on Nonlinear Science and Complexity, NSC 2012 Proceedings. — 2012. — Pp. 109–112.
14. *Kuznetsov N.V., Leonov G.A., Yuldashev M.V., Yuldashev R.V.* Nonlinear analysis of Costas loop circuit// ICINCO 2012 — Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics. — 2012. — Vol. 1. — Pp. 557–560.
15. *Kuznetsov N.V., Neittaanmäki P., Leonov G.A., Seledzhi S.M., Yuldashev M.V., Yuldashev R.V.* High-frequency analysis of phase-locked loop and phase detector characteristic computation// ICINCO 2011 — Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics. — 2011. — Vol. 1. — Pp. 272–278.
16. *Kuznetsov N.V., Leonov G.A., Seledzhi S.M.* Phase synchronization and control of clock generators// 7th Seminar of Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program. — 2010. — Pp. 76–82.
17. *Kuznetsov N.V., Leonov G.A., Neittaanmäki P., Seledzhi S.M., Yuldashev M.V., Yuldashev R.V.* Nonlinear analysis of Phase-Locked Loop// Mathematical and Numerical Modeling in Science and Technology. — 2010.
18. *Kuznetsov N.V., Leonov G.A., Seledzhi S.M., Neittaanmäki P.* Analysis and design of computer architecture circuits with controllable delay line// ICINCO 2009 — 6th International Conference on Informatics in Control, Automation and Robotics. — 2009. — Vol. 3 SPSMC. — Pp. 221–224.
19. *D. Abramovitch* Phase-locked loops: A control centric tutorial// In Proceedings of the American Control Conference. — 1. — 2002. — Pp. 1–15. — Plenary lecture.
20. *Леонов Г.А., Селеджи С.М.* Системы фазовой синхронизации в аналоговой и цифровой схемотехнике// СПб. — Невский диалект. — 2002.
21. *Leonov G.A., Seledzhi S.M.* Stability and bifurcations of phase-locked loops for digital signal processors// International Journal of Bifurcation and Chaos.. — 2005. — 15(4). — Pp. 1347–1360.
22. *Osborne H.C.* Stability analysis of an Nth power digital phase-locked loop — Part 1: First-order DPLL// IEEE Transactions on Communications. — 1980. — Vol. 28. — No 8. — Pp. 1343–1354.
23. *Белых В.Н., Максаков В.П.* Разностные уравнения и динамика цифровой системы фазовой синхронизации первого порядка// Радиотехника и электроника. — 1979. — 24(5). — С. 958–964.
24. *Abramovich S., Kudryashova E., Leonov G.A., Sugden S.* Discrete Phase — Locked Loop Systems and Spreadsheets// Spreadsheets in Education (eJSiE). — 2005. — Volume 2. — Issue 1.
25. *Saleh R. Al-Araji, Zahir M. Hussain, Mahmoud A. Al-Qutayri.* Digital Phase Lock Loops: Architectures and Applications // Springer. — 2006.
26. *Кудряшова Е.В.* Вычисление бифуркационных параметров для цифровой системы фазовой автоподстройки// Вестник С.-Петерб. ун-та. — 2009. — Сер. 10. — Вып. 3. — С. 78–81.

27. *Kudryashova E.V.* Cycles in continuous and discrete dynamical systems: Computations, computer-assisted proofs, and computer experiments// Jyvaskyla Univ. Printing House. — 2009.
 28. *Шурухова Д.К.* Аппроксимация бифуркационных параметров удвоения периода для дискретных систем фазовой синхронизации// Дипломная работа. — СПбГУ. — 2012.
 29. *Banerjee T., Sarkar B.C.* Chaos, intermittency and control of bifurcation in a ZC2-DPLL // Int. J. Electron. Commun. — 2009. — 96(7). — Pp. 717–732.
 30. *Cvitanovich P.* Universality in Chaos// 2nd Edition. — Adam Hilder Publ. — 1989.
 31. *Feigenbaum M.J.* Quantitative universality for a class of nonlinear transformations// Journal of Statistical Physics. — 1978. — 19. — Pp. 25–52.
 32. *Вул Е.Б., Синай Я.Г., Ханин К.М.* Универсальность Фейгенбаума и термодинамический формализм// УМН. — 39:3(237) . — 1984. — С. 3–37.
 33. *Campanino M., Epstein H.* On the existence of Feigenbaum fixed-point// Comm, Math, Phys. — 1981. — Vol. 79. — №2. — Pp. 261–302.
 34. *Lanford O.E.* A computer assisted proof of the Feigenbaum conjectures// Bull. Amer. Math. Soc. — 1982. — 6(3). — Pp. 427–434.
-

РАЗВИТИЕ МЕТОДОВ И ТЕХНОЛОГИЙ В ОБЛАСТИ ИНФОРМАЦИОННОГО МЕНЕДЖМЕНТА

Арсеньев Д.Г., Кузнецов Н.В., Леонов Г.А.,

Оносовский В.В., Шкодырев В.П.

Санкт-Петербургский государственный университет

Санкт-Петербургский политехнический университет

В докладе отражена совместная деятельность исследовательских групп СПбГУ и СПбГПУ в области информационного менеджмента. Рассмотрен широкий класс задач создания новых методов и алгоритмов управления на основе современных информационных технологий., включая такие задачи как анализ и синтез управляющих сетей контроллеров для децентрализованного управления распределенными объектами и процессами, автоматизация и интеграция бизнес-процессов, создание систем бизнес-аналитики и поддержки принятия решений, визуализация математических моделей и расчетов, создание компьютерных презентаций и анимаций.

Уделено внимание разработке и исследованию адаптивных вычислительных алгоритмов на базе стохастических методов, обладающих повышенной скоростью сходимости и возможностью глубокой автоматизации вычислительных процессов, что удается достичь за счет применения идей теории адаптивного управления к таким специфическим объектам, как стохастические вычислительные процессы. Другое перспективное направление связано с развитием концепции и принципов построения распределенных интеллектуальных вычислительных систем и сетей, ориентированных на решение широкого класса сложных и плохо формализуемых задач управления физическими процессами и явлениями. Авторами рассмотрены различные направления использования распределенных интеллектуальных управляющих систем для широкого круга задач группового управления промышленными комплексами и технологическими установками.

Во всех перечисленных направлениях авторами ведутся интенсивные исследования и разработки, ориентированные на практические приложения.

1. *Г.А. Леонов, В.И. Кияев, Н.В. Кузнецов, В.В. Оносовский, С.М. Селеджи*, Некоторые аспекты подготовки IT-менеджеров: специальные курсы и производственная практика — органически связанные части образовательного процесса, стр. 297–528 [Леонид Витальевич Канторович: математика, менеджмент, информатика / Под ред. Г.А. Леонова, В.С. Катъкало, А.В. Бухвалова. СПб.: Изд-во Высшая школа менеджмента, 2010. 596 с. (ISBN 978-5-9924-0044-1)]
2. *Abramovich S., Kuznetsov N.V., Kuznetsov S.V., Leonov G.A., Onosovsky V.V., Seledzhi S.M.*, Learning to develop and use software products: some common aspects of educational preparation of mathematicians and schoolteachers, *Procedia Technology*, 2013 [submitted]

3. *Leonov G.A., Kiyayev V.I., Kuznetsov N.V., Onosovsky V.V., Seledzhi, S.M.*, Computers and software engineering: Developing new models for educating mathematicians, in book *Computers in Education. Volume 2*, Nova Science Publishers, 2012 (ISBN: 978-1-62100-623-7)
 4. *Д.Г. Арсеньев, В.М.Иванов, В.П. Шкодырев.* Адаптивное управление вычислительными процессами. СПб. Наука 2011
 5. *Д.Г. Арсеньев, В.П. Шкодырев.* Распределенные интеллектуальные системы промышленной автоматизации и управления Робототехника. Взгляд в будущее. Межд. Научно-техн.семинар. Март, 2010, СПб.
-

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ КОМПАКТНОГО МОБИЛЬНОГО УСТРОЙСТВА

В.В. Коновалов

младший научный сотрудник лаборатории информационных систем

Аннотация: Современные электронные карманные устройства обладают достаточно высокой вычислительной мощностью, что позволяет реализовывать на них полноценные алгоритмы на высокоуровневых языках программирования.

В статье предложен метод создания приложений для таких устройств, в котором сочетается одновременно полная переносимость на другие аналогичные устройства, а также на полноценный компьютер. В некоторых случаях возможно внесение усовершенствований в алгоритм непосредственно на данном компактном устройстве, в особенности если имеется клавиатура.

Введение

Для различных задач используются различные языки программирования и среды разработки, которые учитывают специфику конкретно поставленной задачи.

Для того, чтобы приложение работало на различных платформах, одним из наиболее удобных решений является использование языка программирования JavaScript. Фактически данный язык программирования присутствует в любом устройстве, которое поддерживает выход в глобальную сеть Internet.

Всевозможные мобильные устройства имеют различные архитектуры, и для каждой из этих архитектур имеется, как правило, среда разработки (SDK), предназначенная для разработки приложений для операционной системы данной архитектуры. Такая среда разработки генерирует машинный код, выполняемый на рассматриваемом устройстве и не совместим с другими устройствами. В связи с этим использование SDK является трудоёмким и узконаправленным решением.

Однако, если в устройстве имеется Internet browser, в этом случае появляется возможность реализовывать приложения на языке Javascript, что позволяет существенно облегчить разработку и обеспечить переносимость приложения.

Метод разработки информационной системы для компактного мобильного устройства

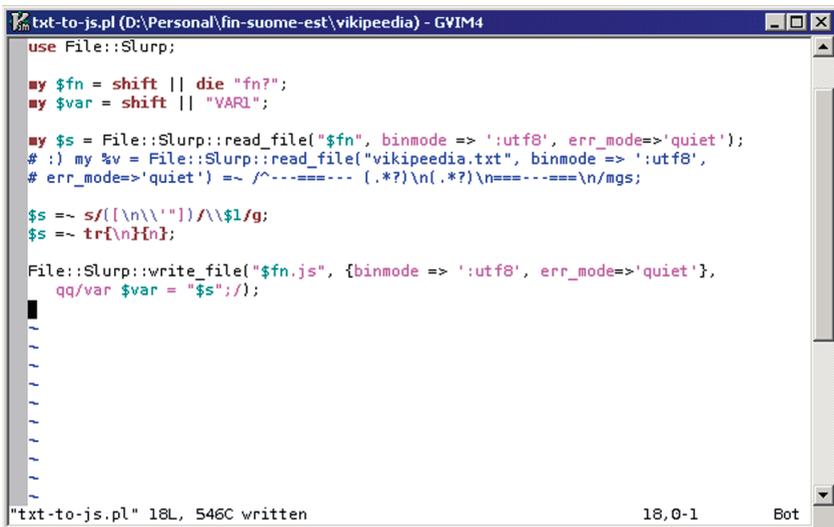
В силу исторических причин, связанных со спецификой использования Javascript, в качестве основного языка для создания динамических страниц

HTML, в данном языке изначально отсутствовали функции работы с файлами локальной файловой системы.

Поскольку в языке Javascript отсутствует возможность чтения файлов [4], [1], имеется проблема получения входных данных. Имеется возможность использования технологии баз данных [2], однако черновая версия спецификации не принята в технологии HTML и является несоместимой и проблематичной.

В данной статье предлагается использование подхода, в котором программные данные включаются в систему более удобно реализуемым и универсальным способом.

А именно, для не слишком большого объема данных (несколько мегабайт) возможно преобразование программных данных напрямую в текст кода Javascript, при помощи следующей Перл-программы:



```

txt-to-js.pl (D:\Personal\fin-suome-est\wikipedia) - GVIM4
use File::Slurp;

my $fn = shift || die "fn?";
my $var = shift || "VAR1";

my $s = File::Slurp::read_file("$fn", binmode => ':utf8', err_mode=>'quiet');
# :) my %v = File::Slurp::read_file("wikipedia.txt", binmode => ':utf8',
# err_mode=>'quiet') =~ /^----- (.*?)\n(.*?)\n=====\/mgs;

$s =~ s/([\n\'"])/\\$1/g;
$s =~ tr/\n/{};

File::Slurp::write_file("$fn.js", {binmode => ':utf8', err_mode=>'quiet'},
qq/var $var = "$s"/);

"txt-to-js.pl" 18L, 546C written                                18,0-1      Bot
  
```

Рис. 1. Программа преобразования текстового файла в код Javascript

Например, имелся текстовый файл с данными следующего вида —

```

----- Eesti
eesti <eesti>эстонский
-----
----- a
<_ _ _[d-, _[de _[sid>буква _
-----
----- aabe
  
```

```

<_ aape _[t-, _[te aape[id>буква
=====
----- aabits

-----
<_ a _a[t-, _a[te _a[id>букварь, азбука, азы чего
=====
----- aabitsa
азбучный, букварный
=====
----- aabitsatõde
азбучная ~ прописная истина
=====
----- aade
<_ aate _[t-, _[te aate[id>идеал, [высокая] идея
=====

```

Для него создаётся файл с расширением JS со следующим содержимым:

```

var sonastik = ";Eesti^eesti эстонский;a^_ 1.
буква _;aabe^_ буква;aabits^_ букварь, азбука, азы
чего;aabitsa^_ азбучный, букварный;aabitsatõde^_
азбучная ~ прописная истина;aade^_ идеал, [высокая]
идея;"

```

Затем, после включения данного фрагмента Javascript строкой

```
<script src="so~nastik-1-small.str.js"></script>
```

в переменной имеется требуемый текстовый массив.

Пример приложения

Поставлена задача со следующими входными ограничениями:

- приложение должно иметь удобный пользовательский интерфейс;
- использовать (мобильное) устройство с небольшим объёмом памяти.
- поиск по словарной статье выполняется в режиме offline, то есть без доступа в интернет
- предполагается переносимость на различные устройства, то есть желательное отсутствие привязанности к конкретной модели устройства

Примерный интерфейс имеет следующий вид —

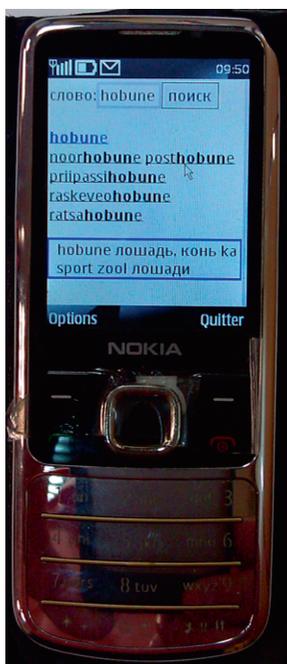


Рис. 2. Мобильный словарь на Nokia C6700c

Аналогичное приложение на устройстве с большим количеством памяти допускает наличие более расширенных словарных статей.

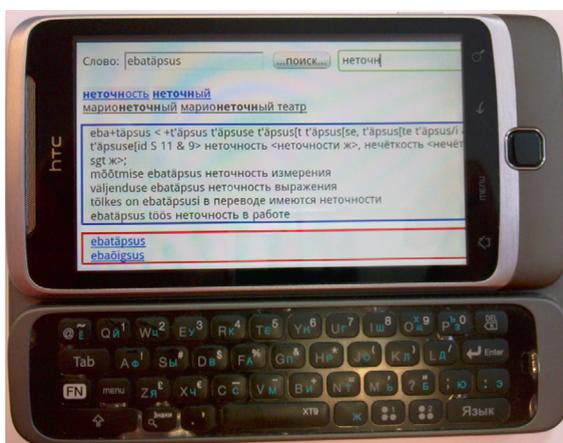


Рис. 3. Мобильный словарь на устройстве с Android

Приведённый в качестве примера интерфейс предусматривает наличие подсказки словарной статьи в процессе набора слова. Также, возможно использование стандартных библиотек для реализации визуальных эффектов и, например, распаковки сжатых данных (zlib-inflate.js [3]).

```

<html>
<head><meta charset="utf-8"></head>
<body>
<style type="text/css">
#res {
    background-color: #eeeeee;
    outline-style: groove;
    outline-color: #1111ff;
    padding-left: 10px;
}
.sona-follow {
    display: inline;
    background-color: #eeeeee;
    text-decoration: underline;
    color: blue;
}
.sona-follow-m {
    display: inline;
    background-color: #f0f0f0;
    text-decoration: underline;
    color: black;
}
</style>

<p>слово:<input style="width: 2cm;" id="ee" type="text"></
input><button id="ee-seek">поиск</button></p>
<div id="preview"> <p>...</p> </div><div id="res"> </div>

<script src="so~nastik-1-small.str.js"></script>

<script>
var ee =document.getElementById("ee");
var res =document.getElementById("res");
var prev =document.getElementById("preview");
var butt =document.getElementById("ee-seek");

ee.onkeyup = function () {
    // seek in word list ;list
    var que = ee.value.replace(/ +$/, '').toLowerCase();

```

```

var k=0, k0, k2, k3, r = new Array(), cq="";+que;
var k4 = cq.length, k5;

// beginning of word .... ;blabla^
while ( (k0=sonastik.indexOf(cq,k)) != -1) {
    k5 = sonastik.indexOf("^",k0+1);
    r.push(
        "<div class='sona-follow'
        onclick='javascript:ee.value=\"\"
        +sonastik.substring(k0+1, k5)+\"\";
        search_s();'>"+
        "<b>"+que+"</b>"+sonastik.substring(k0+k4, k5)
        + "</div>"
    );
    k=k0+1;
    if (r.length>15)
        break;
}

// middle of word ....
r.push(' <br> ');
k=0;
while ( (k0=sonastik.indexOf(que,k)) != -1) {
    if (sonastik.substring(k0-1,k0) != ";" ) {
        // BUT- check that it falls into word definition,
        first!
        k2 = sonastik.indexOf(';',k0+1);
        k3 = sonastik.indexOf('^',k0+1);
        if (k2<k3) {
            // indeed, false positive
            k=k0+1;
            continue;
        }
        k5 = sonastik.lastIndexOf(";",k0)+1;
        r.push(
            "<div class='sona-follow-m'
            onclick='javascript:ee.value=\"\"
            +sonastik.substring(k5,k3)+\"\";search_s();'>"+
            sonastik.substring
            (sonastik.lastIndexOf(";",k0)+1,k0)
            + "<b>" + que + "</b>"
            + sonastik.substring(k0+que.length,k3)
            +"</div>"
        );
    }
}

```

```
k=k0+1;
if (r.length>25)
    break;
}
prev.innerHTML = "<p>" + r.join(' ') + "</p>";
}

function search_s () {
    var s = ee.value.replace(/ +$/, '').toLowerCase(), k0, k1;
    if ((k0=sonastik.indexOf(";"+s+"^")) != -1) {
        k0 += 2+s.length;
        // found...
        k1=sonastik.indexOf(";", k0);
        var str=sonastik.substring(k0,k1); //assert k1!=-1
        res.innerHTML = "<p>"+
            str.replace(/_/g, s).replace(/&/g, '&amp;');
            .replace(/</g, '&lt;');
            .replace(/\n/g, "<br>")+ "</p>";
    }
}
butt.onclick = search_s;
ee.focus();
</script>
</body></html>
```

Заключение

Широкие возможности языка программирования Javascript делают возможным создание полноценных приложений для мобильных устройств, как работающих в режиме онлайн, так и без подключения к интернету. Во многих случаях создание таких приложений возможно непосредственно на мобильном телефоне, что было экспериментально подтверждено.

Литература

1. ECMAScript Language Specification (<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>)
2. <http://dev.w3.org/html5/webstorage/>
3. <http://www.onicos.com/staff/iz/release/zlib-js/zlib-js.html>
4. JavaScript: The Definitive Guide, 5th Edition, Publisher: O'Reilly Pub Date: August 2006 Print ISBN-10: 0-596-10199-6 Print ISBN-13: 978-0-59-610199-2

ИННОВАЦИОННЫЕ ПОДХОДЫ К ПОДГОТОВКЕ СОВРЕМЕННЫХ ВЫСОКОКВАЛИФИЦИРОВАННЫХ СПЕЦИАЛИСТОВ В ОБЛАСТИ ИТ¹

Николай Кузнецов

*доцент кафедры прикладной кибернетики
Математико-механического факультета СПбГУ,
куратор производственной практики студентов в ИТ-компаниях*

Геннадий Леонов

*заведующий кафедрой прикладной кибернетики
Математико-механического факультета СПбГУ*

Валентин Оносовский

*директор отделения компании Exigen Services в Санкт-Петербурге,
научный сотрудник Математико-механического факультета СПбГУ*

Светлана Селеджи

*зам. декана, руководитель Центра переподготовки специалистов в области
математики и информатики при СПбГУ, ст. преподаватель кафедры
прикладной кибернетики Математико-механического факультета СПбГУ*

Евгений Филиппов

*исполнительный директор компании Exigen Services по странам СНГ,
в прошлом — руководитель направления
в Санкт-Петербургском подразделении компании Motorola*

Введение

В настоящее время невозможно представить себе подготовку высококвалифицированного программиста без компьютера. Также, на наш взгляд, невозможно полноценное образование будущего создателя сложного программного продукта без понимания того, как создается такой продукт, без получения практических навыков работы в коллективах, производящих реальный программный продукт. Верно и обратное: студенты, проходящие производственную практику в ИТ-компаниях, совершенно по другому – на более высоком уровне — воспринимают специальные курсы по управлению ИТ-проектами и управлению качеством разработки программного обеспечения (ПО).

¹ Работа выполнена при финансовой поддержке проектов Совета по грантам при Президенте РФ, Минобрнауки РФ, РФФИ и Санкт-Петербургского государственного университета.

Идея объединения теоретических курсов и практики для подготовки руководителей и участников ИТ-проектов осуществляется на протяжении последних более чем 6 лет на кафедре прикладной кибернетики математико-механического факультета [1–4]. Впервые соответствующая образовательная программа стартовала для группы студентов 3 курса в осеннем семестре 2006 года на отделении прикладной математики и информатики. Реализация этой программы была продолжена на кафедре прикладной кибернетики после её образования в мае 2007 года и являлась одним из основополагающих принципов построения кафедры.

Другим важным принципом при организации кафедры являлся принцип мобильности. Если в области чистой математики модернизация образования проходит постепенно, и хорошо поставленные курсы могут практически не меняться в течение нескольких десятков лет, то в области ИТ модернизация образования (в том числе создание новых курсов и семинаров и отказ от курсов, отслуживших свой срок) проходит чрезвычайно быстрыми темпами, причем новые курсы и практикумы в большой степени должны быть согласованы с реальными технологиями и средствами разработки ПО. Из этого следует, что на кафедре необходимо эффективное сочетание небольшого ядра штатных преподавателей (в настоящее время четыре человека) и значительного количества приглашенных профессоров и преподавателей — специалистов в конкретных ключевых направлениях ИТ-науки и ИТ-технологий.

Приведем примеры. По измененному учебному плану в начале третьего курса начинается производственная практика студентов кафедры в ИТ-компаниях Санкт-Петербурга. В настоящее время одной из основных платформ, используемых для промышленной разработки ПО для бизнеса, является платформа Java. И в компаниях, где студенты проходят практику, платформа Java также является одной из основных. Следовательно, в первом семестре третьего курса необходимо провести интенсивное обучение языку программирования Java и различным приложениям, которые разрабатываются на этом языке. Для чтения спецкурса «Java-технологии» кафедрой приглашен профессор В.О. Сафонов — руководитель лаборатории Java-технологий, широко известный специалист в этой области. Одновременно с этим доцент кафедры прикладной кибернетики Н.В.Кузнецов проводит соответствующий практикум по Java в компьютерном классе. И, наконец, совместно с компанией Exigen Services разработан спецкурс «Введение в бизнес-программирование на платформе Java/J2EE», который читается специалистами Exigen в течение всего третьего курса.

Таким образом, студенты кафедры прикладной кибернетики, мотивированные необходимостью применять язык Java в компаниях, где они проходят производственную практику, могут активно освоить эту платформу и продемонстрировать свои знания в ИТ-компаниях. А это чрезвычайно важно, поскольку в это время в компаниях уже формируется мнение об этих студентах, как о потенциальных сотрудниках. И велика вероятность того,

что плохо успевающим, не проявляющим достаточного интереса к работе и не имеющим прогресса студентам руководители практики в ИТ-компании скажут «До свидания, Ваша производственная практика в нашей фирме для Вас закончена, мы не видим перспектив в Вашей дальнейшей работе у нас».

Здесь следует отметить, что прерогативы деканата и кафедры с одной стороны и руководства ИТ-компании с другой стороны, четко определены в соответствующих соглашениях, и представление на отчисление студента с производственной практики — это прерогатива руководства фирмы. Заметим, что факты таких отчислений четко коррелируют со средним баллом студента. Семилетний опыт проведения практики показал, что во всех случаях отчислений, сделанных до сих пор — а это, примерно, 20% от всего количества практикантов — средний балл студента всегда был меньше, чем 3,5. Были, конечно, и исключения, когда сильный студент уходил из компании после практики в связи с подготовкой к международной олимпиаде, с изменением научного интереса, с зарубежной стажировкой — но эти исключения лишь подтверждали полученный обобщенный результат.

Другие примеры. Для чтения спецкурса «Финансовая математика и финансовый менеджмент» приглашен профессор Экономического факультета СПбГУ С.В. Вавилов. Спецкурс «Метрология, стандартизация и качество разработки программного обеспечения: аспекты менеджмента» читает доцент В.И. Киев — заместитель директора НИИ информационных технологий Санкт-Петербургского государственного университета. Для чтения спецкурса «Теория фильтрации случайных процессов» пригласили профессора А.С. Матвеева — известного специалиста по теории оптимизации.

Мобильность профессорско-преподавательского состава, его высокая квалификация, положительная динамика развития учебных программ предполагает также мобильность, мотивированность и готовность к восприятию новых методов и идей со стороны студентов и аспирантов. Распределяясь на кафедру прикладной кибернетики в конце второго курса (а выбор кафедры для студента — дело добровольное) студенты должны быть готовы участвовать в интенсивной целевой образовательной программе. Наряду с выполнением общего учебного плана им приходится выполнять кафедральную часть нагрузки, которая более объемна, чем это рекомендовано Госстандартом.

Так, например, как уже отмечалось выше, в начале третьего курса дополнительно введена производственная ИТ-практика (один полный день в неделю — занятия и работа в ИТ-компании), вместо одного спецкурса по общему плану — два (по платформе Java), практические занятия (по Java), семинар по классической архитектуре и языкам программирования. Кроме того, на кафедре поощряется индивидуальная активность студентов и аспирантов: посещение внеплановых лекций и презентаций (например, ежегодных презентаций компаний Samsung-Electronics, Microsoft, Intel, Sun Microsystems/Oracle), участие в стажировках, выступления на российских и международных конференциях.

Далее мы остановимся на новых тенденциях глобальной мобильности, особенно активно проявляющихся в странах Европейского Союза и их влиянии на подготовку IT-менеджеров.

Одной из основных идей глобализации и мобильности в образовании является создание глобальной университетской образовательной сети с «системой кредитов». В рамках такого образования студент слушает лекции по определенному предмету в одном университете, сдает экзамены, получает за это определенное количество баллов (кредитов), потом перемещается в другой университет (возможно, в другой стране) там выбирает себе некоторое количество курсов, сдает их, также получает определенное количество баллов и так далее.

В результате, когда студентом набрана определенная сумма баллов, он получает возможность претендовать на диплом (т. е. некоторый сертификат о полученном образовании). Конечно, такая система образования сопровождается определенным количеством правил: выбранный студентом университет должен входить в упомянутую нами образовательную сеть, т. е. быть аккредитован в соответствующей организации, если студент претендует на некоторую специальность, например – математическую, он должен сдать определенное количество курсов математической направленности, может быть некоторая «выпускная» (дипломная) работа под руководством профессора и т. д.

Безусловно, для объединенной Европы эта новая образовательная система является шагом вперед с точки зрения интеграции. Молодые люди перемещаются по Европе, осваивают новые для себя языки и культурные ценности, сближаются и устанавливают дружеские отношения с местными студентами. Все это наряду с опытом учебы в разных университетах увеличивает студенческую мобильность на определенном, так сказать первом, начальном образовательном уровне.

Однако для организации второго, более серьезного и высокого уровня образования необходимы другие подходы. Здесь, конечно, следует отметить, что на первом уровне образования в глобальной сети естественным образом происходит определенная стандартизация и усреднение уровня образования. Поскольку в европейских университетах этот уровень в силу ряда причин неуклонно понижается, понижается и средний образовательный уровень во всей сети. Часто бывало так, что студенты математико-механического факультета на первом курсе добивались академического отпуска для обучения в довольно приличных европейских университетах. Там они были круглыми отличниками, а после возвращения на математико-механический факультет — круглыми двоечниками и отчислялись за академическую неуспеваемость, не преодолев учебной программы первого курса.

В результате, несколько лет назад Ученый совет математико-механического факультета принял решение о запрете студентам первого курса уез-

жать для учебы в другие университеты и об обязательном посещении занятий на первом курсе.

Этот пример, весь наш опыт общения с зарубежными коллегами, резко критические статьи академика В.И. Арнольда о падении уровня математического образования на Западе позволяют сделать вывод о том, что принятая сейчас в Европейском Союзе новая система глобализации образования является совершенно неприемлемой уже в начале интенсивного обучения по программам математико-механического факультета. Это касается описанного выше подхода к образованию на кафедре прикладной кибернетики, где необходимы предварительные знания, полученные на первых курсах, по основным математическим дисциплинам (математическому анализу, алгебре, геометрии и топологии, дифференциальным уравнениям, функциональному анализу и теории вероятностей), по основам информатики, математической логике и алгоритмическим языкам, по разным разделам прикладной математики: методам вычислений и теории управления. Кроме того, студенты кафедры, уже обладающие таким багажом общих знаний, должны быть готовы с большой скоростью осваивать новые актуальные дисциплины и применять многие из полученных знаний в производственной практике и в научной работе, которая позиционируется на кафедре также как один из необходимых элементов обучения.

Такая интенсивность обучения, конечно, не предполагает перемещение из одного университета в другой. Производственная практика, продолжающаяся, как правило, для успевающих студентов, в течение трех лет также не предполагает длительных отлучек. Единственно, что оказывается полезным (и рекомендуется нашим студентам) — это не очень длительные стажировки в компаниях и университетах мирового класса в течение одного — двух месяцев. На наш взгляд, очень полезно — особенно на старших курсах — участие наших студентов и аспирантов в семинарах, конференциях различного уровня (региональных, всероссийских, международных). И, что особенно важно, не только как слушателей, но и с докладами. Таким образом, приветствуя и поддерживая студенческую мобильность, описанную в начале нашей статьи, мы с определенной степенью скептицизма относимся к европейским тенденциям глобализации университетского образования в рамках создания глобальных университетских образовательных сетей.

Описанные здесь общие принципы создания новой образовательной программы на кафедре прикладной кибернетики позволяют разрешать многие серьезные проблемы, возникшие в последнее время в современном университетском образовании. Они позволяют сблизить традиционные академические тенденции университетского образования с реальными потребностями ИТ-компаний в специалистах нового типа, которые уверенно чувствуют себя на современном рынке высококвалифицированного интеллектуального труда.

Подчеркнем здесь, что основная цель настоящей работы — распространение не только информации об этих общих принципах, но (это, пожалуй,

самое важное) и описание имеющегося практического опыта реализации этих принципов. При этом, разумеется, существуют большие возможности применения таких принципов и нашего опыта к другим образовательным программам — особенно в области организации менеджмента в высокотехнологичных областях современного производства и в ИТ-областях.

Модели сотрудничества Санкт-Петербургского государственного университета с ИТ-компаниями

В настоящее время в Санкт-Петербургском государственном университете сложились две основные модели сотрудничества с ведущими ИТ-компаниями Санкт-Петербурга. Первая модель — слабосвязанная кооперация. Рассмотрим ее на примере взаимодействия с компанией Samsung Electronics. Модель достаточно традиционна, она не предполагает разработки новых общих курсов, специальных курсов и практикумов, она также не предполагает активного вовлечения сотрудников и ресурсов компании в учебный процесс Университета. Один-два раза в год менеджеры компании проводят презентации, на которых рассказывают о самой компании, её перспективах, академических программах и возможности для студентов стажироваться в южнокорейских университетах на магистерских программах. Отбор студентов и аспирантов на стажировку производится по конкурсу — по результатам собеседования и тестирования.

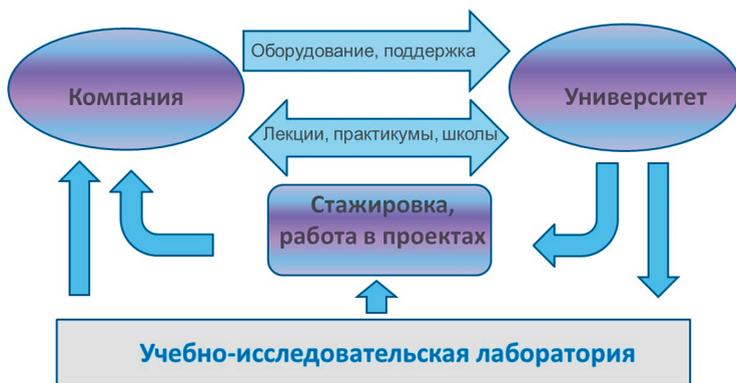
Требования к претендентам достаточно высокие — средний балл по сданным предметам учебного плана факультета должен быть не ниже 4.9. Очень приветствуется наличие практических компетенций. Компания полностью оплачивает перелет, проживание и обучение, во время которого студенты проходят обязательную практику на предприятиях компании и участвуют в реальных проектах, однако при этом выставляется довольно жесткое требование — после окончания обучения отработать в южнокорейских или российских подразделениях Samsung Electronics не менее двух лет. Отметим, что презентованный уровень обучения является достаточно высоким и отвечает современным требованиям подготовки ИТ-специалистов.

К этой же модели можно отнести мероприятия типа «дни компании в Университете», «дня открытых дверей компаний» и «ярмарки вакансий» регулярно проводимые различными компаниями Санкт-Петербурга, занимающимися разработкой программного обеспечения и созданием информационных систем и технологий, с целью заинтересовать выпускников математико-механического факультета условиями работы в компании. В течение одного дня представители презентуют свои компании, предоставляют студентам и аспирантам информацию об условиях найма и работы, призывают желающих пройти стажировку на льготных условиях, раздают сувениры с логотипами компаний и т. д. Такие разовые акции, безусловно, полезны, но

они не предполагают постоянного сотрудничества с компаниями в образовательном плане и не требуют изменения учебных программ.

Вторая модель — тесная кооперация. Исторически взаимодействие по такой модели было первоначально организовано с компанией Motorola российское подразделение которой существовало в Санкт-Петербурге с 1997 г. вплоть до реорганизации и разделения на три независимые компании в 2011 г. За это время сотрудниками компании стало более ста выпускников математико-механического факультета СПбГУ.

Общую схему взаимодействия можно представить следующим образом:



Практическое обучение студентов проводилось непосредственно в офисах компании — компания Motorola привлекала студентов и аспирантов непосредственно в реальные проекты, выполняемые в ее подразделениях. В процессе работы студенты проходили проектное обучение по принципу «от простого к сложному», постепенно набирая необходимые знания и опыт проектной работы, усваивая принципы корпоративной культуры. В конце каждого семестра руководители студенческих групп устраивали публичные отчеты в виде презентаций результатов выполненных работ по проектам. В 2007-2010 учебных годах был запущен 21 проект с участием студентов СПбГУ.

Рассмотрим другой пример реализации той же модели — сотрудничество с компанией Exigen Services. Это взаимодействие осуществляется на базе кафедры прикладной кибернетики Математико-механического факультета СПбГУ, однако оно с успехом может быть расширено и на другие кафедры.

Взаимодействие университета с Exigen Services началось в 2006 г. и явилось результатом совместной инициативы руководства Exigen Services и декана математико-механического факультета Г.А. Леонова. Главной стратегической целью этого взаимодействия является подготовка и адаптация

студентов ИТ-специальностей к последующей профессиональной деятельности в крупных международных ИТ-компаниях, работающих в области коммерческого бизнес-программирования — наиболее обширной и востребованной в настоящее время части программной инженерии.

Взаимодействие факультета с компанией Exigen Services строится на некоммерческой основе — вкладом со стороны компании является экспертиза в области самых современных технологий и методологий разработки ПО, используемых в индустрии, и знание современной корпоративной культуры, передаваемое студентам в рамках участия специалистов компании в учебном процессе; вкладом факультета является предоставление компании доступа «изнутри» к процессу обучения студентов, позволяющего осуществлять целенаправленный отбор отдельных студентов для последующей индивидуальной работы с ними. Как показывает опыт, такой подход позволяет, оставаясь в рамках университетского образовательного процесса, приблизиться к решению главной проблемы — соответствия профиля подготовки ИТ-специалистов реальным (и быстро меняющимся) потребностям индустрии. Кроме того, поскольку курсы лекций и практики Exigen Services начинаются, как правило, на третьем курсе (до этого студенты еще не имеют достаточной базовой подготовки для восприятия материала), реальные перспективы работы или стажировки в ИТ-компаниях (не обязательно Exigen Services) у них появляются к четвертому курсу, когда свободного времени (и, следовательно, возможностей безболезненно совмещать работу с учебой) становится намного больше. Ценность прошедших такую подготовку студентов для потенциальных работодателей оказывается намного выше, чем у «обычных» студентов, пытающихся в первый раз устроиться на работу в ИТ-компанию.

В рамках взаимодействия с университетом компания Exigen выстраивает сквозной непрерывный процесс подготовки студентов, направленный на последовательный отбор наиболее перспективных молодых специалистов и их адаптацию «в правильном направлении», т. е., в соответствии с профилем потребностей крупных ИТ компаний, занимающихся разработкой коммерческого ПО для глобального рынка. По сути, компания Exigen Services строит свою университетскую программу как «конвейер по подготовке кадров» (Рис.1).

На 3 и 4 курсах производится отбор студентов, заинтересованных в сотрудничестве с компанией. Как правило, отбираются студенты профильных специальностей с кафедр, выпускающих инженеров-программистов. Студенты получают темы курсовых проектов, вовлекаются в работу над учебными исследовательскими проектами, работая над которыми, они получают именно те знания, в которых заинтересована компания. Студенты старших курсов приглашаются на занятия ИТ-колледжа для углубления своих знаний и практических навыков или принимаются на работу на позиции инженеров-стажеров для участия в реальных проектах. Для увеличения количества

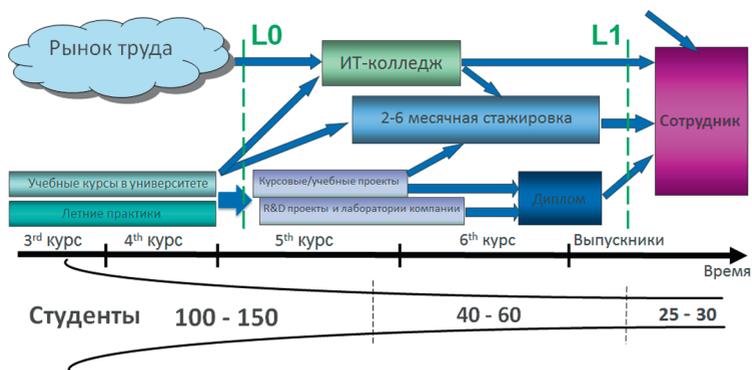


Рис. 1.

потенциальных кандидатов, на занятия ИТ-колледжа принимаются и студенты других специальностей: математики, физики, механики, электроники. Студенты, продемонстрировавшие лучшие результаты, лояльность к компании и интерес к тематике проектов, получают темы дипломных проектов и принимаются на работу при условии наличия открытых позиций.

При такой организации университетской программы, участие специалистов Exigen Services в учебном процессе осуществляется в следующих формах:

Чтение специалистами Exigen Services учебных курсов для студентов по промышленному программированию и тестированию ПО;

Консультационная помощь преподавателям Университета в виде «семинаров для преподавателей», позволяющих им лучше ориентироваться в современных передовых технологиях и программных продуктах;

Совместный анализ и обсуждение содержания учебных курсов относящихся к ИТ, читаемых в Университете в рамках обязательной программы обучения, которое проводится специалистами Exigen и Университета;

Проведение специалистами Exigen интенсивных тренингов и мастер-классов для заинтересованных студентов и преподавателей факультета, посвящённых углублённому изучению современных промышленных технологий и средств программирования;

Организация летних практик для студентов в компании;

Предоставление заинтересованным студентам возможности прохождения углублённой подготовки в рамках образовательного центра компании Exigen — ИТ-колледжа;

Привлечение студентов для участия в учебных проектах на базе университета;

Проведение стажировок (Internship): найм студентов по срочным договорам на позиции инженеров-стажеров для участия в реальных проектах

компании Exigen. Основной лекционный курс, который читается специалистами Exigen Services в рамках программы сотрудничества — это годового курса «Введение в бизнес-программирование на Java/J2EE» для студентов 3 курса кафедры теоретической кибернетики. Курс носит междисциплинарный, синтетический характер. Его главная задача — не только научить студентов использованию разнообразных технологий, входящих в J2EE, но и дать им живое ощущение тех реалий, с которыми они столкнутся, придя на работу в промышленные IT-компании, занимающиеся бизнес-программированием. Поэтому, наряду с изучением конкретных технологий, курс включает в себя элементы менеджмента проектов, организации командной работы, деловые коммуникации и ряда других дисциплин, относящихся к так называемым «soft skills», которые традиционно слабо представлены в университетском обучении, но в то же время жизненно необходимы молодым специалистам при работе над промышленными проектами разработки ПО для бизнеса.

Параллельно с чтением лекций, в рамках курса проводятся практические занятия и лабораторные работы, цель которых — дать студентам адекватное представление о практических аспектах применения изучаемых технологий на «приближенных к реальности» проектах.

Анализ и обсуждение содержания учебных курсов, семинары для преподавателей — регулярная практика в отношениях компании и университета, помогающая университету поддерживать учебные курсы в состоянии, соответствующем реальным требованиям индустрии производства ПО;

Тренинги и мастер-классы, проводимые специалистами Exigen Services для студентов и сотрудников университета представляют собой, как правило, адаптированные для студенческой аудитории варианты используемых в компании внутрифирменных тренингов. Тематика тренингов охватывает два основных направления: с одной стороны, это тренинги и мастер-классы, посвящённые современным технологиям, методологиям и инструментальным средствам разработки и тестирования ПО, которые активно используются в индустрии в настоящее время. Другое направление проводимых тренингов — это развитие надпрофессиональных навыков (soft skills), таких, как эффективная деловая коммуникация, навыки работы в команде, развитие креативности и нестандартного мышления и т. д. Как показывает опыт нашей работы, именно такие тренинги оказываются наиболее востребованы студентами.

Студенческие практики — это групповые практические занятия студентов в условиях, приближенных к условиям реальных проектов. Практики проводятся по двум направлениям — программирование на Java/J2EE и программирование на C#/ .NET.

Практики могут быть как частью учебных курсов, читаемых специалистами Exigen Services для студентов факультета, так и проводиться независимо. Как правило, группа в процессе практики выполняет один или

несколько учебных проектов, представляющих собой облегчённые версии фрагментов реальных бизнес-приложений. Студенты для участия в практиках отбираются университетом, и компания не производит никакого дополнительного тестирования или отбора. Практики проводятся в интенсивном режиме — две или три недели подряд, ежедневно, по 4–5 часов в день. Занятия со студентами обычно проводятся в первую половину дня, чтобы снизить нагрузку на компьютерные классы.

Основные требования к студентам, приходящим на практику — знание основ выбранного языка программирования (Java или C#), основ ООП, алгоритмов и структур данных. Программа каждой конкретной практики выстраивается в зависимости от общего уровня подготовки группы — варьируется степень сложности учебного проекта, а, кроме того, иногда в начале практики специалистам компании приходится проводить «кликбез» для повышения общего уровня программирования студентов.

По завершении практики отдельные студенты-участники могут получить приглашения на обучение в ИТ-колледже компании или на проходные стажировки (internship) по результатам которой кандидат может быть принят на работу в компанию. Проведение летних практик позволяют отбирать людей, персонально мотивированных и заинтересованных в дальнейшей работе именно в компании Exigen Services, организовать с ними дальнейшую индивидуальную работу.

Деятельность образовательного центра подготовки молодых специалистов Exigen Services — ИТ-колледжа — включает в себя обучение по трем дисциплинам:

- Программирование на Java/J2EE;
- Программирование на C#/.NET;
- Тестирование программного обеспечения.

Занятия в ИТ-колледже совмещают в себе теорию и практику. Основное внимание уделяется углублению и развитию уже имеющихся базовых знаний, а не освоению новых технологий; процесс обучения, таким образом, направлен не «вширь», а «вглубь»; методика обучения нацелена, прежде всего, на формирование практических навыков и приобретение опыта работы в условиях, максимально приближенных к реальности. Такой подход ближе к подходу бизнес-школ, чем к традиционному университетскому обучению. Курсы в ИТ-колледже, в зависимости от конкретной дисциплины, занимают 4–6 недель, занятия проводятся по вечерам (обычно с 16 до 20 часов), 2–4 дня в неделю.

Поступление на курсы в ИТ-колледж проходит на конкурсной основе, по результатам отбора и тестирования кандидатов. Отбор достаточно жёсткий, как правило, конкурс составляет не менее 5 человек на место. Основные требования к кандидатам на направления, связанные с программированием — это знание выбранного языка программирования (Java или C#), знание основ ООП, алгоритмов и структур данных, очень желательно наличие

опыта самостоятельного программирования и хорошее знание английского языка. При отборе кандидатов на обучение тестированию, проверяется общий уровень технической подготовки на уровне пользователя ПК, уровень знания английского языка, но основное внимание уделяется персональным качествам кандидата — чёткости мышления, аналитическим способностям, аккуратности и методичности. Существенным условием при прохождении отбора в ИТ-колледж является потенциальная возможность кандидата уделять достаточно времени работе в компании после окончания курсов (в случае получения приглашения на работу в Exigen Services) — 32–40 часов в неделю, в зависимости от дисциплины.

Обучение в ИТ-колледже осуществляется полностью за счет компании и не накладывает на учащихся никаких специальных обязательств по отношению к компании Exigen Services в отношении дальнейшей работы. По окончании курсов многие выпускники получают приглашения от Exigen Services на работу в компании в качестве стажеров (interns).

Участие в учебных проектах. Практическое обучение навыкам командной работы проводится на базе университета, когда компания формулирует темы небольших исследовательских проектов, а ответственные сотрудники университета формируют проектные команды и отслеживают ход выполнения работ. В процессе работы студенты проходят проектное обучение по принципу «от простого к сложному», постепенно набирая необходимые знания и опыт проектной работы, усваивая принципы корпоративной культуры. В конце каждого семестра руководители студенческих групп устраивают публичные отчеты в виде презентаций результатов выполненных работ по проектам.

Стажировка (Internship) в компании Exigen Services. Лучшие студенты, проявившие интерес к компании и выполняемым проектам, сумевшие продемонстрировать свои знания и способности, приглашаются на стажировки длительностью 3–5 месяцев. Со стажерами заключаются временные трудовые договора и они прикрепляются к реальным проектам. Каждому стажеру назначается наставник, который на протяжении первых дней знакомит его с проектом, знакомит его с используемыми технологиями и организацией работ, вводит его в курс дела. Затем стажеру поручаются небольшие самостоятельные задания, выполнение которых происходит под внимательным присмотром наставника. Постепенно стажер осваивает правила работы и, как правило, через несколько недель начинает работать полностью самостоятельно, становясь полноценным членом проектной команды. Хорошо зарекомендовавшие себя стажеры принимаются на постоянную работу в компанию при наличии свободных вакансий.

Отметим при этом, что льготы, получаемые студентами в процессе прохождения стажировки, весьма существенны: студентам выплачивается дифференцированная (в зависимости от достижений) добавка к стипендии, они имеют возможность заниматься на курсах английского языка (английский

язык является официальным языком компании, документация и переписка ведутся на английском языке). Лучшие студенты работают в летнее время на полную рабочую неделю, получая соответствующую заработную плату, а при наличии свободных вакансий могут приниматься на постоянную работу до защиты диплома или окончания аспирантуры.

Требования, предъявляемые к участникам, довольно высоки: использование английского языка на базовом уровне, знание реляционных баз данных (SQL), алгоритмических языков программирования высокого уровня C, C++, C#, Java, скриптовых языков PHP, Python, Perl, Shell, некоторые знания по аппаратному обеспечению, интерес к работе, усидчивость, трудолюбие, ответственность, коммуникабельность, желание работать в команде.

Такой подход является взаимовыгодным и для компании Exigen Services, и для университета и для региона. Компания получает специалистов по «ценам ниже среднего», университет обеспечивает своих студентов углубленными знаниями и практическими навыками, востребованными на рынке, а регион получает хорошо подготовленных специалистов, по тем или иным причинам не нанятых в компанию Exigen Services, удовлетворяя спрос других компаний, увеличивая предложение на рынке труда и, тем самым, снижая «перегрев» рынка ИТ-специалистов.

Экономическая выгода компании может быть проиллюстрирована следующим образом. Обучение 1 специалиста в ИТ-колледже обходится компании примерно в 20 000 руб. В среднем примерно 25 % обучаемых устраиваются работать в компанию. Таким образом стоимость подготовки одного нанятого специалиста равняется примерно 80 000 руб. Работая на позиции инженера-практиканта, студент получает заработную плату порядка 15 000 руб. Как правило, найм на постоянную работу происходит в среднем через 4 месяца, т. е. одному студенту за это время выплачивается порядка 60 000 руб заработной платы. При этом нанимается примерно 80 % от участников такой практики. Таким образом, стоимость одного нанятого увеличивается в 1,25 раза и равняется 75 000 руб. Такой молодой специалист, прошедший ту или иную форму «доучивания» имеет достаточную подготовку для выполнения реальных работ в проекте, он прошел адаптацию в компании и не тратит на это рабочее время. Как правило, такой специалист приходит на зарплату близкую к минимуму ставки младшего инженера порядка 30 000–35 000 руб. При найме сравнимого по знаниям специалиста с рынка, компания вынуждена предлагать заработную плату на уровне средней для этой категории (по исследованиям рекрутинговых агентств, порядка 40 000–45 000 руб). Уже за первый год работы экономия по заработной плате составляет примерно 120 000 руб, что полностью окупает затраты компании на подготовку специалистов и для себя и для региона.

Попробуем более точно посчитать экономическую эффективность университетской программы на примере мероприятий, проводимых компанией Exigen Services в 2011–13 годах в Санкт-Петербурге. Для подсчета эффек-

тивности воспользуемся данными по уровню заработных плат в ИТ компаниях представленными в отчете одного из агентств по найму персонала. Рассмотрим три уровня специалистов: L1 — Инженер 3-й категории (младший инженер), L2 — Инженер 2-й категории (инженер) и L3 — Инженер 1-й категории (старший инженер). В качестве средней зарплаты будем считать среднее значение зарплат разработчиков С.С++, .Net и Java и инженеров по тестированию. В качестве минимальной зарплаты возьмем среднее значение 10 перцентиль зарплат. Данные по минимальным и средним зарплатам приведены в таблицах 1 и 2.

Квалификация	Уровень зарплат по специальностям (средний), руб				Средний уровень для расчетов, руб
	Java	.NET/C#	C/C++	Test	
Стажер/техник (L0)	20 000	20 000	20 000	20 000	20 000
Инженер 3 кат (L1)	35 000	35 000	35 000	30 000	33 000
Инженер 2 кат (L2)	70 000	60 000	60 000	50 000	60 000
Инженер 1 кат (L3)	95 000	85 000	85 000	75 000	85 000

Таблица 1. Уровень зарплат по специальностям (средний)

Квалификация	Уровень зарплат по специальностям (10 перцентиль), руб				Минимальный уровень для расчетов, руб
	Java	.NET/C#	C/C++	Test	
Стажер/техник (L0)	10 000	10 000	10 000	10 000	10 000
Инженер 3 кат (L1)	25 000	25 000	25 000	25 000	25 000
Инженер 2 кат (L2)	45 000	50 000	45 000	30 000	43 000
Инженер 1 кат (L3)	65 000	70 000	65 000	50 000	62 000

Таблица 2. Уровень зарплат по специальностям (минимальный)

Также в качестве исходных данных будем использовать следующие:

Годовой бюджет университетской программы на чтение лекций в университете и проведение занятий в ИТ-колледже — 240 000 рублей (\$8,000)

Средняя продолжительность практики в реальных проектах (internship) — 4 месяца

Средняя заработная плата практиканта (intern) — 18 000 рублей (\$600)

Средний процент часов практикантов, перевыставляемых заказчиком — 10%

Приблизительный месячный рейт практиканта (цена, по которой он выставляется заказчиком для оплаты) — \$4,000

Уровень месячной зарплаты по категориям:

Минимальный L1 — \$800

Средний L1 — \$1 075

Минимальный L2 — \$1 350

Средний L2	— \$2000
Минимальный L3	— \$2100
Средний L3	— \$2800

Статистика по найму внешних кандидатов и по карьерному росту выпускников университетской программы показывает, что внешние кандидаты приходят, как правило, на зарплату не ниже средней. Выпускники университетской программы начинают с минимума L1 и при карьерном росте (аттестации на следующую категорию) также в большинстве случаев получают минимум следующей категории.

По результатам проведения летних практик, чтения спецкурсов, обучения в ИТ-колледже с охватом порядка 100 студентов мы отбирали группу инженеров-практикантов порядка 25 человек, большинство из которых после прохождения 4-х месячной практики нанималось на вакантные позиции в проекты. Из 25 вновь нанятых через 1 год, примерно 5 человек покидали компанию, 10 человек проходили аттестацию на следующую категорию (L2), а 10 оставшихся продолжали работу в качестве младших инженеров (категория L1). Еще через год 5 человек получали категорию L3, 4 человека покидали компанию (по 2 специалиста категорий L1 и L2) и 8 человек прошли аттестацию на L2. При этом предполагается, что все сотрудники, не получившие повышения, получают повышение зарплаты в 20%.

В таблице 3 содержатся сводные данные по расходам университетской программы и экономии заработной платы за счет найма ее выпускников. Возврат инвестиций, посчитанный как отношение экономии заработной платы принятых на работу выпускников университетской программы к расходам на саму программу, составляет более 300% за три года работы таких специалистов.

Такая модель взаимодействия факультета с компаниями лежит в русле общей системы образовательных программ, принятых в компании для взаимодействия с вузами, в ее создании и разработке существенное участие принимали преподаватели математико-механический факультет факультета. За годы существования программы сотни студентов прошли практику или стажировку в компании, прослушали специальные курсы на базе университета и в ИТ-колледже компании, а десятки лучших выпускников были приняты на работу.

Понятно, что на выходе этого непрерывного «конвейера» оказывается, как правило, больше подготовленных специалистов, чем компания Exigen может использовать в качестве сотрудников в данный момент — соответственно, предложения о работе в Exigen получают лучшие из выпускников, остальные вливаются в ИТ-рынок труда Санкт-Петербурга и, как показывает многолетний опыт, сразу же становятся востребованы другими ИТ-компаниями. Такая «избыточность» является одним из выражений социальной миссии компании, вносящей свой вклад в повышение качества подготовки молодежного ИТ-сообщества.

Статьи затрат (приход/расход)		Баланс 1-й год	Баланс 2-й год	Баланс 3-й год
Затраты на доучивание Студент → L0	Количество обучаемых	36		
	Затраты на доучивание	-250 000 р.		
4-х месячная стажировка L0 (стажер) → L1 (мл.инженер)	Затраты на зарплату	-1 800 000 р.		
	Оплачено заказчиками	1 200 000 р.		
Работа младших инженеров (L1)	Кол-во нанятых L1	25	10	0
	Кол-во уволившихся		5	2
	Экономия зарплаты L1	1 600 000 р.	432 000 р.	0 р.
Работа инженеров (L2)	Кол-во аттестованных L2		10	11
	Кол-во уволившихся			2
	Экономия зарплаты L2		2 448 000 р.	706 800 р.
Работа старших инженеров (L3)	Кол-во аттестованных L3			5
	Кол-во уволившихся			
	Экономия зарплаты L3			1 380 000 р.
Ежегодная экономия		1 600 000 р.	2 880 000 р.	2 086 800 р.
Общая экономия нарастающим итогом		1 600 000 р.	4 480 000 р.	6 566 800 р.
Возврат инвестиций (ROI)		36,59 %	177,07 %	278,87 %
Бюджет на отбор студентов (лекции, практики)			250 000 р.	
Средняя продолжительность стажировки (internship), мес.			4	
Процент часов стажеров, оплачиваемых заказчиками			10 %	
Приблизительный месячный рейт оплаты стажеров			120 000 р.	
Средняя зарплата стажера			18 000 р.	
Уровень зарплаты младшего инженера (L1)	Миним.		25 000 р.	
	Средний		33 000 р.	
Уровень зарплаты младшего инженера (L1)	Миним.		43 000 р.	
	Средний		60 000 р.	
Уровень зарплаты младшего инженера (L1)	Миним.		62 000 р.	
	Средний		85 000 р.	
Ежегодное повышение зарплаты без изменения категории			20 %	85 000

Таблица 3. Сводный баланс по расходам университетской программы

Описанная модель взаимодействия хорошо масштабируется и тиражируется и, на наш взгляд, является наиболее привлекательной при выстраивании отношений кафедры с новыми ИТ-компаниями с точки зрения соотношения затраченных ресурсов и получаемого результата. Именно по такой модели в 2013 году начало развиваться взаимодействие с крупной

петербургской компанией «Транзас» (с подразделением компании, которое занимается разработкой систем авионики для отечественных самолетов и вертолетов).

Обе описанные модели взаимодействия ориентированы на «штучную» подготовку студентов для ИТ-индустрии, и это естественно, поскольку при нынешней конфигурации рынка ИТ в России это — наиболее востребованный индустрией формат.

В то же время, при наличии соответствующего запроса со стороны индустрии, такое учебное заведение, как математико-механический факультет СПбГУ, могло бы обеспечить третью модель взаимодействия с индустрией — массовую специализированную подготовку студентов для нужд отечественного крупного бизнеса и государственных корпораций в формате сквозной подготовки целевых групп. Такие группы могли бы формироваться из студентов 3 курса и обучаться в течение 3 лет по специальной программе, ориентированной на потребности компании-заказчика. Такая программа, наряду с базовыми университетскими курсами, должна включать набор специализированных курсов, отражающих специфику деятельности компании. Возможна ситуация, когда некоторые из специальных курсов проводятся сотрудниками компании (аналогично тому, как это делается в рамках второй модели). Необходимым компонентом программы такого обучения должны быть регулярные производственные практики в подразделениях компании-заказчика, формирующие у студентов адекватный взгляд на промышленный производственный процесс, включая критерии успешности, правильности и должного качества выполнения работы, набор необходимых надпрофессиональных навыков (таких как деловая коммуникация, умение работать в команде, презентационные навыки) и т. д.

Опыт отечественных ИТ-компаний и центров разработки ПО крупных международных корпораций следует активнее внедрять в практику отечественных компаний и корпораций. В качестве потенциальных партнеров университета следует рассматривать в первую очередь компании, сопоставимые по размерам, скажем, с Роснано или крупными нефтяными компаниями. Авторы уверены, что именно этот путь даст реальные возможности массовой подготовки высококвалифицированных специалистов для модернизации отечественной промышленности, способных преодолеть «разрыв поколений», возникший в большинстве отраслей индустрии за последние 20 лет.

Литература

1. *Г.А. Леонов, В.И. Кияев, Н.В. Кузнецов, В.В. Оносовский, С.М. Селеджи*, Некоторые аспекты подготовки ИТ-менеджеров: специальные курсы и производственная практика – органически связанные части образовательного процесса, В книге Леонид Витальевич Канторович: математика, менеджмент, информатика (Под ред. Г.А. Леонова, В.С. Катькало, А.В. Бухвалова) СПб.: Изд-во Высшая школа менеджмента, 2010. стр. 297–528

2. *Leonov G.A., Kiyaev V.I., Kuznetsov N.V., Onosovsky V.V., Seledzhi, S.M.*, Computers and software engineering: Developing new models for educating mathematicians, in book *Computers in Education. Volume 2*, Nova Science Publishers, 2012 (ISBN: 978-1-62100-623-7), pp. 157 – 169
 3. *Abramovich S., Kuznetsov N., Kuznetsov S., Leonov G., Onosovsky V., Seledzhi S.*, Learning to develop and use software products: some common aspects of educational preparation of mathematicians and schoolteachers, *World Conference on Information Technology*, 2012 (Barcelona, Spain)
 4. *Abramovich S., Kuznetsov N.V., Kuznetsov S.V., Leonov G.A., Onosovsky V.V., Seledzhi S.M.*, Learning to develop and use software products: some common aspects of educational preparation of mathematicians and schoolteachers, *Procedia Technology*, 2013 [принята к печати]
-

Математические методы и алгоритмы в системах хранения данных высокой производительности



**Нестеров
Вячеслав Михайлович**

д.ф.-м.н., генеральный директор
Центра разработок компании EMC в Санкт-Петербурге



**Фёдоров
Андрей Рюрикович**

генеральный директор компании Digital Design

ИССЛЕДОВАНИЕ И ОПТИМИЗАЦИЯ АЛГОРИТМОВ УСТРАНЕНИЯ ИЗБЫТОЧНОСТИ В ЦИФРОВОМ ПРЕДСТАВЛЕНИИ ГЕНОМА ЧЕЛОВЕКА

Д.А. Драган

студент каф. 43, СПбГУАП

Целью данного доклада является рассмотрение методов и алгоритмов устранения избыточности в цифровом представлении генома человека. В докладе рассматриваются такие свойства данных человеческого генома как внутренняя и взаимная избыточность. Где под взаимной избыточностью понимается избыточность данных нескольких человеческих геномов, а под внутренней — избыточность данных единичного человеческого генома.

В рамках вопроса устранения внутренней избыточности, дается оценка эффективности применения к генетическим данным классических алгоритмов сжатия. Исследуется вопрос о максимально достижимой степени сжатия для данных такого рода.

В рамках вопроса устранения взаимной избыточности, исследуется вопрос сходства геномов различных людей. Дается теоретическая оценка эффективности устранения взаимной избыточности. Рассматриваются методы дедупликации применительно к данным человеческого генома

Конечной целью представленной исследовательской работы является создание системы хранения цифровых данных человеческого генома обладающая следующим функционалом:

- Устранение избыточности данных множества человеческих геномов.
- Поиск данных в базе данных по нечеткому образцу.
- Быстрый доступ к данным.

Критериями эффективности системы являются:

- Качество устранения избыточности (отношение размера «сырых» данных к размеру базы данных)
- Скорость обработки данных (устранение избыточности, поиск).
- Объем потребляемой памяти.

В целом в работе затрагиваются следующие вопросы:

1. Структура и свойства цифровых данных генома человека.
2. Внутренняя избыточность данных генома. Теоретический предел сжатия. Модель источника данных (источник с памятью или нет).
3. Взаимная избыточность данных генома. Задача дедупликации данных множества человеческих геномов. Алгоритмы и методы дедупликации применительно к генетическим данным.
4. Задача поиска данных по нечеткому образцу.

В процессе работы были исследованы различные подходы к устранению избыточности в данных человеческого генома. Применены и оценены алгоритмы сжатия и дедупликации. Проведено сравнение используемых алгоритмов и методов с существующими аналогами. На основании проведенных исследований можно сделать вывод о перспективности подходов основанных на методах дедупликации. Это объясняется большой взаимной избыточностью данных, в противоположность маленькой внутренней избыточности.

На данный момент работа носит скорее теоретический характер. Не смотря на полученные практические результаты, открытым остается вопрос о актуальности их практического применения.

Литература:

1. *Тарантул В.З.*, Геном человека: Энциклопедия, написанная четырьмя буквами // 2003 г.
 2. *J.R. Ecker et al.*, Genomics: ENCODE explained, Nature 489, pp. 52–55, 06 September 2012.
 3. PLOS BILOGY // The Diploid Genome Sequence of an Individual Human. URL: <http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/> (дата обращения 12.11.2013).
 4. Сайт nature biotechnology // статья «Compressive genomics using BLAST and BLAT» URL: <http://www.plosbiology.org/article/info:doi/10.1371/journal.pbio.0050254> (дата обращения 12.11.2013).
 5. Сайт nature biotechnology // статья «Compressive genomics» URL: <http://www.nature.com/nbt/journal/v30/n7/full/nbt.2241.html> (дата обращения 12.11.2013).
-

МЕНЕДЖЕР РАСПРЕДЕЛЕНИЯ ДОСТУПА К РЕСУРСАМ

Белокопытова Е.А.

студентка, elizaveta.bel@gmail.com

Фёдоров А.А.

студент, fedorov.andrei@icloud.com

Александрова О.В.

старший преподаватель

Аннотация: Проведение системных тестов, в случае систем хранения и репликации данных со сложной топологией, осложнено возможной занятостью устройств. Тесты могут менять конфигурацию устройств, а могут только следить за их работой. Происходит параллельное использование устройств, что ведёт к неконтролируемым результатам тестирования.

Для проведения тестирования в оптимальных условиях, имеет смысл рассматривать устройства составляющие данную систему, как набор ресурсов. Тесты, проводимые на устройствах, требует значительного времени исполнения. Тесты, можно рассматривать, как клиентов многопользовательской системы. Устройства в свою очередь можно рассматривать, как ресурсы. Так же важно заметить, что рассматриваемой системе устройств свойственна иерархическая структура — может быть запрошено устройство, которое в свою очередь состоит из нескольких. Данная система обычно содержит множество таких ресурсов, в ней, а также в любой многопользовательской системе с большим количеством доступных ресурсов, часто возникает проблемы корректной диспетчеризации. Необходимо нужно разрешать конфликты между конкурирующими клиентами, причем непрерывно, в автоматическом режиме, также должна быть возможность совместного использования ресурсов.

В связи с необходимостью совместного использования ресурсов, разрешения конфликтов между конкурирующими процессами, хранения и предоставления информации о текущих блокировках ресурсов было разработано приложение реализующее сервис управления доступом клиентов к ресурсам.

Система

Система, реализует:

1. Организацию доступа к ресурсам
2. Регистрацию и идентификацию клиентов в системе
3. Разрешение конфликтов между конкурирующими процессами

Для какого-либо взаимодействия с ресурсом клиент должен запросить доступ к ресурсу или группе ресурсов. Клиент одновременно может ре-

зервировать один или группу ресурсов (за один запрос), это обусловлено тем, что в данном проекте было решено использовать механизм предотвращения взаимных блокировок, а не их исправление. В связи с длительной работой тестов, время потраченное на исправление подобных ситуаций будет использовано не оптимально. Реализуется простая модель захвата, когда клиент не может запрашивать дополнительные ресурсы (иначе последовательность следующего вида: сохранение списка текущих занятых ресурсов, освобождение текущих ресурсов, добавление к сохранённому списку дополнительных ресурсов, запрос на резервирование обновлённого списка ресурсов).

Существует несколько возможных сценариев использования сервиса, каждый из которых определяет роль пользователя в случае возникновения конкурентной ситуации относительно ресурса или группы ресурсов.

Первый сценарий использования может быть следующим: клиент без авторизации или регистрации может получить доступ к любому незанятому ресурсу, в случае конкуренции с другим клиентом обладающим такими же правами, возможность использовать ресурс предоставляется клиенту сделавшему запрос первым. Такой клиент обозначается в системе как Regular.

Второй сценарий использования сервиса может быть следующим: клиент регистрируется в системе, закрепляя за собой право владения определёнными ресурсами или группой ресурсов. В случае если данный клиент запрашивает ресурсы, права на которые за ним закреплены, то такой клиент получает преимущество над клиентом из первого сценария. Но если такой клиент запрашивает группу ресурсов не закреплённую за ним, то он конкурирует с другими клиентами на общих правах неавторизованных или незарегистрированных пользователей. По отношению к закреплённым за таким клиентом ресурсам он будет обозначаться в системе как Priveleged.

При запросе ресурсов, Priveleged-пользователем, сервер может: подтвердить резервирование и предоставить ресурсы, подтвердить и попросить ожидать или отказать. В свою очередь, при запросе Regular-пользователем, либо предоставляют свободный ресурс, либо отказывают.

Также введены различные режимы резервирования ресурсов (Concurrent Read, Concurrent Write, Protected Read, Protected Write, Exclusive). Concurrent доступ предполагает совместный доступ. Protected — защищенный доступ, другие клиент могут только читать ресурс. Read/Write — намерение просматривать, просматривать и изменять соответственно. При эксклюзивном доступе, совместное использование невозможно.

Для удобства контроля серверная часть предполагает наличие веб-интерфейса. Сервер хранит в базе данных информацию, какие группы ресурсов закреплены за зарегистрированным клиентам, о резервировании ресурсов — тип резервирования и информация о клиенте. Веб-интерфейс позволяет посмотреть эти таблицы. Также дает возможность администрирования: изменение роли в системе, экстренное освобождение ресурса.

Заключение

К преимуществам данной систем можно отнести: предотвращение взаимных блокировок, долговременных простоев, структуризация доступа к совместным ресурсам, многофункциональный контроль за системой, веб-интерфейс облегчает работу администратора. Применимость в любых системах, где требуется управление ресурсами с иерархической структурой.

Для хранения данных используется базы данных PostgreSQL, для разработки приложения — фреймворк Spring, а также контейнер сервлетов Apache Tomcat, ORM Hibernate.

КОРПОРАТИВНАЯ СИСТЕМА ОРГАНИЗАЦИИ ИНФОРМАЦИИ И СОБЫТИЙ

Для большинства государственных учреждений и предприятий существует проблема информирования сотрудников о новостях и предстоящих событиях. Для этого, администрация использует либо свои информационные ресурсы, сторонние ресурсы, связь либо непосредственный личный контакт.

Существующие сервисы для решения таких проблем не учитывают специфику структуры организации, так как направлены, прежде всего, на использование сервиса отдельными группами людей, не имеющих привязки к месту работы, учебы и должностям.

Предлагаемое решение позволяет сотруднику создавать, просматривать и управлять событиями внутри корпоративной сети. В качестве примера работы сервиса была выбрана ИС Югорского государственного университета.

Система реализована с помощью технологий ASP .NET (C#) и MSsql. В составе имеет два основных элемента:

- Блок «Battleship» (*анг. Линкор*) — Ядро системы. CMS сайта. Отвечает за все манипуляции с данными.
- Блок «Sail» (*анг. Паруса*) — Интерфейсная оболочка.

И два дополнительных:

- Блок «Dock» (*анг. Пристань*) — Модуль отвечает за привязку сервиса к внешней системе, источнику данных о пользователях и их ролях, в нашем случае ИС ЮГУ.
- Блок «Mariner» (*анг. Матрос*) — Синтаксические анализаторы внешних информационных ресурсов для формирования списка событий, не принадлежащих корпоративной сети.

Так же в системе используется семантический анализатор, помогающий отличить зерна от плевел, при формировании внешних событий со сторонних информационных ресурсов синтаксическими анализаторами.

Особенностью предложенного решения является то, что разработанная система может использоваться, как интегрировано во внешний источник данных о пользователях, так и отдельно от него. В дополнение ко всему, модуль «Mariner» можно как использовать, так и отключить.

Технология написания сервиса позволяет адаптировать его под различные внешние источники данных, путем настройки блока «Dock» под определенную структуру организации.

РАЗРАБОТКА МОДУЛЯ ВЫЧИСЛЕНИЯ СИНДРОМОВ И ВОССТАНОВЛЕНИЯ УТРАЧЕННЫХ ДИСКОВ В RAID-МАССИВЕ С ИСПОЛЬЗОВАНИЕМ АРИФМЕТИКИ ПОЛЯ $GF(2^8)$ ¹

Зайберт В.С.

*студентка кафедры системного программирования СПбГУ,
gethappy90@gmail.com*

Научный руководитель:

Платонов С.М.

*руководитель исследовательской лаборатории RAIDIX,
platonov.s@raidixstorage.com*

Аннотация: В данной работе рассматривается эффективная реализация расчетов RAID с использованием арифметики конечных полей $GF(2^8)$. Рассмотрен метод с использованием параллельных вычислений и уменьшения числа операций. Проводился анализ возможности уменьшения размера кода с сохранением высокой производительности. Проведены тесты корректности и замеры производительности получившихся функций.

Введение

Системы хранения данных (СХД) это комплексное решение для хранения больших объемов информации, а так же быстрого и бесперебойного доступа к ней. Существует множество различных технологий, обеспечивающих выполнение таких требований. Одна из них RAID, что расшифровывается как Redundant Array of Independent Disks — «отказоустойчивый массив из независимых дисков», концепция которой состоит в объединении нескольких дисков для обеспечения отказоустойчивости. Кроме того, RAID распараллеливает процесс чтения и записи на все диски, что увеличивает скорость доступа к информации. В частности, в данной работе рассматривается RAID6, использующий для восстановления сбойных данных две разные контрольные суммы, называемые синдромами. Данная технология позволяет восстановить до двух отказавших дисков в дисковом массиве.

Задачей данной работы было реализовать эффективное вычисление этих синдромов при помощи кодов Рида-Соломона в поле $GF(2^8)$, восстановление дисков и сравнение с предыдущими результатами. Использование кодов Рида-Соломона позволит нам в будущем реализовать Silent Data Corruption, то есть восстановление данных в тех случаях, когда нам не известно место

¹ Работа выполнена по заказу компании RAIDIX.

сбоя [5, 6]. К тому же операции, позволяющие реализовать данный метод, не трудоемки.

Использование поля Галуа $GF(2^8)$ имеет ряд преимуществ:

- большая часть работ в этой области проводилась именно с этим полем. Таким образом, можно сравнить, выигрывает ли наше решение в скорости за счет непосредственно алгоритмов расчета, распараллеливания и способов представления информации;
- данное поле дает возможность обрабатывать до 255 дисков. Если брать поля меньшей размерности, то максимальное количество дисков резко падает. К примеру, поле $GF(2^4)$ позволяет использовать только 15 дисков;
- За счет сравнительно небольшого поля мы имеем возможность хранить некоторые, необходимые для работы алгоритма значения в заранее рассчитанных таблицах и эти таблицы будут иметь небольшой размер.

Формат данных

У нас имеется массив дисков. Каждый диск разбивается на блоки одинакового размера так, что размер блока является делителем размера диска. Все блоки с одинаковыми номерами образуют страйп (Stripe). Эти блоки будем обозначать как $D_0, D_1, D_2, \dots, D_{N-1}$, где N — количество дисков без учета дисков для хранения синдромов. Для каждого страйпа часть блоков выделена специально для хранения синдромов. Так как для RAID6 мы используем два синдрома, то необходимо выделить два блока. Не умаляя общности можно считать, что это последние блоки страйпа. Остальные блоки хранят исходную информацию.

Для вычисления контрольных сумм в RAID6 используются формулы вида:

$$P = D_{N-1} + D_{N-2} + \dots + D_0$$

$$Q = x^0 D_{N-1} + x^1 D_{N-2} + x^2 D_{N-3} + \dots + x^{N-1} D_0,$$

где x — примитивный элемент поля, например, $x = \{02\}$. [3]

Используя схему Хорнера, можно представить формулы таким образом, чтобы в них использовались только умножения на примитивный элемент поля и сложения:

$$Q = (((\dots D_0 \dots)x + D_{N-3})x + D_{N-2})x + D_{N-1}.$$

В качестве неприводимого многочлена в поле лучше всего брать такой, чтобы единиц в нем было как можно меньше. В этом случае уменьшится

количество операций при умножении на x [5]. В поле $\text{GF}(2^8)$ мы можем рассматривать многочлен $f = x^8 + x^4 + x^3 + x^2 + 1$. В нашем алгоритме распараллеливания и распределения данных у нас получилось умножать или складывать 256, при использовании AVX, и 128, при использовании SSE, элементов за то же количество операций, которое мы ранее тратили на один элемент.

Рассмотрим теперь формулу для восстановления данных (j и k — номера сбойных дисков, $k > j$):

$$D_k = P + \sum_{i=0, i \neq k}^{N-1} D_i$$

$$D_j = \frac{(P + \tilde{P}) + (Q + \tilde{Q})x^{-(n-k-1)}}{x^{k-j} + 1}.$$

Здесь появляется умножение на x в некоторой отрицательной степени и деление на некий элемент поля. Так как мы находимся в поле, то

$$x^{k-j} + 1 = x^i$$

при $k \neq j$, i мы сможем найти, используя предподсчитанные таблицы степеней примитивного элемента и логарифмов элементов нашего поля. Далее с помощью малой теоремы Ферма эту формулу можно свести всё к следующей:

$$D_j = ((P + \tilde{P}) + (Q + \tilde{Q}))x^{2^{8-1-(n-k-1)}}x^{2^{8-1-i}}.$$

Стоит отметить, что умножение на произвольный элемент поля так же сводится через факторизацию к умножениям на x и сложениям в зависимости от битов элемента, на который умножаем [4]:

$$a(x) * b(x) = (a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0)(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_0) =$$

$$= ((\dots(b_{n-1}a(x)x + b_{n-2}a(x))x + b_{n-3}a(x))x + \dots + b_1a(x))x + b_0a(x).$$

Есть два способа производить умножение:

1. Массив функций, где -ая функция умножает свой аргумент на x^i
2. Обобщенное умножение. Используя факторизацию, мы перемножаем два элемента.

Реализация

Для повышения эффективности кода мы старались уменьшить количество условных переходов и циклов, так как на них падает производи-

тельность про сравнению с линейным кодом [1,2]. Для этого для каждого количества дисков были написаны свои функции. Функции писались не вручную, их создавали код-генераторы. Получились довольно объемные файлы, в каждом из которых функции одного типа, но для разного количества дисков. В качестве эксперимента и для уменьшения размера модуля мы попробовали различные сворачивания их в циклы по 8. В конце работы приведены графики, на которых видна зависимость производительности от степени свернутости в циклы.

Для реализации нами был выбран язык C, а не ассемблер, так как первый имеет определенные плюсы:

1. Простота переносимости с одной архитектуры на другую. Нам нет нужды беспокоиться о том, что на разных архитектурах могут отличаться ассемблерные команды.
2. Возможность использования оптимизаций компилятора. Современные компиляторы предоставляют различные возможности оптимизации, позволяющие наиболее эффективно распределять память, работать с регистрами и преобразовывать код. [1,2]
3. Отсутствие необходимости распределять регистры. Для хранения каждого синдрома нам необходимо по 8 регистров. Сейчас, используя RAID6, нам нужно только два синдрома, но мы уже задействуем все 16 регистров. В дальнейшем планируется увеличивать количество синдромов, но тогда нам потребуется больше регистров, чем у нас есть. Возникает вопрос о порядке операций и грамотном распределении регистров. В предыдущем пункте указано, что этот вопрос одним из лучших способов сможет решить компилятор. [1,2]

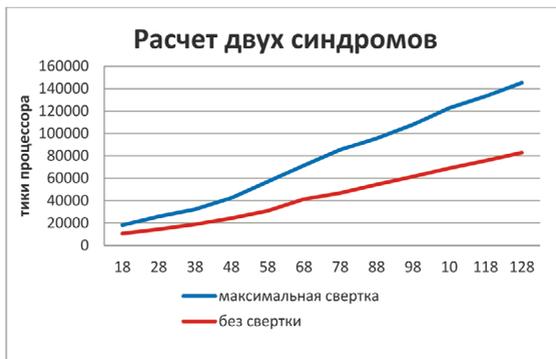
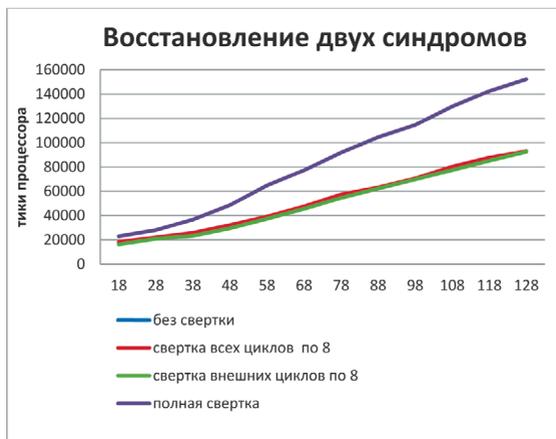
Выводы

Получилось реализовать распараллеливание вычислений и перераспределение данных в блоках, что позволило заметно ускорить алгоритмы обработки страйпа. Исследованы различные способы свертки циклов функций расчета синдромов и восстановления данных, что позволило уменьшить размер исходного кода, произведены сравнения этих реализаций.

Ниже приведены графики сравнения различных реализаций для восстановления двух синдромов. Измерения производились на следующей конфигурации:

- ОС: Debian 7.0
- Тип ОС: x64
- CPU: Intel® Xeon(R) CPU E5-2620 0 @ 2.00GHz × 18
- Оперативная память: 39.4 Гб

Код собирался gcc 4.7, оптимизация — O3.



В таблице показано сравнение производительности реализаций с развернутыми циклами и с различными способами свертки.

В итоге мы получили, что за счет оптимизации компилятора только один вид свертки дает отличные результаты от реализации без свертки. При восстановлении двух дисков максимальная свертка работает в среднем в два раза медленнее, но исходный код весит почти в семь раз меньше. При расчете двух синдромов максимальная свертка медленнее на 80%, а исходный код меньше в 5,3 раза. Нам хотелось уменьшить размер кода, но не ценой падения производительности в два раза.

Дальнейшие исследования

В ближайшее время хотелось бы попробовать исследовать различные опции компилятора, чтобы найти золотую середину между размером кода и производительностью.

В дальнейшем планируется реализовать исправление ошибок Silent Data Corruption, выполнить подсчет трех синдромов, что позволит нам восстанавливать до трех сбойных дисков в массиве. Сделать восстановление сбойных дисков с помощью обобщенного умножения и сравнить результаты.

Литература

1. *Richard Gerber, Aart J.C. Bik, Kevin B. Smith, Xinmin Tian* The Software Optimization Cookbook High-Performance Recipes for IA-32 Platform, Second Edition.
 2. *Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman*. Compilers: Principles, Techniques, & Tools, Second Edition.
 3. *H. Peter Anvin*. The mathematics of RAID-6. 2006–2011. <http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf>
 4. *Путерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.Мир. 1976.
 5. *Утешев А.Ю.* Поля Галуа. <http://pmpu.ru/vf4/gruppe/galois>
 6. *Утешев А.Ю.* Математика отказоустойчивых дисковых массивов. <http://pmpu.ru/vf4/codes/raid>
-

РАЗРАБОТКА МОДУЛЯ ВЫЧИСЛЕНИЯ СИНДРОМОВ И ВОССТАНОВЛЕНИЯ УТРАЧЕННЫХ ДИСКОВ В RAID-МАССИВЕ С ИСПОЛЬЗОВАНИЕМ АРИФМЕТИКИ ПОЛЯ $GF(2^{16})^1$

Демьяненко И.И.

*студент кафедры системного программирования СПбГУ,
diib@yandex.ru*

Савельев Н.Г.

*студент кафедры системного программирования СПбГУ,
viosng@gmail.com*

Платонов С.М.

*руководитель исследовательской лаборатории RAIDIX,
platonov.s@raidixstorage.com*

Санкт-Петербургский государственный университет

Аннотация: В данной работе рассматривается подход к вычислениям в RAID с использованием арифметики конечных полей $GF(2^{16})$, а так же схемы параллельных вычислений. Основной целью ставилась эффективная реализация вычислений в указанном конечном поле. Разработаны код-генераторы для генерации кода функций расчета двух синдромов и восстановления двух отказавших дисков. Проведены тесты корректности и замеры производительности получившихся функций, а так же сравнение с имеющимися аналогами.

Введение

В настоящий момент существует большой спрос на системы хранения данных (СХД). Так как объёмы данных, которые необходимо хранить, постоянно растут, в СХД используется технология RAID. Она не только увеличивает возможный объём хранилища, но также позволяет распараллелить процесс чтения-записи, что обеспечивает высокую скорость доступа к данным, а также предоставляет механизм отказоустойчивости благодаря контрольным суммам. Но из-за появления новых, более производительных дисковых устройств возникает необходимость в увеличении быстродействия алгоритмов работы с RAID-массивами. Цель нашей работы состояла в исследовании возможного прироста производительности при переходе с полей Галуа размера 2^8 на поля размера 2^{16} , а также применении параллельных вычислений с использованием векторных инструкций процессора.

¹ Работа выполнена по заказу компании RAIDIX.

Актуальность задачи

Реализованные алгоритмы для работы с RAID-массивами имеют следующие преимущества:

- **Увеличение максимального количества дисков по сравнению с полем $GF(2^8)$.** Большой размер элемента поля позволяет работать с 65535 дисками вместо 255. И поскольку задача расчёта и использования кодов Рида-Соломона важна не только для RAID, снятие этого ограничения даёт возможность использовать разработанные алгоритмы в других областях, где требуется помехоустойчивое кодирование.
- **Эффективность операции умножения на x по сравнению с непараллельной реализацией.** Благодаря эффективной реализации вычислений и использованию технологий SSE и AVX количество инструкций при одном умножении зависит только от количества единиц в двоичном представлении образующего элемента поля. В нашем случае, на умножение на x 128 или 256 элементов расходуется всего три процессорных инструкции.
- **Увеличение объёма данных, обрабатываемых за один вызов функции по сравнению с полем $GF(2^8)$.** Из-за удвоения размеров элементов поля за раз обрабатывается вдвое больший объём данных, используя для умножения на x то же число операций. Но существуют технические ограничения, такие как количество регистров и размер кэша процессора, которые не позволяют предсказать результат заранее.

Описание алгоритмов

Для расчёта синдромов используются следующие формулы:

$$S_0 = D_0 + D_1 + \dots + D_n$$

$$S_1 = x^{n-1} * D_0 + x^{n-2} * D_1 + \dots + D_n.$$

При этом вторая формула может быть представлена в виде

$$S_1 = x * (\dots (x * (x * (D_0) + D_1) + D_2) \dots) + D_n.$$

Как мы видим, все операции можно выразить через умножение на x и сложение. Прирост производительности ожидался за счёт повышения эффективности умножения на x , а также обработки большего участка памяти за один вызов функции расчёта.

Формулы, используемые для восстановления двух дисков с данными:

$$D_j = ((S_1 + \overline{S_1}) * x^{-(n-k-1)} + S_0 + \overline{S_0})(x^{k-j} + 1)^{-1}$$

$$D_k = S_0 + \overline{S_0} + D_j,$$

где n — количество дисков с данными, j и k — номера вышедших из строя дисков, а $\overline{S_0}$ и $\overline{S_1}$ — синдромы, рассчитанные без их учёта. Для восстановления были рассчитаны таблицы степеней x и $(x^{k-j} + 1)^{-1}$ для всевозможных $k-j$, а также написана функция умножения набора элементов на произвольный элемент поля. При каждом вызове функции восстановления страйпа ей передаются необходимые значения из таблицы, что позволяет поместить их в кэш при множественном вызове.

Используемые инструменты

- **Разработка велась на языке программирования C.** Если бы мы использовали язык ассемблера, было бы неясно, как использовать регистры SSE/AVX. Например, для расчёта двух синдромов в нашей схеме векторных вычислений требуется 32 регистра, при том, что процессор предоставляет всего 16. Мы считаем, что компилятор C использует регистры более эффективно в отношении скорости работы, чем распределение их вручную. К тому же, порядок использования регистров может быть самым разнообразным, что принесло бы трудности в случае реализации на языке ассемблера. Также плюсом C является переносимость кода на другие платформы.
- **Был использован генератор кода.** В связи с тем, что наличие в коде условных переходов негативно влияет на скорость его выполнения, было принято решение использовать отдельную функцию для каждого количества дисков в массиве (от 5 до 128) с последующим объединением их в массив. Генератор позволяет автоматизировать написание большого количества похожих функций. При этом значительно возрастает объём исполняемого файла, но также увеличивается и скорость выполнения алгоритмов.

Результаты

Каждая из функций запускалась 2000 раз, отбрасывались крайние 5% значений, вычислялся средний результат. На графиках приведено сравнение времени работы реализованных функций с аналогами, использующими другие алгоритмы, в условиях одинакового тестового окружения.

Конфигурация тестового стенда:

- ОС: Debian 7.0
- CPU: Intel® Xeon(R) CPU E5-2620 0 @ 2.00GHz × 18
- Оперативная память: 39.4 Гиб

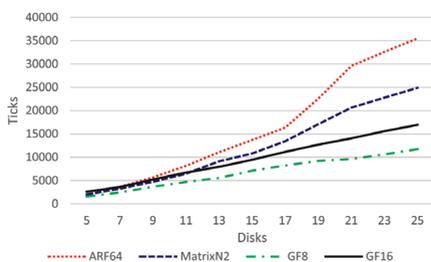


Рис. 1. Время расчёта двух синдромов для 5–25 дисков

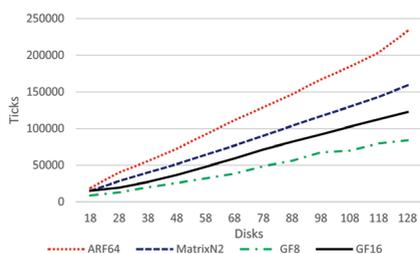


Рис. 2. Время расчёта двух синдромов для 18–128 дисков

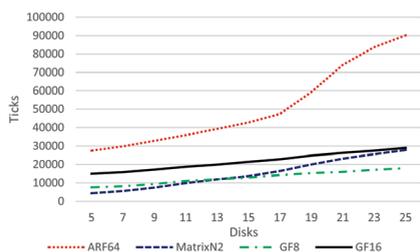


Рис. 3. Время восстановления двух дисков с данными для 5–25 дисков

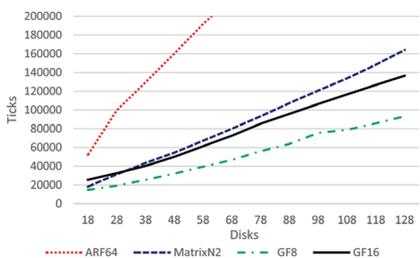


Рис. 4. Время восстановления двух дисков с данными для 18–28 дисков

Как мы видим, быстрдействие данного модуля в среднем выше, чем у предшествующих алгоритмов, но уступает аналогичному алгоритму для $GF(2^8)$. По нашему мнению, это связано с сильной нехваткой регистров и недостаточным размером кэша процессора, а следовательно, с большим количеством обращений к памяти.

Заключение

В данной работе рассмотрена реализация модуля работы с RAID-массивами на основе векторных вычислений в поле $GF(2^{16})$. Полученные результаты ниже ожидаемых. При этом не исключено, что в будущем появятся процессоры с большим объёмом кэша, на которых данный алгоритм проявит себя значительно лучше. Возможные направления дальнейших исследований — оценка производительности других функций (работа с тремя синдромами, выявление и устранение скрытых повреждений диска), а также эксперименты с $GF(2^b)$ при простых b , что позволит снизить количество операций при умножении на x до одной.

Литература

1. *Richard Gerber, Aart J.C. Bik, Kevin B. Smith, Xinmin Tian*, Second Edition. The Software Optimization Cookbook High-Performance Recipes for IA-32 Platform.
 2. *H. Peter Anvin*. The mathematics of RAID-6. 2006–2011. <http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf>
 3. *Путерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.Мир. 1976.
 4. *Утешев А.Ю.* Поля Галуа. <http://pmpu.ru/vf4/gruppe/galois>
 5. *Утешев А.Ю.* Математика отказоустойчивых дисковых массивов. <http://pmpu.ru/vf4/codes/raid>
-

РАЗРАБОТКА МОДУЛЯ ВЫЧИСЛЕНИЯ СИНДРОМОВ И ВОССТАНОВЛЕНИЯ УТРАЧЕННЫХ ДИСКОВ В RAID-МАССИВЕ С ИСПОЛЬЗОВАНИЕМ АРИФМЕТИКИ ПОЛЯ $GF(2^{256})$

Быкова Ю.С.

*студентка кафедры системного программирования СПбГУ,
yulya_byk@mail.ru*

Веселков И.Д.

*студент кафедры системного программирования СПбГУ,
veselkov.id@gmail.com*

Научный руководитель:

Платонов С.М.

*руководитель исследовательской лаборатории RAIDIX,
platonov.s@raidixstorage.com*

Введение

В данной работе рассматривается подход к вычислениям в RAID с использованием арифметики конечных полей $GF(2^{256})$. Использование такого поля позволяет реализовывать параллельную обработку большого числа элементов за очень малое число векторных операций, исследуется влияние технических ограничений на производительность алгоритма. Разработаны код-генераторы для генерации кода функций расчета двух синдромов и восстановления двух отказавших дисков. Проведены тесты корректности и замеры производительности получившихся функций.

Основные термины и технологии

Система Хранения Данных (СХД) — это комплекс, который позволяет надежно хранить большой объем данных и предоставляет к ним бесперебойный и быстрый доступ. В следствии поломки одного или нескольких жестких дисков, данные могут быть утрачены и для их восстановления используется RAID, «Избыточный массив независимых дисков». Вычисляются контрольные суммы, «синдромы» для которых требуется дополнительное дисковое пространство.

Страйп (Stripe), основная единица обработки данных в системе хранения, разбивается на буферы (Buffers) одинакового размера, обозначаемые $D_0, D_1, D_2, \dots, D_{N-1}$. Для организации вычислений буфера разбиваются на блоки (Blocks) одинакового размера. Размер блока является делителем размера буфера. Количество буферов N равно количеству дисков данных в

массиве. В рамках одного страйпа буфера могут называться дисками. Для обеспечения отказоустойчивости в страйп вводятся дополнительные буфера (диски), предназначенные для хранения синдромов. Размер этих дополнительных буферов (дисков) равен размеру буферов (дисков) данных.

Наиболее надежным и экономным по занимаемому дисковому пространству является структура RAID6, дисковый массив с чередованием, использующий две контрольные суммы, вычисляемые двумя независимыми способами, таким образом мы можем восстанавливать данные при выходе из строя до двух жестких дисков в дисковой группе. За счет чередования достигается параллельное выполнение процесса чтения записи на все диски, что обеспечивает высокую скорость доступа к данным.

Цель данной работы

С помощью кода Рида-Соломона эффективно реализовать вычисление синдромов и восстановление утраченных дисков с элементами в поле $GF(2^{256})$, а также сравнить производительность с полями размера 2^8 и 2^{16} .

Данная технология RAID при работе с элементами из поля $GF(2^{256})$ имеет несколько преимуществ:

- Эффективность операции умножения на x по сравнению с непараллельной реализацией
 За счет эффективной реализации вычислений и использованию технологий SSE и AVX при одном умножении количество инструкций совпадает с количеством единиц в двоичном представлении образующего многочлена поля. В нашем случае, на умножение на x большого числа элементов расходуется всего три процессорных инструкции.
- Увеличение максимального количества дисков по сравнению с полем $GF(2^8)$ и $GF(2^{16})$.
 Большой размер элемента поля позволяет работать с огромным количеством дисков, около $1.16 \cdot 10^{77}$, что примерно равно 0,12 процентам от количества всех атомов в обозримой Вселенной. Это дает нам возможность полностью пренебречь их количеством.
- Оптимальное использование процессорного кэша
 Цифра 256 была выбрана не случайно. Длина страйпа имеет вполне определенное значение в 4 Кбайта. Если мы разобьем его на переменные по 16 байт, чтобы можно было их обрабатывать с помощью технологии SSE, то получим как раз 256 переменных. Будем представлять их совокупность как многочлен поля $GF(2^{256})$.

Для вычисления контрольных сумм в RAID6 используются формулы вида

$$P = D_0 + D_1 + D_2 + \dots + D_{n-1}$$

$$Q = g_0 * D_0 + g_1 * D_1 + g_2 * D_2 + \dots + g_{n-1} * D_{n-1},$$

где g_i — это произвольный элемент поля.

Используя схему Хорнера, можно представить формулу вычисления Q так чтобы в ней использовались только операции умножения на примитивный элемент поля и сложения.

$$Q = D_0 + x * (D_1 + x * (D_2 + \dots + x * (D_{n-2} + x * D_{n-1}) \dots)).$$

Умножение на произвольный элемент поля так же сводится через факторизацию к умножению на x и сложениям в зависимости от битов элемента на который умножаешь

$$\begin{aligned} a(x) * b(x) &= \\ &= (a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0) * (b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0) = \\ &= b_0 * a(x) + x * (b_1 * a(x) + \dots + x * (b_{n-3} * a(x) + x * (b_{n-2} * a(x) + x * b_{n-1} * a(x)))) \dots, \end{aligned}$$

b_i — это биты элемента, на который производится умножение. То есть в зависимости от значения этих битов будут делаться умножения на x и сложения.

При восстановлении данных также появляется умножение на x^n .

Проделанная работа и инструментарий.

Был написан генератор кода для создания отдельных функций для каждого количества дисков (от 5 до 128) с последующим объединением их в массив. Это вызвано тем, что условные переходы понижают производительность по сравнению с линейным. Генератор позволил автоматизировать написание большого количества похожих функций.

Для написания генератора был выбран язык язык C, по следующим причинам:

1. Субъективно, реализовать на нём было проще, нежели мы бы пытались написать это на языке ассемблера.
2. При вычислении векторных операций нам важно в каком порядке используются регистры процессора. Компилятор берет работу по их распределению на себя, причем выполняет, конечно, оптимальнейшим способом.
3. В случае изменения архитектуры выполняющего процессора, нам будет достаточно переписать определяющие функции (сложение, умножение на примитивный элемент поля).
4. Оптимизация, которую нам предоставляет компилятор, по времени затраченному на написание кода и по времени, которое требуется на выполнение программы, является лучшей, чем мы бы реализовывали её самостоятельно.

Результаты

Ожидания

В теории, обработка целого страйпа будет проходить очень быстро. Например, умножение на x будет сильно зависеть от значащих единиц в образующем многочлене. Таким образом, мы стараемся выбрать многочлен с их наименьшим количеством. В нашем случае использовался многочлен $x^{256} + x^{10} + x^5 + x^2 + 1$. [5]

Полученный результат.

Ниже приведены результаты замеров работы функций (Диаграмма 1 и Диаграмма 2) для 48, 78, 108 и 128 дисков. Для сравнения приведены результаты замеров работы тех же функций из модулей, работающих с полями GF(2⁸) и GF(2¹⁶).

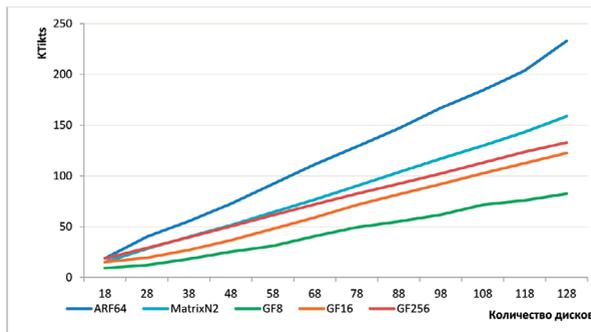


Диаграмма 1. Расчет двух синдромов

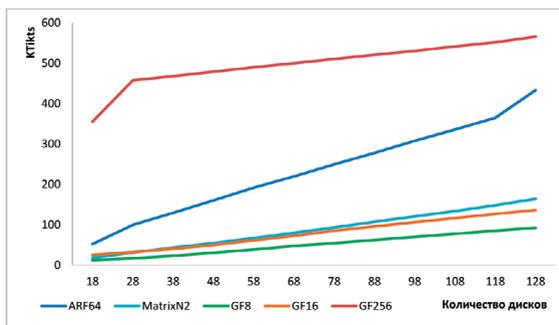


Диаграмма 2. Восстановление двух дисков.

Используемая конфигурация тестовой машины:

- ОС: Debian 7.0
- Тип ОС: x64
- CPU: Intel® Xeon(R) CPU E5-2620 0 @ 2.00GHz × 18
- Оперативная память: 39.4 Гб

Код собирался gcc 4.7, оптимизация — O3.

Как видно из таблиц, функции работают дольше, чем в других полях. Вызвано это гораздо большим обращением к памяти.

Планы на дальнейшую работу.

В дальнейшем, планируется написать функции для расчета трех синдромов и, соответственно, для восстановления трех дисков, а также функции для восстановления скрытых повреждений информации.

Также планируется рассмотреть работу функций в полях $GF(2^{64})$ и $GF(2^{128})$.

Литература.

1. *Путерсон У., Уэлдон Э.* «Коды, исправляющие ошибки»
 2. *H. Peter Anvin* «The mathematics of RAID-6»
 3. Сайт *Утешева Алексея Юрьевича* <http://pmpu.ru/vf4/users/au/index>
 4. *Richard Gerber, Aart J.C. Bik, Kevin B. Smith, Xinmin Tian* «The Software Optimization Cookbook High-Performance Recipes for IA-32 Platform» Second Edition.
 5. *Gadiel Seroussi* «Table of Low-Weight Binary Irreducible Polynomials»
-

СРАВНЕНИЕ АЛГОРИТМОВ ОБРАЩЕНИЯ ЭЛЕМЕНТОВ $GF(2^N)$ ¹

Щербаков А.В.

студент кафедры СП СПбГУ, candiduslynx@gmail.com

Научный руководитель:

Платонов С.М.

*руководитель исследовательской лаборатории RAIDIX,
Platonov.S@raidixstorage.com*

Аннотация: В работе рассматриваются алгоритмы обращения элементов конечного поля $GF(2^N)$. Данная задача возникает при восстановлении данных в RAID с числом контрольных сумм больше одного.

Актуальность такой задачи возрастает с ростом числа дисков в массиве, числа контрольных дисков, а так же с ростом размера поля. Производилось сравнение таких алгоритмов как возведение в степень, подсчёт континуанты, а также деление с использованием ганкелевых матриц.

Введение

Во многих системах хранения данных, использующих технологию RAID, расчёт контрольных сумм производится кодами Рида-Соломона с помощью арифметики конечных полей Галуа $GF(2^N)$. Использование этих полей удобно, так как позволяет работать при кодировании со словами-векторами из 0 и 1.

Расчёта контрольных сумм производится для того, чтобы можно было восстановить данные при отказе одного или нескольких дисков. В такой операции есть необходимость производить деление в рабочем поле Галуа. Обычно, используется лишь 2 контрольные суммы, и количество необходимых обратных элементов достаточно мало, чтобы быть рассчитанным заранее и использоваться в последствие при расчётах.

Однако, с ростом размера поля Галуа, количества дисков в RAID-массиве и дисков с контрольными суммами хранение таблиц обратных элементов становится невозможным. Дело в том, что в общем случае пришлось бы хранить таблицу обратных для всех элементов поля, а это (для $GF(2^N)$), в котором каждый элемент можно однозначно представить N -битным двоичным числом) составляет:

$$N * 2^N \text{ бит.}$$

При $N=32$ получаем 16 Гб, что тяжело хранить в памяти.

¹ Работа выполнена при поддержке компании RAIDIX.

Обращение элементов $GF(2^N)$

Как выглядит обратный элемент в $GF(2^N)$? $GF(2^N)$ — это поле, построенное из кольца многочленов $Z_2[x]$ по модулю неприводимого в $Z_2[x]$ многочлена f вида:

$$f(x) = x^N + p(x), \deg p(x) < N.$$

Тогда, для любого $a \in GF(2^N) \setminus \{0\}$ существует обратный элемент $a^{-1} \in GF(2^N) \setminus \{0\}$, такой что:

$$a * a^{-1} \equiv 1 \pmod{f}.$$

Обращение элементов $GF(2^N)$ возведением в степень

По умножению, $GF(2^N) \setminus \{0\}$ — это циклическая группа порядка $2^N - 1$, поэтому справедливо соотношение:

$$a^{-1} = a^{2^N - 1}.$$

Сложность этого метода составляет $O(N^2)$, так как возведение в такую степень можно проделать за $O(N)$ [1] и сложность умножения также составляет $O(N)$.

Обращение элементов $GF(2^N)$ подсчётом континуанты

Ещё один алгоритм обращения элементов в $GF(2^N)$ вытекает из следующего соображения: f — неприводим в $Z_2[x]$, а само $GF(2^N)$ — евклидово кольцо (то есть, можно делить с остатком), поэтому:

$$\forall a \in GF(2^N) \setminus \{0\}, \text{НОД}(a, f) = 1.$$

Если мы напишем линейное представление $\text{НОД}(a, f)$, получим:

$$1 = f * q + a * w \pmod{f}.$$

$$f * q \equiv 0 \pmod{f} \Rightarrow a * w \equiv 1 \pmod{f} \Rightarrow w = a^{-1}.$$

Если в процессе нахождения $\text{НОД}(a, f)$ получаем последовательность q_1, \dots, q_s — частные при делении с остатком — то w можно представить в виде:

$$w = K_s(q_1, \dots, q_s), \text{ где } K_s(q_1, \dots, q_s) \text{ — это континуанта [2].}$$

Для расчёта континуанты в данном случае удобно воспользоваться рекуррентным соотношением [3]:

$$K_{-1}=0, K_0=1,$$

$$K_{m+1}(x_1, \dots, x_m, x_{m+1})=x_{m+1} * K_m(x_1, \dots, x_m) + K_{m-1}(x_1, \dots, x_{m-1}).$$

Сложность этого алгоритма напрямую зависит от числа шагов нахождения НОД. Поэтому оценкой его скорости в самом плохом случае является $O(N^2)$.

Обращение элементов $GF(2^N)$ с использованием ганкелевых матриц

Ещё один метод, основанный на результате Кронекера по представлению результата полиномов посредством ганкелевой матрицы, предложен профессором А.Ю. Утешевым. Сам алгоритм сводится к выполнению следующих действий:

1. Из f получается набор $\{d_i\}$ — коэффициентов разложения $\frac{1}{f}$ в ряд Лорана;
2. Из $\{d_i\}$ и обращаемого элемента a получается последовательность $\{c_i\}_{i=1}^{2*(N-1)}$;
3. Эта последовательность подается на вход алгоритму Берлекампа-Мессис, на выходе получаем набор коэффициентов обратного элемента².

Сложность алгоритма в общем случае составляет $O(N^2)$. Но при хорошем f сложность относительно невысока, так как количество единиц в наборе $\{d_i\}$ оказывается мало (что облегчает работу остальных этапов).

Сравнение скорости работы алгоритмов

$N(GF(2^N))$	Возведение в степень	Подсчёт континуанты	Использование ганкелевых матриц
8	741	480	1181
16	2395	1481	2991
32	8929	5675	8191
64	35774	19169	22671
128	878762	158103	51344
256	5216242	1176736	182407

Таблица 1. Скорость работы алгоритмов обращения (в ts)

² Более подробно про первые 2 шага алгоритма можно прочитать в [4].

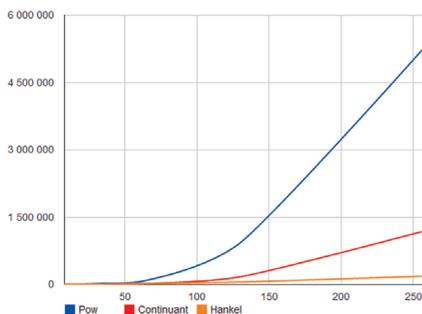


График 1.

Сравнение производительности алгоритмов обращения

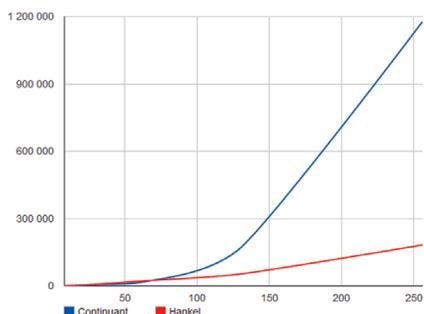


График 2.

Сравнение производительности алгоритмов 2 и 3

Замеры производились на компьютере со следующими характеристиками:

- Процессор — Intel® Xeon(R) CPU E5-2620 0 @ 2.00GHz × 18;
- ОЗУ — 39,4 ГиБ;
- ОС — Debian 7.0 ×64.

Заключение

Из графиков очевидно, что обращение континуантой выгодно использовать для полей с $N \leq 64$, а дальше — пользоваться последним алгоритмом. Дело в том, что, хоть при расчётах и можно использовать регистры длин 128 (SSE) или 256 (AVX), операция логического сдвига всего регистра на k разрядов пока что отсутствует, поэтому наблюдается резкое падение производительности на $N > 64$.

На данный момент последовательности $\{d_i\}$ и $\{c_i\}$ рассчитываются с помощью массивов. В дальнейшем планируется переписать последний алгоритм с помощью только арифметики поля Галуа и провести повторные замеры производительности с целью понять, стоит ли и при каких N использовать получившуюся реализацию, а при каких — исходную.

Литература

1. Кнут Д.Э. Искусство программирования. Т. 2: Полученные алгоритмы / Кнут Д.Э. // Издательство Вильямс — 2004. — С. 832
2. Лидл Р., Нидеррайтер Г. Конечные поля. Том 1. / Лидл Р., Нидеррайтер Г. // М.Мир. — 1988.
3. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. / Блейхут Р. // М. Мир. — 1989
4. Uteshev A.Yu., Cherkasov T.M. The search for the maximum of a polynomial. / Uteshev A.Yu., Cherkasov T.M. // J. Symbolic Computation. — 1998. — Vol. 25, № 5. — P. 587-618.

Информационная безопасность и технологии защиты информации



**Зикратов
Игорь Алексеевич**

д.т.н., профессор
заведующий кафедрой
безопасных информационных технологий СПбГУ ИТМО

ОЦЕНКА ПРИМЕНИМОСТИ МОДЕЛЕЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ДЛЯ РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ К ЗАДАЧЕ УПРАВЛЕНИЯ РОБОТОТЕХНИЧЕСКИМИ СИСТЕМАМИ

Басманов Д.Ю.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
basmanovdaniil@gmail.com*

Аннотация: В работе рассматриваются проблемы информационной безопасности в сфере робототехники, для решения которых могут быть применены наработки из области распределённых вычислений. Дается оценка их применимости, рассматриваются возможные проблемы.

Введение

Благодаря техническому прогрессу робототехнические системы становятся всё доступнее, что делает их более привлекательными для использования в автоматизации задач. Вместе с тем, как это часто бывает с новыми технологиями, информационная безопасность не рассматривается как важный фактор при проектировании подобных систем. Внедрение решений по обеспечению информационной безопасности постфактум или на завершающих этапах разработки зачастую требует переделки архитектуры системы и больших временных и денежных затрат. Ситуация осложняется тем, что на сегодняшний день не существует стандартов информационной безопасности для робототехнических систем. В то же время, робототехнические системы имеют ряд особенностей, которые не позволяют напрямую переложить опыт по защите информации из других областей. Данная работа нацелена на исследование этих противоречий и разработку их решения.

Проблемы безопасности

В современных многороботных системах и групповой робототехнике для принятия решений о дальнейших действиях роботов часто используют совещательные протоколы [1][2]. Это делается по разным причинам, в числе которых распределение нагрузки между роботами и сбор более полной информации о задаче. Тем не менее, такой подход потенциально позволяет злоумышленнику через одного скомпрометированного робота влиять на исход совещания и управлять группой роботов. Возьмём для примера группу роботов, случайным образом расположенных на плоскости, которым требуется найти центр масс своих местоположений и собраться в этой точке.

Простейшим решением будет вариант, в котором каждый робот по общему каналу связи вещает свои координаты и получает координаты остальных роботов, после чего рассчитывает центр масс и движется к получившейся точке. Примерно так задача сбора в точке и решается в действительности с вариациями алгоритмов по сложности, синхронности и другим параметрам. С точки зрения информационной безопасности у данного алгоритма есть несколько проблем. Во-первых, скомпрометированный робот может отправлять любые координаты, а значит может влиять на конечное положение центра масс, что в свою очередь означает управление группой роботов. Во-вторых, каждый робот узнаёт местоположение всех остальных роботов, что в случае с военными роботами может представлять тайну. Другая довольно частая операция в кооперативном поведении роботов — выбор временного лидера — тоже может привести к негативным последствиям, если реализована небезопасно. В свете изложенного, становится очевидна необходимость в обеспечении информационной безопасности принятия решений группой роботов.

Описание предлагаемого подхода

Сходные проблемы уже решались в сфере распределённых вычислений, кроме того, во некоторых работах коалиции роботов рассматривались с этой точки зрения[3][4]. При данном подходе каждый робот моделируется как мобильная вычислительная единица, которая имеет собственную локальную память и способна выполнять локальные вычисления. Тогда процесс принятия решения можно рассматривать как процесс вычисления функции в распределённой среде. С учётом этого, логичным шагом было бы пойти дальше и попытаться перенести имеющиеся наработки из области информационной безопасности распределённых вычислений в область робототехники.

Информационная безопасность в распределённых вычислениях имеет несколько аспектов. Один из них касается проблемы достижения верного решения группой совещающихся при возможном некорректном или умышленном вредоносном поведении части участников. В общем виде эту задачу называют проблемой византийских генералов или проблемой византийской отказоустойчивости. Существующие работы в области распределённых вычислений классифицируют проблему по наличию широковещания, пассивному или активному поведению проблемных участников, существованию сговора между ними и т. п. Разработанные протоколы позволяют достигнуть верного решения, если численность злоумышленников не превышает от половины до трети всех участников[5][6][7]. Применение данных методов могло бы принести пользу в групповой робототехнике, где вероятность нарушения протокола больше в силу количества участников.

Другой класс вопросов рассматривает возможность передачи вычислений недоверенной стороне с сохранением конфиденциальности исходных данных. Для решения этой задачи применяют гомоморфное шифрование и протоколы с нулевым разглашением. Подобные подходы могут быть применены для защиты протоколов принятия решений в робототехнических системах. Приведённая выше задача вычисления точки сбора роботов могла бы быть решена распределённо с сохранением конфиденциальности местоположения участников вычисления.

Конечно, существуют и ограничения применимости озвученных методов. В первую очередь слабым звеном является гомоморфное шифрование, рабочие алгоритмы которого несмотря на более чем тридцатилетнюю историю исследований появились лишь совсем недавно. Кроме того, их стойкость пока слабо изучена, а производительность оставляет желать лучшего. Также следует учитывать, что дополнительные меры безопасности могут потребовать увеличения вычислительной или коммуникативной нагрузки на роботов, что возможно не во всех областях робототехники. Тем не менее, для будущих исследований было бы полезно иметь набор целей, которые стоит учитывать при разработке, что позволило бы избежать проблемы позднего внедрения.

Заключение

В данной работе рассмотрены некоторые проблемы информационной безопасности возникающие в ситуациях кооперативного поведения роботов. Проведенная параллель между робототехническими системами и системами для распределённых вычислений, даёт основания для возможности переноса моделей информационной безопасности. На текущий момент применимость некоторых методов ограничена, но в перспективе они могут быть полезны.

Литература

1. Florian Rohrmueller, Dirk Wollherr, Martin Buss. MuRoCo: A Framework for Capability — and Situation-Aware Coalition Formation in Cooperative Multi-Robot Systems — 2012 — http://www.lsr.ei.tum.de/fileadmin/publications/Rohrmueller/rohrmueller_muroco.pdf
2. Lovekesh Vig, Julie A. Adams. A Framework for Multi-Robot Coalition Formation — 2005 — <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.9047&rep=rep1&type=pdf>
3. Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro. Distributed Computing by Mobile Robots: Gathering — 2012 — <http://sbrinz.di.unipi.it/~peppe/Articoli/Riviste/2012-SIAM.pdf>
4. Xavier Defago. Distributed Computing on the Move: From mobile computing to cooperative robotics and nanorobotics — 2012 — <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.8372&rep=rep1&type=pdf>

5. Martin Hirt, Jesper Buus Nielsen, Bartosz Przydatek. Cryptographic Asynchronous Multi-Party Computation with Optimal Resilience — 2005 — <ftp://ftp.inf.ethz.ch/pub/crypto/publications/HiNiPr05.pdf>
 6. Matthias Fitzi, Daniel Gottesman, Martin Hirt, Thomas Holenstein, Adam Smith. Detectable Byzantine Agreement Secure Against Faulty Majorities — 2002 — <ftp://ftp.inf.ethz.ch/pub/crypto/publications/FGHHS02.pdf>
 7. Zuzana Beerliova-Trubiniova, Martin Hirt, Jesper Buus Nielsen. On the Theoretical Gap Between Synchronous and Asynchronous MPC Protocols — 2010 — <ftp://ftp.inf.ethz.ch/pub/crypto/publications/BeHiNi10.pdf>
-

МЕТОДЫ РАСЧЕТА РИСКА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Ендовский А.С.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
dovski@gmail.com*

Научный руководитель:

И.А. Зикратов

д. т. н., профессор НИУ ИТМО, zikratov@cit.ifmo.ru

Аннотация: В статье рассмотрены методы расчета риска информационной безопасности, произведено сравнение с указанием критичных недостатков. Представлено предложение использование собственного метода оценки рисков.

Ключевые слова: риск, информационная система, информационная безопасность, метод расчета риска, оценка рисков, информационный актив.

Введение

Система управления рисками информационной безопасности (ИБ) является актуальной задачей на всех этапах работы комплекса защиты информации. При этом управлять рисками невозможно предварительно их не оценив, что в свою очередь должно быть произведено по какой-либо методике. На этапе оценки рисков наибольший интерес представляют непосредственно формулы и входные данные для расчета значения риска. В статье проанализировано несколько разных методов расчета риска и представлена собственная методика. Целью работы является вывод формулы расчета риска информационной безопасности, позволяющей получить массив актуальных рисков и оценить потери в денежном эквиваленте.

Риск информационной безопасности в классическом виде определяется как функция трёх переменных [6]:

- вероятность существования угрозы;
- вероятность существования уязвимости (незащищенности);
- потенциальное воздействие.

Если любая из этих переменных приближается к нулю, полный риск так же стремится к нулю.

Методы оценки рисков

ISO/IEC 27001. В отношении методологии расчета значения риска сказано: выбранная методология должна гарантировать, что оценки риска дают

сравнимые и воспроизводимые результаты. При этом в стандарте не приводится конкретной формулы расчета. [1]

NIST 800-30 предлагает классическую формулу расчета риска:

$$R=P(t)*S,$$

где R — значение риска;

$P(t)$ — вероятность реализации угрозы ИБ (применяется смесь качественной и количественной шкалы);

S — степень влияния угрозы на актив (цена актива в качественной шкале и количественной).

В итоге вычисляется значение риска в относительных единицах, которое можно ранжировать по степени значимости для процедуры управления рисками информационной безопасности. [5]

ГОСТ Р ИСО/МЭК ТО 13335-3-2007. Расчет риска, в отличие от стандарта NIST 800-30 происходит по трем факторам:

$$R=P(t)*P(v)*S,$$

где R — значение риска;

$P(t)$ — вероятность реализации угрозы ИБ;

$P(v)$ — вероятность наличия уязвимости;

S — ценность актива.

В качестве примера значений вероятностей $P(t)$ и $P(v)$ приведена качественная шкала с тремя уровнями: низкая, средняя и высокая. Для оценки значения ценности актива S представлены числовые значения в интервале от 0 до 4. Сопоставление им качественных значений должна произвести организация, в которой производится оценка рисков информационной безопасности. [2]

BS 7799. Уровень риска вычисляется с учетом трех показателей — ценности ресурса, уровня угрозы и степени уязвимости. С увеличением значений этих трех параметров риск возрастает, таким образом, формулу можно представить в следующем виде:

$$R=S*L(t)*L(v),$$

где R — значение риска;

S — ценность актива/ресурса;

$L(t)$ — уровень угрозы;
 $L(v)$ — уровень/степень уязвимости.

На практике вычисление рисков ИБ происходит по таблице позиционирования значений уровня угроз, степени вероятности использования уязвимости и стоимости актива. Значение риска может изменяться в диапазоне от 0 до 8, в результате по каждому активу получается список угроз с различными значениями риска. Так же стандарт предлагает шкалу ранжирования рисков: низкий (0–2), средний (3–5) и высокий (6–8), что позволяет определить наиболее критичные риски. [3]

СТО БР ИББС. Согласно стандарту оценка степени возможности реализации угрозы ИБ производится по качественно-количественной шкале, нереализуемая угроза — 0%, средняя — от 21% до 50% и т. д. Определение степени тяжести последствий для разных типов информационных активов так же предлагается оценивать с использованием качественно-количественной шкалы, т. е. минимальное — 0,5% от величины капитала банка, высокое — от 1,5% до 3% от величины капитала банка. [4]

Для выполнения качественной оценки рисков информационной безопасности используется таблица соответствия степени тяжести последствий и вероятности реализации угрозы. Если необходимо произвести количественную оценку, то формулу можно представить в виде:

$$R = P(v) * S,$$

где R — значение риска;

$P(v)$ — вероятность реализации угрозы ИБ;

S — ценность актива (степень тяжести последствий).

Предложенный метод

Рассмотрев все вышеперечисленные методы оценки рисков в части расчета значения риска информационной безопасности стоит отметить, что расчет риска производится с использованием значения угроз и ценность актива. Значительным недостатком является оценка стоимости активов (размер ущерба) в виде условных значений. Условные значения не имеют единиц измерения применимых в практике, в частности не являются денежным эквивалентом. В итоге это не дает реального представления уровня риска, который возможно перенести на реальные активы объекта защиты.

Таким образом предлагается разделить процедуру расчета риска на два этапа:

1. Вычисление значения технического риска.
2. Вычисление потенциального ущерба.

Под техническим риском понимается значение риска информационной безопасности состоящего из вероятностей реализации угроз и использования уязвимостей каждого компонента информационной инфраструктуры с учётом уровня их конфиденциальности, целостности и доступности. Для первого этапа имеем следующие 3 формулы:

$$R_c = K_c * P(T) * P(V),$$

$$R_i = K_i * P(T) * P(V),$$

$$R_a = K_a * P(T) * P(V),$$

где R_c — значение риска конфиденциальности;

R_i — значение риска целостности;

R_a — значение риска доступности;

K_c — коэффициент конфиденциальности информационного актива;

K_i — коэффициент целостности информационного актива;

K_a — коэффициент доступности информационного актива;

$P(T)$ — вероятность реализации угрозы;

$P(V)$ — вероятность использования уязвимости.

Применение данного алгоритма позволит произвести более детальную оценку риска, получив на выходе безразмерное значение вероятности возникновения риска компрометации каждого информационного актива в отдельности.

В последующем возможно вычисление значения ущерба, для этого используется усредненное значение риска каждого информационного актива и размер потенциальных потерь:

$$L = R_{cp} * S,$$

где L — значение ущерба;

R_{cp} — среднее значение риска;

S — потери (в денежном эквиваленте).

Предложенная методика позволяет корректно оценить значение риска информационной безопасности и скалькулировать денежные потери в случае возникновения инцидентов безопасности.

Литература

1. ISO/IEC 27001. Международный стандарт содержит требования в области информационной безопасности для создания, развития и поддержания системы менеджмента информационной безопасности. 2006. — 54 с.

2. ГОСТ Р ИСО/МЭК ТО 13335-3-2007. Национальный стандарт Российской Федерации. Методы и средства обеспечения безопасности. Часть 3. Методы менеджмента безопасности информационных технологий. Москва. 2007. — 76 с.
 3. BS 7799-2:2005 Спецификация системы управления информационной безопасностью. Англия. 2005. — 86 с.
 4. РС БР ИББС-2.2-200. Обеспечение информационной безопасности организаций банковской системы российской федерации. Методика оценки рисков нарушения информационной безопасности. Москва. 2009. — 23 с.
 5. Risk Management Guide for Information Technology Systems. Recommendations of the National Institute of Standards and Technology. США. 2002. — 56 с.
 6. Электронный источник Википедия, статья «Риск» [<http://ru.wikipedia.org/wiki/%D0%E8%F1%EA>].
-

СПОСОБЫ РЕАЛИЗАЦИИ SIDE CHANNEL АТАК И МЕТОДЫ ПРОТИВОДЕЙСТВИЯ ИМ

Калинин Д.А.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
kalinindal@gmail.com*

Аннотация: Данный документ содержит описание актуальных Side Channel атак на алгоритм Rijndael (AES) и возможные методы противодействия данным атакам.

Введение

Криптографические алгоритмы сегодня используются повсеместно, для поддержания целостности и конфиденциальности защищаемой информации, противодействию несанкционированному доступу к ней.

Криптоалгоритм Rijndael (AES) является блочным алгоритмом симметричного шифрования. Алгоритм широко применяется в различных программных и программно-аппаратных комплексах шифрования (к примеру OpenSSL, AES Crypt, PolarSSL, Scrypt), и является государственным стандартом шифрования США (стандарт FIPS 197). На текущий момент является защищенным от взлома посредством криптографического анализа, но уязвим для ряда атак по сторонним каналам, в силу недостатков конкретных реализаций криптоалгоритма, а также ряда фундаментальных уязвимостей.

Актуальными на сегодня можно назвать следующие виды атак:

- атаки на основе генерируемых ошибок;
- атаки на основе анализа времени выполнения с использованием кеш промахов.

Атака на основе генерируемых ошибок

Атака на основе генерируемых ошибок использует фундаментальную уязвимость алгоритма при его программно-аппаратных реализациях — итерационную структуру и отсутствие связи между раундовыми ключами.

Атака базируется на потенциальной возможности возникновения ошибки (как преднамеренной, так и специально внесенной) в процессе шифрования и дальнейшее использование ошибочных результатов работы для анализа и сравнения с валидными результатами, что позволяет получить секретный ключ. Получение ошибки возможно широким набором способов: изменение напряжения на входе шифратора, изменение частоты тактового генератора, внесение ошибки оптическим способом, посредством случайной или преднамеренной ошибки в работе аппаратных средств и/или программного обеспечения, использования различного специального ПО или

применение НДВ используемого. Одним из эффективнейших способов противодействия подобным атакам — добавление избыточности во время процесса шифрования посредством использования корректирующих кодов.

Атаки на основе времени выполнения

Атаки на основе анализа времени выполнения описывают методы и способы получения различных временных характеристик процесса шифрования. Эти характеристики используются для последующего анализа времени, которое затрачивается на исполнение отдельных операций криптоалгоритма и частоты промахов в кэш процессора. Также методики подобных атак описывают разнообразные способы воздействия на процесс шифрования для более эффективной атаки. Для подобного класса атак уязвимы современные мобильные операционные системы, например RichOS и среда виртуализации DalvikVM (Android OS), использующие Rijndael, как средство взаимной аутентификации процессов. Методами противодействия подобным атакам могут быть такие средства как: отказ от деления кеша между процессами, автоматическая загрузка таблиц в кеш, без ожидания запроса, косвенное управление процессом шифрования, для усложнения процесса управления им со стороны процесса-шпиона и получения процессом-шпионом необходимых для атаки данных.

Заключение

На текущий момент атака на основе генерируемых ошибок была успешно проведена для различных сочетаний размера исходного блока текста и секретного ключа и проверяется возможность защиты описанными выше способами для различных видов кодов и выбор оптимального вида кодирования.

Для атаки на основе времени выполнения на текущий момент проектируется система, реализующая озвученные выше методы противодействия.

Литература

1. *Калинин Д.А.* Практическая реализации Side Channel атаки на алгоритм Rijndael / Д.А. Калинин // Аннотированный сборник научно-исследовательских выпускных работ специалистов НИУ ИТМО. / СПб: НИУ ИТМО, 2012, с. 11–13



ОБЕСПЕЧЕНИЕ ЗАЩИТЫ ИНФОРМАЦИИ ДОПОЛНИТЕЛЬНЫХ УСЛУГ ОПЕРАТОРОВ СОТОВОЙ СВЯЗИ

Васильева Н.А.

*к. т. н. ассистент кафедры безопасных информационных технологий
СПб НИУ ИТМО nata-m@mail.ru*

Аннотация: В статье произведен обзор рынка услуг от сотовых операторов, произведен анализ законодательной базы, рассмотрены актуальные виды мошенничества при обращении к услугам сотовых операторов, а так же определены сложности защиты информации абонентов при обращении к услугам сотовых операторов связи.

Введение

Современные мобильные телефоны — это мобильные компьютеры, позволяющие хранить, передавать, принимать и обрабатывать аудио-, видео-, и текстовую информацию. Два неоспоримых факта делают вопросы защиты абонентских устройств и оборудования операторов связи все более актуальными: круглосуточная близость к абоненту и растущий список возможностей. Целью проведенной работы был анализ защищенности услуг сотовых операторов связи. Полученные результаты говорят об акцентировании операторов связи на вопросах защиты абонентов от мошеннических действий и отсутствии официально принятой научно-методической базы для решения комплексных задач защиты абонентов при обращении к услугам связи.

Решение

В настоящее время количество введенных операторами связи дополнительных услуг исчисляется сотнями, а с учетом количества существующих на рынке операторов связи возможно приближается к 1000. Услуги от сотовых операторов связи

Перечислим самые известные услуги от сотовых операторов:

- Передача SMS,
- Передача MMS,
- Видео- конференции,
- Мобильный Интернет,
- Короткие номера,
- Баланс,
- Теле- и радиовикторины и голосования,
- Яндекс-пробки,

- Мобильный банк,
- Обнаружение местоположения абонента (Услуга Радар от ОАО «Мегафон»), [1]
- Знакомства, а так же подборка самых свежих анекдотов, прогноз погоды, тесты, курсы валют, информация о портале, стоимость сервисов, управление подписками и др. [2]

Анализ законодательной и нормативной базы

Федеральный Закон № 126 «О связи» от 07.07.2003 (ред. от 03.12.2011)
Статья 2, пункт 33:

Услуга связи — деятельность по приему, обработке, хранению, передаче, доставке сообщений электросвязи или почтовых отправлений; [3]

Статья 53, пункт 1:

Сведения об абонентах и оказываемых им услугах связи, ставшие известными операторам связи в силу исполнения договора об оказании услуг связи, являются конфиденциальной информацией и подлежат защите в соответствии с законодательством Российской Федерации. [3]

К сведениям об абонентах относятся:

- фамилия, имя, отчество или псевдоним абонента-гражданина, наименование (фирменное наименование) абонента — юридического лица, фамилия, имя, отчество руководителя и работников этого юридического лица, а также
- адрес абонента или адрес установки оконечного оборудования,
- абонентские номера и другие данные, позволяющие идентифицировать абонента или его оконечное оборудование,
- сведения баз данных систем расчета за оказанные услуги связи, в том числе о соединениях, трафике и платежах абонента. [3]

Федеральный закон № 152 «О персональных данных» от 27.07.2006 г.

Статья 8, пункт 1.

В целях информационного обеспечения могут создаваться общедоступные источники персональных данных (в том числе справочники, адресные книги). В общедоступные источники персональных данных с письменного согласия субъекта персональных данных могут включаться его фамилия, имя, отчество, год и место рождения, адрес, абонентский номер, сведения о профессии и иные персональные данные, предоставленные субъектом персональных данных. [4]

Руководящий документ «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации от 30 марта 1992 г. и аналогичные ему выдвигают технические требования по защите, которые могут быть применены к оборудованию сотовых операторов.

ГОСТ Р ИСО/МЭК 15408-1,2,3-2008 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий» позволяет сформировать требования защиты информации практически к любому техническому средству. Стандарт может быть использован с целью формирования требований по защите информации к услугам сотовых операторов.

Виды мошенничества с использованием услуг сотовых операторов связи

Условно все современные виды мошенничества можно разделить на две группы:

- методы психологического воздействия на абонента (социальная инженерия).
- внедрение в абонентское устройство несанкционированного программного обеспечения или программного обеспечения с недекларированными возможностями.

Более подробно виды мошенничества описаны на тематических порталах сотовых операторов связи.

ОАО «Мегафон» разработал и поддерживает сайт «Безопасное общение» расположенный по адресу <http://stopfraud.megafon.ru/>.

Сотовый оператор «Билайн» разработал сайт «Безопасный Билайн» расположенный по адресу <http://safe.beeline.ru/>

Сотовая компания МТС разработала сайт «Безопасность — это просто», который находится по адресу <http://www.safety.mts.ru/>

Сотовый оператор Tele2 посвятил вопросам безопасности страницу «Мобильное мошенничество» — http://www.krasnodar.tele2.ru/help_warning_mobile.html, где даются советы по профилактики мошеннических действий и рекомендации по защите от них.

Виды мошенничества:

- Мошенничество с использованием коротких premium номеров SMS: Открытка «с сюрпризом», SMS-знакомства, «Изменились условия Вашего тарифного плана. Предложения псевдороботодателей Некорректное указание стоимости и количества отправляемых SMS, Отсутствие заявленного на ресурсе сервиса.
- Мошенничество с использованием голосовых premium номеров: «Позвони мне, позвони...»
- Мошенничество на доверии: Телефон «на один звонок», Клонирование, Жертва вымогательства, Дилерские нарушения.
- Внедрение вредоносного программного обеспечения: Порнобаннеры, Блокираторы и шифровальщики, Sms-отправители и «порнозвонилки», Псевдоантивирусы, «Задания работодателей».

- Мошеннический спам с целью вымогательства: Sms-просьбы, «Выигрыши», «Нигерийские письма» и псевдоакции, «Ошибочные» платежи, Звонки «Техподдержки Оператора», Звонки «из банка». [1]

Рекомендации по защите абонентов от мошеннических действий:

- Не давайте телефон незнакомым людям.
- Уточняйте стоимость перед звонком или отправкой SMS на короткий номер.
- Не сообщайте посторонним людям номера карт оплаты.
- Не выполняйте никаких инструкций, полученных от незнакомых людей по телефону
- Не сообщайте посторонним людям личную информацию. [2]
- Устанавливайте на компьютер хорошо зарекомендовавшие себя антивирусные программы и межсетевые экраны (Firewall) и своевременно их обновляйте.
- При скачивании контента внимательно читайте Условия использования сервиса, а также информацию, размещенную с символом «звездочка» (*).
- Не устанавливайте сомнительное программное обеспечение на свой компьютер / мобильный телефон.
- Не открывайте и не запускайте (*.exe) вложенные файлы неизвестного происхождения.
- Будьте осторожны при всплывающих окнах, не переходите по неизвестным ссылкам.
- Не отправляйте SMS для разблокировки Windows и разархивирования файлов.
- Пользуйтесь услугами: «Мобильный прайс», «Блокировка отправки сообщений на короткие номера». [1]

Особенности обеспечения защиты информации услуг операторов сотовой связи

Ответственность за защиту сведений об абоненте и оказываемых ему услугах связи возлагается на оператора связи и застройщика при строительстве средств связи и сооружений связи (ФЗ «о Связи» Статья 7 п.2).

Разработчиками («застройщиками») услуг связи являются как зарубежные (Nokia, Huawei, HP и др.), так и отечественные компании, специализирующиеся на разработки непосредственно дополнительных услуг (Протей, Беркут, Комплит и др.)

Касперский, Dr.Web, McAfee, Green Head предлагают решения для защиты абонентских мобильных устройств.

Федеральный орган исполнительной власти в области связи устанавливает требования защиты сетей связи от несанкционированного доступа

к ним и передаваемой по ним информации (ФЗ «о Связи» Статья 12, п.2; Статья 26, п.3.3) по согласованию с Федеральным органом исполнительной власти в области обеспечения безопасности (ФЗ «о Связи» Статья 12, п. 2.1).

Минкомсвязью разработаны несколько отраслевых нормативных документов:

- Концепция защиты персональных данных в информационных системах персональных данных оператора связи.
- Модель угроз и нарушителя безопасности персональных данных, обрабатываемых в специальных информационных системах персональных данных отрасли.
- Модель угроз и нарушителя безопасности персональных данных, обрабатываемых в типовых информационных системах персональных данных отрасли.

Так же в открытом доступе в Интернете можно найти Концепцию и Модель угроз разработанные Инфокоммуникационным союзом непосредственно для операторов связи.

Сложности обеспечения защиты информации (сведений) абонентов при обращении к услугам сотового оператора:

- Постоянно вводятся новые сервисы и заканчивают свое существование экономически не целесообразные или временные.
- Часть сервисов является встроенными в оборудование сотового оператора связи.
- Часть сервисов являются программными продуктами и используют аппаратно-программные комплексы оператора сотовой связи, при чем каждая новая услуга задействует только ей присущий набор технических средств.
- Часть сервисов использует только каналы сотового оператора связи и находится в эксплуатации либо у заказчика, либо в аутсорсинге и следовательно использует для своей реализации аппаратно-программные комплексы не принадлежащие оператору
- Никто не знает, какой дополнительная услуга будет завтра, в чем будет потребность рынка, частного и корпоративного клиент.

Заключение

Для корректного проведения работ по защите информации дополнительных услуг операторов связи, очевидно, требуется научно-методического подход. Разработанная методика должна учитывать и постоянно-меняющуюся инфраструктуру, и отдельную аттестацию крупных аппаратно-программных комплексов оператора сотовой связи и различную логическую инфраструктуру для каждой конкретно взятой услуги и многое другое.

Литература

1. Официальный сайт ОАО «Мегафон» «Безопасное общение». Лицензии №№ 15002, 41541, 41542, 42688, 54741. [Электронный ресурс]. URL: <http://stopfraud.megafon.ru/> (дата обращения: 31.01.13)
 2. *К.Ю. Смирнов*. «Мобильное мошенничество». СМИ «Tele2 (Теле2)». Свидетельство о регистрации Эл №ФС 77-36815 от 03 июля 2009. [Электронный ресурс]. URL: http://www.krasnodar.tele2.ru/help_warning_mobile.html (дата обращения: 31.01.13)
 3. Федеральный Закон № 126-ФЗ «О связи» от 07.07.2003 (ред. от 03.12.2011).
 4. Федеральный закон № 152-ФЗ «О персональных данных» от 27.07.2006 г.
-

SIDE-CHANNEL АТАКИ. ВЗЛОМ COMP128

Коровкин М.Г.

*аспирант кафедры, безопасных информационных технологий
СПб НИУ ИТМО, mikhail.korovkin@gmail.com*

Левина А.Б.

*доцент кафедры Безопасных Информационных Технологий НИУ ИТМО,
levina@cit.ifmo.ru*

Аннотация: В данной статье будет представлена атака по сторонним каналам на алгоритм аутентификации в сетях GSM. Будут рассмотрены различные варианты ее улучшения с помощью комбинации нескольких атак по сторонним каналам.

Введение

В последние 20 лет активно развивается новое направление криптоанализа, называемое Side Channel Attacks (SCA), или атаки по побочным каналам [1–9]. Основная идея данного подхода заключается в том, что шифрующее устройство рассматривается не только как математический аппарат, но и как конкретная его реализация на практике. Классический криптоанализ рассматривает криптоалгоритм с точки зрения математики — как некоторую функцию от входных данных, на выходе которой зашифрованный текст. Новая концепция рассматривает криптоалгоритм вместе с его материальной реализацией, обладающей определенными физическими свойствами, такими как время выполнения алгоритма, потребляемая при шифровании мощность, электромагнитное излучение от шифрующего устройства и другие.

В настоящее время SCA являются более результативным вариантом криптоанализа, нежели его классический вариант. С развитием SCA многие известные реализации используемых алгоритмов шифрования были взломаны, что побудило криптографов к созданию мер защиты от данной угрозы.

В данной статье будет рассмотрена одна из таких атак по побочным каналам — распределенная атака на COMP128 (алгоритм аутентификации в GSM-стандарте).

Распределенная атака на COMP128

В мае 2002 года сотрудники IBM Watson Research Center Josyula R. Rao, Pankaj Rohatgi и Helmut Scherzer совместно с Stephane Tinguely из Swiss Federal Institute of Technology опубликовали статью «Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards» (Распределенные атаки или как быстро клонировать GSM-карты), в которой рассказали, как можно реали-

зывать атаку по потребляемой мощности (Simple Power Analysis — SPA) на SIM-карту [6]. Данная атака позволяет получать секретный ключ всего за пару минут. При этом авторы использовали около 1000 случайных запросов и доказали, что можно использовать всего 255 заранее подготовленных запросов (атака с выбором открытого текста). Более того, атаку можно еще усовершенствовать до 8 приспособляющихся запросов, что позволяет проводить вскрытие секретного ключа за несколько секунд. Предыдущая известная атака на COMP128 — BGW (Briceno, Goldberg, Wagner — по фамилиям авторов) — требовала примерно 150 000 запросов.

Авторы распределенной атаки воспользовались тем фактом, что на первом раунде шифрования используется замена по таблице T0, которая содержит 512 элементов. Т.е. для индексации по ней необходимо использовать 9-битовые значения, в то время как в SIM-карте используется лишь 8-битовая архитектура. Тогда авторы предположили, что таблица должна быть разбита на 2 подтаблицы размером по 256 элементов. Анализируя энергопотребление SIM-карты при различных запросах, исследователи смогли определить, к какой части таблицы T0 был адресован запрос. Таким образом, измеряя энергопотребление запросов при изменении входных данных, им удалось вычислить секретный ключ.

Комбинированные атаки

Атаки по побочным каналам имеют большое многообразие. Как уже упоминалось ранее, можно использовать такие побочные каналы, как время шифрования [1], мощность потребления [2], электромагнитное излучение от шифратора [3, 4]. Помимо этого можно использовать яркость света, излучаемого монитором и отраженного от стены [5]. Можно достать секретную информацию даже по звукам, издаваемым внутренними компонентами электронного шифратора [7]. Существуют также различные атаки, которые воздействуют на шифратор и создают в нем ошибки, по которым потом восстанавливается ключ [8, 9].

В связи с этим многообразием возникла идея о возможности проведения комбинированной атаки по побочным каналам. Суть заключается в том, чтобы скомбинировать несколько атак для того, чтобы вскрыть какой-то алгоритм быстрее, чем это может сделать каждая из атак по отдельности. Также возможно сочетать несколько атак для получения большего количества секретных сведений, это даст результат, лучший, чем если бы использовалась одна из атак или обе, но по отдельности.

Варианты комбинированных атак на COMP128

Распределенная атака на COMP128, разработанная в 2002 году, является простой атакой по потребляемой мощности (SPA). Чтобы улучшить ее ре-

зультаты, нужно скомбинировать эту атаку с какой-то другой. Рассмотрим атаку по времени (ТА), атаку на основе зондирования (РА) и атаку на основе генерируемых ошибок (FIA).

Комбинация с атакой по времени

Комбинация с атакой по времени была бы неплохим вариантом, поскольку и атака по потребляемой мощности, и атака по времени являются неинвазивными и пассивными. Это очень удобно, поскольку в этом сочетании не требуется дополнительных активных воздействий, а лишь «прослушивание» шифрующего устройства. И обнаружить такую атаку после ее совершения было бы крайне трудно, поскольку обе ее составляющие никак не воздействуют на SIM-карту.

Однако на алгоритм COMP128 такую атаку провести довольно сложно, поскольку в нем используются операции, не подверженные атаке по времени.

Комбинация с атакой на основе зондирования

Комбинация с атакой на основе зондирования теряет те преимущества, которые имела комбинация атаки по потребляемой мощности и атаки по времени, поскольку зондирование является инвазивной атакой, т.е. потребуются «вскрывать» SIM-карту. Соответственно и обнаружить такую атаку после ее осуществления становится проще. Однако, зондирование дает куда большую свободу в выборе анализируемых данных, поскольку с его помощью можно наблюдать за практически любым местом шифрующего алгоритма.

В COMP128 можно наблюдать за последними 16 байтами на каждом из 8-ми раундов, а не только на первом. Это даст выигрыш в скорости в 8 раз.

Комбинация с атакой на основе генерируемых ошибок

Комбинация с атакой на основе генерируемых ошибок является чем-то средним между двумя предыдущими вариантами комбинированных атак. С одной стороны, данное сочетание уже нельзя назвать пассивной атакой, как комбинацию с атакой по времени, поскольку генерация ошибок является ярко выраженным примером активной атаки. Это дает возможность чуть-чуть изменять ход алгоритма, вносить туда какие-то ошибки, предоставляя более широкий спектр данных для анализа по потребляемой мощности. С другой стороны, генерация не такой мощный инструмент, как зондирование, поэтому ее применение не столь разносторонне и охватывающее. В то же время атака с генерацией ошибок, как правило, не требует столь дорогого оборудования.

Аналогично предыдущей атаке, хотелось бы вносить ошибку в начале каждого раунда в последние 16 байт. COMP128 записывает в эти 16

байт перемешанные биты, полученные на предыдущем раунде шифрования. Если бы удалось внести ошибку, которая позволяла бы записывать в последние 16 байт нужные нам значения, то и замеры по потребляемой мощности удалось бы проводить не только на первом раунде шифрования. Например, если бы с каждым раундом в последние 16 байт записывалось значение RAND, увеличенное на 1, то количество необходимых запросов бы сократилось в 8 раз.

Заключение

В данной статье была рассмотрена распределенная атака на COMP128 и предложены методы ее усовершенствования с помощью различных комбинированных атак по сторонним каналам. Можно сделать вывод, что атаки по сторонним каналам являются очень сильным инструментом криптоанализа.

Следует отметить, что противодействие комбинированным side channel атакам состоит в том, чтобы противодействовать составляющим атакам. Это является основным недостатком комбинированных SCA, поскольку сложность защиты от них равна сложности защиты от одной из составляющих атак, причем именно той, которой легче всего противодействовать.

Литература

1. *Paul C. Kocher*. Timing attacks on implementations of Diffie-Hellmann, RSA, DSS, and other systems // *Advances in Cryptology — CRYPTO '96* : сборник. — Springer, 1996. — С. 104–113 <http://www.cryptology.com/public/pdf/TimingAttacks.pdf> [дата просмотра: 02.04.2013]
2. *Paul Kocher, Joshua Jaffe, Benjamin Jun*. Differential Power Analysis // *Proc. of Advances in Cryptology (CRYPTO '99)*, LNCS : сборник. — 1999 — С. 388–397 <http://www.cryptology.com/public/pdf/DPA.pdf> [дата просмотра: 02.04.2013]
3. *Jean-Jacques Quisquater and David Samyde*. ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards // *E-SMART '01 Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security* : сборник. — Springer-Verlag, 2001 — С. 200–210
4. *Karine Gandolfi, D. Naccache, C. Paar, Karine G., Christophe Mourtel, Francis Olivier*. Electromagnetic Analysis: Concrete Results // *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems* : сборник. — Springer-Verlag, 2001. — С. 251–261 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.5990&rep=rep1&type=pdf> [дата просмотра: 02.04.2013]
5. *Kuhn, M.G.* Optical time-domain eavesdropping risks of CRT displays // *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on* : сборник. — 2002. — С. 3–18. <http://www.cl.cam.ac.uk/~mgk25/ieec02-optical.pdf> [дата просмотра: 02.04.2013]
6. *J. Rao, P. Rohatgi, H. Scherzer, S. Tinguely*, Partitioning Attack: Or How to Rapidly Clone Some GSM Cards // *IEEE Symposium on Security and Privacy* — 2002

7. *Adi Shamir, Eran Tromer* Acoustic cryptanalysis: On nosy people and noisy machines — 2011 <http://tau.ac.il/~tromer/acoustic/> [дата просмотра: 02.04.2013]
 8. *D. Boneh, R.A. DeMillo, and R.J. Lipton*. On the importance of checking cryptographic protocols for faults // *Advances in Cryptology — EUROCRYPT '97* : сборник. — Springer, 1997. — С. 37–51. <http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.48.9764&rep=rep1&type=pdf> [дата просмотра: 02.04.2013]
 9. *Eli Biham and Adi Shamir*. Differential Fault Analysis of Secret Key Cryptosystems // *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '97)*: сборник. — Springer-Verlag, 1997. — С. 513–525.
-

ОЦЕНКА ТОНАЛЬНОСТИ ТЕКСТОВОЙ ИНФОРМАЦИИ, АНАЛИЗ И СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ

Красильников А.Е.

*магистрант кафедры безопасных информационных технологий
СПб НИУ ИТМО, krasilnikov_ae@live.com*

Научный руководитель:

Комаров И.И.

доцент, к. ф.-м. н. НИУ ИТМО, i_krov@mail.ru

*Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Кафедра безопасных информационных технологий*

Аннотация: Задача анализа тональности текстовой информации, генерируемой в сети Интернет, является частью задачи обеспечения информационной безопасности. Она крайне важна для успешной работы любых организаций и структур. Для решения задач, связанных с выявлением и дальнейшим анализом эмоционально окрашенной лексики в тексте, используются методы, общее название которых — анализ тональности текстовой информации (Sentiment Analysis). Другое название данной области на русском языке — «анализ эмоциональной окраски текста». Анализ тональности текстовой информации входит в область задач компьютерной лингвистики и является подзадачей получения и обработки информации.

Введение

Зачастую мнение пользователей о объекте сопровождается его оценкой по заданной шкале. Поэтому в большинстве исследований определение полярности рассматривается как задача классификации текста. Классами могут быть оценки, поставленные автором обзора, а признаки извлекаются из текста мнения: часто в качестве признаков выбирают n -граммы слов и их части речи. Для решения задачи классификации используются различные методы машинного обучения. В работе сравниваются: метод с использованием лингвистического словаря и алгоритмы обучения с учителем (наивный байесовский классификатор и метод максимальной энтропии).

Использованные методы определения тональности

Анализ тональности текста, реализуемый с помощью тонального словаря, состоит из нескольких этапов. Сначала отработывает отдельный лингвистический модуль, автоматически производящий анализ текста, лемматиза-

цию всей лексики, отношения между словами. Затем слова размечаются по заранее подготовленным словарным спискам тональной лексики. Каждому слову приписывается значение, указывающие на силу тональности. Если слово не нашлось в списках тональной лексики, то оно считается нейтральным. Анализируется каждое слово в предложении, затем идет построение биграмм, происходит анализ тональности предложения с учетом полученных данных, тональность предложения равняется среднему арифметическому тональностей его составляющих.

Наивный байесовский классификатор — вероятностный классификатор, основанный на теореме Байеса и (наивном) предположении о статистической независимости случайных величин.

$$P(C|F_1, \dots, F_2) = \frac{p(C)p(F_1, \dots, F_2|C)}{p(F_1, \dots, F_2)}.$$

Основное достоинство данного классификатора заключается в низкой вычислительной сложности, а также в оптимальности, при условии действительной независимости признаков.

Метод максимальной энтропии — вероятностным распределением с максимальной энтропией является распределение Гиббса:

$$P(x_j|I) = \frac{1}{Z(\lambda_1, \dots, \lambda_m)} \exp \left[\sum_{i=1}^m \lambda_i f_i(x_j) \right].$$

Из заданной коллекции документов произвольным образом выделяется обучающее множество (некоторое небольшое, относительно размеров всей коллекции, количество документов, которые размечаются вручную; на основе этих данных подбираются коэффициенты для характеристических функций в функциях распределения).

Обучающее множество разбивается на токены с которыми ведется дальнейшая работа. В данной задаче токены — слова. Для каждого токена определяется состояние, к которому он относится.

Для обучения и работы методов был использован тональный словарь проекта AFINN.

Оценка результатов работы алгоритмов

Традиционно эффективность задачи классификации текста формулируется в терминах точности и полноты.

Точность — отношение числа правильно отнесенных текстов к определенному классу, к числу всех текстов, отнесенных к этому классу.

Полнота — отношение числа правильно отнесенных текстов класса 1 к числу текстов класса выбранного в коллекции.

Для оценки эффективности работы реализованных алгоритмов был проведен анализ выборки, состоящей из 60 текстов различной тематики. Для сравнения с существующими решениями, были использованы открытые проекты Python NLTK и анализатор проекта Brandlisten. В основе работы обоих анализаторов лежит Наивный байесовский классификатор.

	Точность	Полнота
Python NLTK	0.9	1
Brandlisten	0.25	0.25
Метод, использующий тональный словарь	0.6	0.6
Наивный байесовский классификатор	0,8	0,9
Метод максимальной энтропии	0,8	0,9

Заключение

Исходя из результатов можно сделать вывод: эффективность работы анализатора с использованием наивный байесовского классификатора (и метода максимальной энтропии) выше чем у метода, основанного только на использовании тонального словаря, однако ниже чем у классификатора в проекте Python NLTK. Добиться повышения эффективности работы созданного анализатора возможно увеличением обучающей выборки, а также добавлением правил анализа текстовой информации.

Литература

1. *Pang B. & Lee L.* Opinion Mining and Sentiment Analysis // Foundations and Trends in Information Retrieval, v.2 n.1–2, January, 2008 — pp.1–135.
 2. *Bing Liu.* Sentiment Analysis and Subjectivity // Handbook of Natural Language Processing, Second Edition, 2010.
 3. *Frantzi, K., Ananiadou, S. and Mima, H.* Automatic recognition of multi-word terms.// International Journal of Digital Libraries 3(2), pp.117–132., 2000.
-

МЕТОД ПОВЫШЕНИЯ ТОЧНОСТИ ЧАСТЕРЕЧНОЙ КЛАССИФИКАЦИИ СЛОВ В МОРФОЛОГИЧЕСКИХ АНАЛИЗАТОРАХ DLP-СИСТЕМ

Лапшин С.В.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
sv.lapshin@gmail.com*

Аннотация: В статье предложен метод повышения точности частеречной классификации слов предложения в морфологических анализаторах DLP-систем. Предложенный метод обладает важным преимуществом – для его работы не требуются предварительно размеченные тексты или иные специально подготовленные данные, за исключением морфологических словарей.

Введение

Ключевой момент в работе DLP системы – получение вердикта анализатора данных в ответ на запрос: является ли перемещаемая информация защищаемой? [1] Одним из важных этапов в работе анализатора является морфологический анализ передаваемых сообщений. От качества морфологических характеристик разобранных словоформ зависят показатели полноты и точности определения угроз информационной безопасности DLP-системы. Однако автоматический анализ естественного языка, в том числе и морфологический, небезошибочен и многозначен — часто он дает несколько вариантов анализа для одной языковой единицы. [2]

В статье описан метод повышения точности автоматического определения частей речи слов в предложении. Важной особенностью метода является то, что он не требует предварительно размеченных текстов или иных специально подготовленных данных, за исключением словарей морфологических описаний слов.

Приводимый в статье метод предназначен для текстовой информации, представленной на русском языке.

Суть метода

Предлагаемый метод разметки слов предложений по частям речи основывается на использовании определенного множества известных, заранее корректных последовательностей признаков части речи (множество корректных шаблонов) для аналогичных предложений. Аналогичным в данном случае является предложение, имеющее одинаковую длину и последовательность частей речи с анализируемым. Выбор одной из гипотез определяется наличием ее во множестве известных. При этом, если в известном мно-

жестве содержится более одной гипотезы для анализируемого предложения, то выбирается наиболее часто встречающаяся.

Предлагаемый метод состоит из следующих частей:

1. Формирование множества корректных шаблонов предложений.
2. Формирование множества гипотез о признаках части речи слов в предложении.
3. Поиск во множестве корректных шаблонов каждой из гипотез.
4. Выделение одной из гипотез на основе результатов поиска.

Формализация задачи представлена следующим образом.

Пусть $S = \{s_i\}$, $i = 1, \dots, n$ — множество корректных шаблонов предложений.

Пусть $H = \{h_i\}$, $i = 1, \dots, m$ — множество гипотез о признаках части речи слов анализируемого предложения.

Пусть R — правильное описание анализируемого предложения с точки зрения признаков части речи его слов.

Тогда решаемая задача состоит в том, чтобы выбрать такое $h_j \in H$, что $h_j = R$.

При этом важными подзадачами являются формирование множества корректных шаблонов S и поиск h_i в множестве S .

Формирование множества корректных шаблонов предложений

Сформировать достаточное множество корректных шаблонов вручную не представляется возможным. Если рассматривать 14 частей речи и только предложения длиной до 10 слов, то мощность множества S , необходимого для анализа таких предложений, можно грубо оценить по следующей формуле:

$$|S_{10}| = \chi 14^i \approx 3 \cdot 10^{11}. \quad (1)$$

Очевидно, что корректно разметить такое или сравнимое с таким число предложений за разумное время невозможно. Автоматическое решение этой подзадачи «напрямую» сталкивается с исходной решаемой проблемой — с проблемой точности автоматического определения частей речи в предложениях.

Для решения этой подзадачи была использована следующая идея. Искомое множество S можно формировать на основе предложений, состоящих только из однозначных с точки зрения признака части речи слов.

Таким образом, процесс формирования множества корректных шаблонов предложений состоит из следующих шагов:

1. Формирование морфологического словаря без омонимов.
2. Определение предложений, состоящих только из полученных на шаге 1 слов.

3. Создание шаблонов на основе полученных на шаге 2 предложений.
4. Добавление полученных шаблонов в искомое множество.

Описанный способ дает возможность получить необходимое множество S автоматически, без использования предварительно размеченных текстов. Таким образом, реализуется первая часть предлагаемого метода.

Формирование множества гипотез о признаках части речи слов в предложении

Следующей подзадачей является формирование множества гипотез H о признаках части речи слов в предложении. Для этого, для каждого слова в предложении из словаря извлекается список возможных частей речи. Далее, с помощью перестановки возможных частей речи для каждого слова формируется искомое множество H .

Поиск в множестве корректных шаблонов каждой из гипотез

Полученное в первой части множество корректных шаблонов S предложений достаточно велико (1). Поиск перебором в таком множестве будет крайне неэффективен. В связи с этим, предлагается организовать хранение множества корректных шаблонов с помощью словаря.

В нашей реализации описываемого метода шаблон предложения s_i представляет собой последовательность чисел, т. е. части речи кодируются числами

$s_i = \{m_i\}, i = 1, \dots, n, m_i$ — закодированная часть речи i -ого слова в предложении.

Для организации хранения и поиска по большому числу таких числовых последовательностей был выбран словарь неограниченной вложенности, где на каждом уровне ключом является код части речи, а значением — кортеж из словаря следующего уровня и числа, означающего количество встреч предложения с частями речи, которые были закодированы использованными числами.

$S[m_1][m_2] \dots [m_n] = (S_{n+1}, N)$, где S_{n+1} — словарь следующего уровня вложенности, а N — число встреч шаблона s_i .

Такая структура хранения довольно проста в реализации и использовании, и позволяет выполнять поиск по большому числу шаблонов предложений с приемлемой скоростью, что подтверждается экспериментом.

Выделение одной из гипотез на основе результатов поиска

Поиск каждой из гипотез h_i в множестве известных шаблонов S может привести к трем различным исходам:

1. $H \cap S = \emptyset$ — в имеющемся множестве корректных шаблонов не найдено ни одной гипотезы
2. $H \cap S = \{h_j\}$ — в имеющемся множестве корректных шаблонов найдена одна гипотеза
3. $H \cap S = \{h_j \dots h_k\}$ — в имеющемся множестве корректных шаблонов найдено несколько гипотез

Большое число результатов первого исхода говорит о том, что имеющегося множества корректных шаблонов недостаточно, и требуется его пополнение.

В случае второго исхода единственная найденная гипотеза h_j считается верной.

В случае третьего исхода выбирается гипотеза, которая встречалась наиболее часто.

Полученная в результате гипотеза h_j является в среднем более корректной, чем случайно выбранная из исходного множества H . Это подтверждается экспериментом.

Вывод

Описанный в статье метод позволяет повысить точность частеречной классификации слов в предложении. Поставленные эксперименты подтверждают этот факт. При этом, он обладает важным преимуществом — для его работы не требуются предварительно размеченные тексты или иные специально подготовленные данные, за исключением словарей морфологических описаний слов.

Улучшение указанной характеристики позволяет повысить качество морфологического анализа в системах DLP, что, в свою очередь, приводит к повышению защищенности таких систем.

Литература

1. Левцов В.Ю. Контроль подмены символов в системах борьбы с утечками конфиденциальных // «Information Security/ Информационная безопасность» — 2009. — № 5. — С. 28.
2. Боярский К.К., Каневский Е.А. Разработка инструментария для полуавтоматической морфологической разметки текста // Труды международной конференции «Корпусная лингвистика — 2008». — СПб.: С-Петербургский гос. Университет, Факультет филологии и искусств, 2008. С. 83–88.

ЗАДАЧА ОПТИМИЗАЦИИ ЗАЩИЩЕННОСТИ КАНАЛОВ УПРАВЛЕНИЯ РОБОТОТЕХНИЧЕСКИМИ СИСТЕМАМИ

Кустов В.А.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
vitaly_kustov@mail.ru*

Научный руководитель:

Комаров И.И.

*доцент кафедры безопасных информационных технологий СПб НИУ ИТМО,
i_krov@mail.ru*

Аннотация: В данной статье рассмотрена специфика робототехнических систем при обеспечении безопасности, описана задача оптимизации защищенности каналов управления данных систем, определены цели, частные задачи и ожидаемый результат исследования.

Введение

В последние годы наблюдаются высокие темпы технологического развития, и одним из передовых направлений является робототехника, имеющая очень широкую область применения. При таком разнообразном использовании робототехнических устройств и систем возникает вопрос обеспечения безопасности их управления. Рост технических возможностей робототехнических систем позволяет применять более сложные методы и средства защиты каналов управления.

Актуальность работы обуславливается противоречием между практической потребностью обеспечения безопасности управления робототехническими системами различного назначения и недостаточным уровнем развития научно-методического аппарата оптимизации безопасности каналов управления робототехническими системами для обеспечения приемлемого уровня риска, особенно в режиме автономного коалиционного поведения роботов.

Основной целью работы является повышение качества защиты информации за счет применения оптимизации при обеспечении безопасности каналов управления робототехническими системами с использованием средств защиты информации.

Задача оптимизации защищенности каналов управления

Основными параметрами защищенности для робототехнических систем, как и для любого объекта защиты, являются целостность, доступность и конфиденциальность. К этим параметрам, учитывая специфику рассматриваемых автономных мобильных комплексов можно добавить непрерывность, гибкость и скрытность. Требование непрерывность обуславливается необходимостью управления системой с заданным интервалом; гибкость характери-

зует возможность восстановления управления при недоступности основного канала или управляющего узла и использованием других структур или методов управления; требование скрытности появляется в случае необходимости маскировки осуществления управления робототехнической системой.

Построение систем защиты каналов управления должно происходить в соответствии с предназначением и потенциальными возможностями робототехнической системы и может основываться на классической концепции управления рисками информационной безопасности. Поэтому одной из частных задач работы является разработка модели оценки рисков, основанной на нормативно закреплённом подходе, предписывающем оценку степени ущерба в случае наступления инцидента информационной безопасности и вероятности реализации этого инцидента.

Основное влияние на значение оценки риска робототехнических систем оказывают два независимых фактора: потенциальные возможности собственно активных компонентов и технология реализации каналов управления ими.

Задача оценка степени (величины) ущерба, вызванного инцидентом информационной безопасности в зависимости от потенциальных возможностей оконечных элементов является нетривиальной задачей и предлагает учет таких разнородных факторов как: возможность человеческих жертв, энергетическая и инструментальная оснащённость роботов, кинетические характеристики, стоимость владения и многие другие.

Оценка вероятности возникновения инцидентов информационной безопасности предполагает учет таких факторов как: модель безопасности, технология связи, модель управления и может учитывать статистику компрометации конкретных каналов.

По результатам оценки рисков информационной безопасности робототехнических систем может быть сформулирована задача синтеза оптимальной системы управления, общее решение которой и является прогнозируемым результатом работы.

Заключение

Для решения задачи оптимизации защищённости каналов управления робототехнических систем необходимо разработать модель оценки рисков информационной безопасности и, на ее основе, сформулировать и решить задачу синтеза оптимальной системы защиты каналов управления, учитывающие все ограничения, накладываемые особенностями исследуемых робототехнических систем.

Литература

1. *Макаров И.М., Топчиев Ю.И.*, Робототехника: История и перспективы. — М.: Наука; Изд-во МАИ, 2003.
2. *Медведев В.С., Лесков А.Г., Ющенко А.С.* Системы управления манипуляционных роботов. — М.: Наука, 1978.

ПРОБЛЕМЫ АУТЕНТИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ БИОМЕТРИЧЕСКИХ ХАРАКТЕРИСТИК

Малышкин Г.А.

*магистрант кафедры безопасных информационных технологий
СПб НИУ ИТМО, malyshkin.g.a.spb@yandex.ru*

Научный руководитель:

Каторин Ю.Ф.

*д. в. н., профессор кафедры безопасных информационных технологий
СПб НИУ ИТМО, katorin@mail.ru*

Аннотация: В работе рассматриваются проблемы аутентификации пользователя при использовании биометрических данных человека.

Цель данной работы — анализ возможностей преодоления системы защиты, основанной на использовании методов биометрической аутентификации.

Решение

В настоящее время одним из самых перспективных направлений в системах контроля доступа становится использование биометрических данных человека. Такой способ аутентификации очень удобен. Однако биометрия находится в самом начале длинного пути, и существует ряд проблем, связанных с относительной новизной данной технологии.

На данный момент в биометрических системах для аутентификации пользователя успешно реализованы следующие характеристики: радужная оболочка глаза, отпечаток пальца, отпечаток ладони, сосудистые рисунки, геометрия лица, отпечаток голоса, подпись, сравнение ДНК.

Основные биометрические характеристики, а также виды их реализации приведены в табл. 1.

В работе рассмотрены известные методы атак на биометрические системы: подделка отличительной черты, воспроизведение поведения пользователя, перехват биометрических показателей, воспроизведение «биометрической подписи». Проанализированы преимущества и недостатки указанных методов. Предложены возможные способы защиты от подобных атак, которые приведены в табл. 2.

Кроме технических недостатков процедуры биометрической аутентификации, отмечены и другие аспекты, волнующие пользователей при применении данного класса систем, такие как: вмешательство в частную жизнь; личные, культурные и религиозные аспекты; пригодность для всех пользователей.

В результате сделан вывод, что использование аутентификации с использованием биометрических характеристик является очень перспектив

Биометрическая характеристика	Регистрирующее устройство	Образец	Исследуемые черты
Радужная оболочка глаза	видеокамера, способная работать в инфракрасном диапазоне, камера для ПК	чёрно-белое изображение радужной оболочки глаза	полоски и бороздки в радужной оболочке глаза
Отпечаток пальца	периферийное устройство настольного компьютера, карта стандарта PC Card, мышь, микросхема или считыватель, встроенный в клавиатуру	изображение отпечатка пальцев (оптическое, на кремниевом фотоприёмнике, ультразвуковое, или бесконтактное)	расположение и направление гребешковых выступов и разветвлений на отпечатке пальцев, мелкие детали
Геометрия лица	видеокамера, камера для ПК, фотоаппарат	изображение лица (оптическое или тепловое)	относительное расположение и форма носа, расположение скул
Кисть	запатентованное настенное устройство	Трёхмерное изображение верха и боков кисти	высота и ширина костей и суставов кисти и пальцев
Сетчатка	запатентованное настольное или настенное устройство	изображение сетчатки	расположение кровеносных сосудов на сетчатке
Отпечаток голоса	микрофон, телефон	запись голоса	частота, модуляция и продолжительность голосового образа
Подпись	планшет для подписи, перо для ввода данных	изображение подписи и показания соответствующих динамических измерений	частота, модуляция и продолжительность голосового образа
Динамика нажатия клавиш	клавиатура	ритм машинной подписи	время задержки (промежуток времени, в течение которого пользователь удерживает конкретную клавишу) время «полёта» (промежуток времени, который требуется пользователю для перехода с одной клавиши на другую)

Таблица 1. Биометрические характеристики

Описание атаки	Защита от данной атаки
1. Подделка отличительной черты	
Злоумышленник изготавливает копию физической отличительной черты законного пользователя и предъявляет эту копию биометрическому датчику	<p><i>Снятие показателей с высоким уровнем детализации</i></p> <p>При изготовлении эталонного шаблона, с законного пользователя снимают дополнительные биометрические показатели, так что простая копия физической отличительной черты законного пользователя не будет отражать все ее параметры.</p>
2. Воспроизведение поведения пользователя	
Злоумышленник записывает поведенческую отличительную черту пользователя и воспроизводит ее биометрическому датчику	<p><i>Изменяемое поведение</i></p> <p>При каждой попытке аутентификации система требует от пользователя различного проявления его поведенческой биометрической характеристики, так что просто ее запись и воспроизведение не будут приниматься.</p>
3. Перехват биометрических показателей	
Злоумышленник перехватывает биометрические показатели законного пользователя в момент их передачи между устройствами, для дальнейшего их использования	<p><i>Шифрование биометрических данных</i></p> <p>Биометрические данные шифруются сразу после их получения от пользователя устройством считывания, их передача между устройствами осуществляется только в зашифрованном виде.</p>
4. Воспроизведение «биометрической подписи»	
Злоумышленник воспроизводит показатель биометрического датчика — «подпись», которые далее обрабатываются системой так, словно были сняты с реального человека.	<p><i>Шифрование биометрических данных</i></p> <p>Биометрические данные шифруются сразу после их получения от пользователя устройством считывания, их передача между устройствами осуществляется только в зашифрованном виде.</p> <p><i>Аутентификация биометрической «подписи»</i></p> <p>Меры аутентификации принимаются в отношении биометрических данных, чем гарантируется их поступление только из заслуживающих доверия источника.</p> <p>Использование ЭЦП для обеспечения целостности биометрической «подписи»</p>

Таблица 2. Методы атак на биометрические системы

ным направлением развития систем контроля доступа, несмотря на существующие проблемы и недостатки. Биометрический параметр можно использовать совместно с паролем, смарт-картой, либо другой биометрической характеристикой для обеспечения двухфакторной или многофакторной аутентификации. Такая комбинация существенно усложнит возможность преодоления системы контроля доступа.

Литература

1. Руководство по биометрии / Р.М. Болл, Дж.Х. Коннел, Ш. Панканти и др. — М.: Техносфера.
 2. *Иванов А.* Высоконадежная биометрическая аутентификация пользователя: последний дюйм первой мили / А. Иванов, А. Малыгин // Первая миля. — 2007. — № 2. — С. 20–24.
 3. *Бабенко Л.К.* Аутентификация диктора с использованием изменяемого множества ключевых слов / Л.К. Бабенко, В.М. Федоров, П.Ю. Юрков // Известия ТРТУ. Спец. вып.: Матер. науч. конф. ТРТУ. — Таганрог : Изд-во ТРТУ, 2004. — № 1(36). — С. 128.
 4. *Hansen J.* Speech under stress: Analysis, Modeling and Recognition / J. Hansen, S. Patil // Speaker Classification I, LNAI 4343 / S. Müller (Ed.). — Berlin; Heidelberg: Springer-Verlag, 2007. — P. 108–137.
 5. *Uludag U.* Attacks on biometric systems: a case study in fingerprints / U. Uludag, A. Jain // Proc. SPIE. — 2004. — Vol. 5306. — P. 622–633.
-

МЕТОД ИДЕНТИФИКАЦИИ ОБЪЕКТОВ В СИСТЕМАХ ВИДЕОНАБЛЮДЕНИЯ НА ОСНОВЕ МАТЕМАТИЧЕСКОГО АППАРАТА ИММУНОКОМПЬЮТИНГА

Соломатин А.Ю.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
solomatin@gmail.com*

Аннотация: В статье предложен подход для решения задачи идентификации объектов в видеопотоке. Рассмотрена возможность использования научно-методического аппарата иммунокомпьютинга для решения поставленной задачи. Формулируются цели, задачи исследования, а также полученные промежуточные результаты.

Введение

В рамках предлагаемой статьи объекты в видеопотоке описываются рядом признаков и показателей, с помощью которых можно четко характеризовать объект или его состояние. Например, собрав все показатели воедино и проанализировав их, можно идентифицировать объект или его состояние из ряда множества других объектов и состояний. Основная сложность при таком подходе заключается в том, что зачастую информация, описывающая объект, представлена в виде большого количества регистрационных данных.

Решение

Для решения проблемы по обработке и анализу большого неструктурированного потока информации существуют различные математические подходы. Например, одним из таких подходов является метод главных компонент (Principal Component Analysis, PCA). Задача данного метода заключается в извлечении из большого массива данных нужной информации, поскольку данные всегда (или почти всегда) содержат в себе нежелательную составляющую, иначе называемую шумом. Другой популярный подход — деревья решений. Метод заключается в представлении данных в иерархической последовательной структуре [1]. Этот метод успешно применяется для решения задач классификации и регрессии.

В рамках статьи предлагается использовать метод иммунокомпьютинга. Иммунокомпьютинг изучает принципы обработки неструктурированной информации для решения сложных практических задач. Математический базис, применяемый в рамках вычислительных процедур иммунокомпьютинга, основан на свойствах сингулярного разложения матриц (Singular Value Decomposition, SVD). Математическая процедура представляет собой декомпозицию вещественной матрицы с целью приведения ее к канониче-

скому виду. Сингулярное разложение является удобным методом при работе с матрицами. Оно показывает геометрическую структуру матрицы и позволяет наглядно представить имеющиеся данные.

Одна из наиболее часто решаемых задач с помощью иммунокомпьютинга — это задача распознавания объектов. В данных задачах для определения связи между объектами вводится понятие энергии связи, которое используется в качестве аналога расстояния между объектами [2].

Для иллюстрации задачи идентификации объектов в видеопотоке проводится эксперимент, в рамках которого применяется предлагаемый способ на основе иммунокомпьютинга в целях поиска и идентификации лиц человека в видеопотоке в режиме реального времени. Предполагается возможность создания такого признакового пространства, которое сможет четко характеризовать найденное лицо в виде числового вектора. В настоящее время проводятся исследования, направленные на формирование такого признакового пространства и оценки влияния информационного шума на качество распознавания.

Предполагается создание обучающих выборок с множеством классов, каждый класс характеризует лицо одного человека. При получении входного изображения видеопотока будут выделяться лица и по каждому лицу будет сформирован «портрет», представляющей собой вектор характеристик распознаваемого лица, и, в результате применения аппарата иммунокомпьютинга отнесение его к тому или иному классу обучающей выборке.

В результате исследования предполагается разработка программного модуля для автоматической идентификации лиц в видеопотоке в режиме реального времени.

Литература

1. *S. Murthy*. Automatic construction of decision trees from data: A Multi-disciplinary survey. 1997.
2. *Alexander O. Tarakanov, Victor A. Skormin, S.P. Sokolova*. Immunocomputing: Principles and Applications. N.Y. Springer. 2003.

ИДЕНТИФИКАЦИЯ АВТОРОВ КОРОТКИХ СООБЩЕНИЙ ПОРТАЛОВ СЕТИ ИНТЕРНЕТ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАТЕМАТИЧЕСКОЙ ЛИНГВИСТИКИ

Сухопаров М.Е.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
sukhoparovm@gmail.com*

Аннотация: В тезисах предложен метод идентификации пользователей порталов сети Интернет. Подход основан на использовании вероятностных характеристик типовых элементов идентификации. Подход позволяет повысить точность идентификации пользователей, необходимой при борьбе с различными проявлениями астротерфинга в сети Интернет.

Введение

Повсеместное использование ИТКС и сравнительная легкость доступа к ресурсам сети Интернет обуславливают необходимость в контроле над информационными потоками и идентификации возможных направлений информационного воздействия. В связи с чем встает вопрос о необходимости анализа текстовой информации и установления авторства текста коротких сообщений различных порталов сети Интернет, вследствие недостаточной развитости механизмов идентификации пользователей. Анализ текстовых сообщений комментариев, блогов, а так же борьба с различными проявлениями астротерфинга являются крайне важным для борьбы с процессом искусственного формирования общественного мнения в сети Интернет.

Задачи работы

Решаемая научная задача состоит в обосновании и разработке научно-методического аппарата идентификации авторства текстовой информации (M_m), предназначенного для автоматического вычисления информации (\hat{I}) о текстах предметной области с минимальной потерей полноты, точности и адекватности (ΔI_{\min}) при обработке I в комплексах вычислительных средств СЗИ, базирующегося на использовании:

- множество моделей ($H = \{H | f(q', d')\}$), свойства которых q' и d' имеют функциональную зависимость $f(q', d')$ от свойств Q' представления информации и требуемых свойств D' представления данных в ИС;
- множество методов (M) обеспечивающих отражение текстовой информации T в элементы данных O с помощью функции F .

Математически указанная задача формулируется следующим образом: найти множество $M : Q \rightarrow Q', D \rightarrow D', \forall q'(q' \in Q), \forall d'(d' \in D)$ такое, что $\exists (\hat{I} = \hat{I})\{M, H, q, q', d, d', \Delta I\}$, при $\|I - \hat{I}\| \Rightarrow \Delta I_{\min}$. $M : T \xrightarrow{F} O$.

В качестве задачи работы также ставится разработка и создание программного макета, реализующего предлагаемый подход, а также оценка производительности и эффективности разработанной макетной реализации.

Заключение

Реализация предлагаемых решений позволит идентифицировать авторов коротких сообщений форумов и блогов сети Интернет во время различных пиар-акций с целью борьбы и контроля над формированием и манипуляцией общественным мнением и другими проявлениями астротерфинга.

ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ ИХ ПОВЕДЕНИЯ В СЕТИ ИНТЕРНЕТ

Юрасов Д.С.

*аспирант кафедры безопасных информационных технологий СПб НИУ ИТМО,
yurasov17@gmail.com*

Аннотация: В данной работе рассматривается проблема идентификации двух и более пользователей, осуществляющих доступ в сеть Интернет с помощью общих компьютера и браузера. Предложен алгоритм разделения истории посещений сайтов такими пользователями.

Введение

Одной из важных задач в теории защиты информации является задача идентификации пользователя в сети Интернет. Проблемы идентификации и аутентификации являются основными при построении систем разграничения доступа, которые, в свою очередь, играют ключевую роль в системах обеспечения информационной безопасности.

В современных информационных системах, как правило, применяются способы идентификации, основанные на информации о IP-адресах компьютеров посетителей и хранения на компьютере пользователя данных Cookie [1, 2]. Существенным недостатком этих методов является отождествление пользователя и его компьютера или браузера, что не всегда корректно. По результатам последних исследований, доля многопользовательских компьютеров варьируется от 20 до 40 процентов [3].

Целью данной работы является разработка метода, позволяющего идентифицировать различных пользователей, осуществляющих доступ в сеть Интернет с помощью общих компьютера и браузера.

Для решения данной задачи в работе использованы данные о посещении пользователем сайтов в сети Интернет, которые могут быть собраны с помощью Cookie или других технологий, либо получены из сторонних источников.

Метод идентификации пользователей с общими компьютером и браузером

Подготовка данных

Для сбора информации о посещении пользователями сайтов использовались Cookies, поэтому далее будем отождествлять пользователей с браузером и называть их «куками».

Как правило, системы разграничения доступа не располагают информацией о количестве людей, пользующихся конкретными компьютером и

браузером. Чтобы решить эту проблему, склеим истории посещений сайтов для случайно выбранных пар кук. Всего было склеено 131868 пар пользователей. В результате получили данные, в которых за каждой кукой будет скрываться минимум два пользователя. Такие куки будем называть многопользовательскими.

Сайты категории 1

При решении поставленной задачи будем опираться на тот факт, что у большинства пользователей существуют сайты, которые он посещает достаточно часто и регулярно. Для выявления таких сайтов историю посещений каждого пользователя разобьем на так называемые сессии так, чтобы временной интервал между соседними посещениями сайтов внутри одной сессии не превышал 30 минут. Сессии склеенных кук были случайным образом перемешаны между собой, но так, чтобы порядок сессий исходных не склеенных пользователей сохранился. Будем говорить, что сайт принадлежит к категории 1 для пользователя, если выполнены два условия:

1. он встретился хотя бы в десятой части всех известных сессий;
2. если пронумеровать все известные сессии пользователя, то дисперсия номеров сессий, в которых встретился данный сайт, больше определенного числа.

Первое условие гарантирует частоту посещения сайта, а второе регулярность, исключая сайты, к которым пользователь проявил лишь локальный интерес, то есть интенсивно интересовался им лишь на протяжении некоторого относительно короткого интервала времени.

В результате в изучаемых нами данных у всех пользователей нашелся хотя бы один сайт первой категории.

Если взять историю обычных, не склеенных пользователей, то более чем у 88% из них найдется хотя бы один сайт первой категории. Для всех сайтов было определено количество пользователей, для которых данный сайт является сайтом первой категории. Самыми популярными такими сайтами оказались vk.com, e.mail.ru и odnoklassniki.ru.

Кластеризация сайтов первой категории

Известно, что за каждой кукой в полученных данных скрывается как минимум два реальных пользователя, следовательно, и сайты первой категории должны для них делиться минимум на две соответствующие группы; далее предполагаем, что их ровно две.

Рассмотрим конкретную куку u . Пусть у нее было всего n сессий, ее сайтами первой категории являются s_1, \dots, s_m . Для всех j от 1 до m и всех k от 1 до n сайту s_j поставим в соответствие вектор s_{jk}, \dots, s_{jm} , где s_{jk} принимает значение 1, если пользователь u посещал сайт s_j в k -ю сессию и 0 иначе.

В качестве расстояния между сайтами первой категории s_i и s_j пользователя u будем использовать расстояние Жаккара J , равное

$$J = \frac{M_i + M_j}{M_i + M_j + M_{ij}},$$

где M_{ij} — количество сессий пользователя u , в которых присутствуют и s_i и s_j ; M_i — количество сессий пользователя u , в которых присутствует сайт s_i , но нет сайта s_j ; M_j — количество сессий пользователя u , в которых присутствует сайт s_j , но нет сайта s_i [4].

Понятно, что в идеальной ситуации расстояние между сайтами разных пользователей куки u должно быть близко к 1, а между сайтами одного пользователя к 0. Следовательно, далее можем разделить сайты первой категории куки u на две группы с помощью стандартной иерархической кластеризации, используя введенное расстояние.

Для примера, рассмотрим одну из склеенных кук. На графике по оси абсцисс отметим номера сессий этой куки, по оси ординат номера ее сайтов первой категории. Точку для сайта i сессии j отмечаем, если кука посетила этот сайт в эту сессию. Звездочка или черта обозначают, какому из изначальных не склеенных пользователей принадлежит данная сессия (см. Рис. 1). То есть все точки над одной сессией отмечены одинаковым символом. Из данного рисунка видно, что сайты 1,4 и 9 посещал только пользователь помеченный «*», сайты 2,6,7,8 посещал только пользователь помеченный «\|», сайт 3 в основном пользователем помеченным «\|».

Получившееся автоматическое разделение сайтов на кластеры выглядит вполне естественным и логичным (см. Рис. 2).

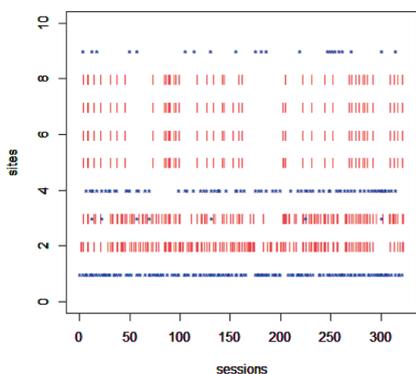


Рис. 1. Сессии пользователя

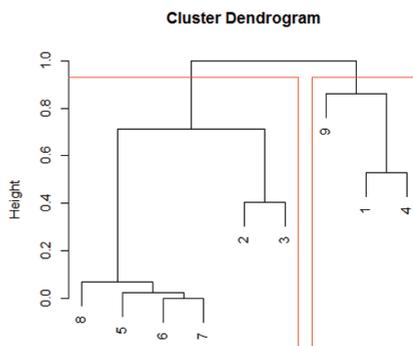


Рис. 2. Кластеризация сайтов категории 1

Результаты

Таким образом, для каждой куки у нас есть список ее сайтов первой категории и их кластеризация на две группы А и В. Для каждой ее сессии посчитаем сколько в ней сайтов из одного кластера и сколько из другого. Будем говорить, что сессия принадлежит пользователю U_A , если в ней встречается больше сайтов из кластера А, чем из В; сессия принадлежит пользователю U_B , если в ней встречается больше сайтов из кластера В, чем из А. Почти все сессии получили метку U_A или U_B , сессии, не получившие таких пометок, далее не рассматриваем. В итоге все оставшиеся сессии имеют ровно по две метки: U_A или U_B и «*» или «|». Для каждой куки построим таблицу сопряженности начальных и новых меток. В качестве количества правильно разгаданных сессий для нее возьмем большую из сумм чисел на главной и побочной диагоналях. Средний процент правильно распознанных сессий по всем пользователям равен 76.7%.

Отметим, что процент правильно распознанных сессий пользователя зависит от их количества. Это видно из графика (см. Рис. 3), где по оси абсцисс отмечен результат целочисленного деления числа сессий на 25, а по оси ординат средний процент правильно распознанных сессий для пользователя с их соответствующим количеством. Таким образом, для качественного разделения истории посещения сайтов многопользовательской кукой необходимо, чтобы у пользователя было, по меньшей мере, 100 сессий.

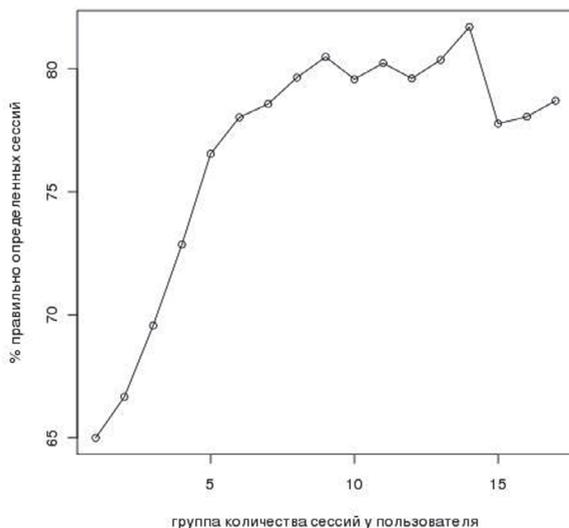


Рис. 3. Качество распознавания в зависимости от количества сессий у пользователя

Заключение

В данной работе представлен алгоритм идентификации пользователей, про которых известно, что они осуществляют доступ в сеть Интернет с помощью общих компьютера и браузера. Он основан на выявлении для пользователя сайтов, которые он посещает часто и регулярно. Целесообразность использования представленного алгоритма была обоснована. Был проведен эксперимент и получены положительные результаты на собранных нами данных при выбранном критерии качества. Средний процент правильно распознанных пользовательских сессий равен 76.7%.

Литература

1. Understanding IP Addressing: Everything You Ever Wanted To Know http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf [дата просмотра: 04.04.2013].
 2. *McKinkley, K.*: Cleaning Up After Cookies. iSec Partners White Paper http://www.isecpartners.com/storage/white-papers/iSEC_Cleaning_Up_After_Cookies.pdf [дата просмотра: 04.04.2013].
 3. *Gian Fulgoni*. When the Cookie Crumbles http://www.comscore.com/Insights/Blog/When_the_Cookie_Crumbles [дата просмотра: 04.04.2013].
 4. *Lipkus A.H.* A proof of the triangle inequality for the Tanimoto distance/Lipkus A.H. //J Math Chem 26 (1999) (1–3): 263–265.
-

МЕТОДЫ И СРЕДСТВА КЛОНИРОВАНИЯ RFID-МЕТОК

Юрьева Р.А.

*магистрант кафедры безопасных информационных технологий
СПб НИУ ИТМО, raddayurieva@gmail.com*

Левина А.Б.

*доцент кафедры Безопасные информационные технологии НИУ ИТМО
alla_levina@mail.ru*

Аннотация: Производители чипов RFID в настоящее время сосредотачиваются на улучшении модификации меток с тем, чтобы усложнить их клонирование. Предлагаемый здесь метод заключается в генерации нового случайного числа, основанного на теории хаоса и произведенного хаотически, каждый раз, когда метка прочитана.

Введение

Клонирование RFID-меток может привести к огромным денежным убыткам во многих коммерческих и государственных организациях. Предлагаемый способ представляет собой хаотически синхронизированный метод «секретов» для обнаружения атак на RFID-метки и их клонирования, чтобы точно определить различные теги с тем же ID. Данный метод требует лишь небольшого количества перезаписи памяти для написания случайного числа в метке.

Защита RFID-меток от клонирования

Предлагаемый метод заключается в генерации нового случайного числа, основанного на теории хаоса и произведенного хаотически, каждый раз, когда метка прочитана. Процессор окончательной обработки данных, который генерирует эти хаотические числа, обнаруживает клонирование метки, как только и подлинная и клонированная метка прочитана.

Анализ уровня безопасности данной методики

Теперь оценим уровень безопасности представленной методики. Вероятность успешной догадки хаотического числа, которое изменяется после каждого прочтения метки, равняется $\frac{1}{2}n$, где n — длина этого числа в битах. Когда происходит атака на клонирование, возможны три взаимоисключающих результата:

- Случай 1: Подлинная метка считана до скопированной метки и тревога таким образом вызвана, когда считана скопированная метка.

- Случай 2: Скопированная метка перед подлинной меткой, и тревога таким образом вызвана, когда подлинная метка прочитана.
- Случай 3: Подлинная метка не прочитана вообще, и таким образом не было вызвано тревоги при копировании.

В первом случае, если клонированная метка обнаружена, как только она прочитана один раз, негативный эффект от клонирования может быть предотвращен. Во втором случае, если клонированная метка пройдет проверку, не поднимая тревогу, но система обнаруживает атаку на клонирование, когда подлинная метка будет прочитана. В третьем случае безопасность терпит неудачу, и атака остается незамеченной. Уровень системы безопасности характеризуется вероятностью первого случая, который демонстрирует, как часто угрозы предотвращены, и вероятностью первого и второго случаев, которые показывают, как часто обнаружены угрозы.

Заключение

В предложенном методе используется хаотическая система. Она быстро вычислима, и благодаря этому можно создать последовательность с чрезвычайно большой длиной цикла. Каждый раз, когда метка «прочитывается», индикатор сначала проверяет статический идентификатор метки. Если это число является верным, индикатор затем сравнивает метку с синхронизированным «тайным» числом, которое хранится для данного тега. Если эти числа совпадают, то тег проходит проверку, в противном случае раздается сигнал тревоги. После проверки индикатор создает новый синхронизированный «секрет».

Литература

1. *Mikko Lehtonen, Florian Michahelles, and Elgar Fleisch*, How to detect Cloned Tags in a Reliable Way from Incomplete RFID-Traces, Information Management, ETH Zuerich, 8092 Zuerich, Switzerland
-

РАЗДЕЛИМОСТЬ ЗАПРОСОВ ПРИ НАГРУЗОЧНОЙ АТАКЕ ВЕБ-СЕРВЕРОВ

Пряхин В.Е.

аспирант кафедры БИТ СПбНИУ ИТМО, moov252@gmail.com

Зикратов И.А.

*д. т. н., профессор, заведующий кафедрой БИТ СПбНИУ ИТМО,
zikratov@cit.ifmo.ru*

Аннотация: В работе рассмотрен проведенный эксперимент по нагрузочной атаке работающего веб-сервиса, накопление статистики во время штатной работы и во время атаки, анализ этой статистики и выявление закономерностей, позволяющих отделить легитимные запросы и атакующие. Графически показана линейная делимость запросов по весу отдаваемого контента.

Введение

В настоящее время угроза распределенных атак в обслуживании веб-серверов стоит как никогда остро — простой сервисов, вызванные распределенной нагрузочной атакой, наносят серьезный финансовый и репутационный урон своим владельцам. Также, распределенные атаки в обслуживании все чаще стали использоваться как средство конкурентной борьбы, или способ достижения некоторых политических целей.

Цель работы — повышение эффективности распознавания нагрузочных атак в обслуживании веб-серверов с помощью методов интеллектуального анализа данных, выявление закономерностей в распределении запросов, благодаря которым можно более точно отделять периоды атаки от периодов штатной работы.

Идея работы заключается в том, что каждый веб-сервис имеет уникальное распределение характерных запросов на отдачу контента (тяжелого — большие изображения, медиа-контент, файлы; и легкого — пользовательские скрипты, стили оформления, небольшие статичные изображения), либо генерацию веб-страниц (нагружающих как сам веб-сервер, так и сервер баз данных), что отражается на структуре входящего и исходящего трафика, имеет уникальный, характерный для него тип нагрузки и распределения процессорного времени и памяти по выполняющимся процессам — вся эта статистика формирует уникальный «отпечаток», который можно использовать как эталон для сравнения и выявления аномалий, критических ситуаций, и причин, их вызывающих.

Исследование

В ходе работы был проведен подробный сбор статистики в условиях реальной нагрузочной атаки на веб-сервис, работающий на кластере серверов.

Информация по каждому запросу добавлялась в базу данных и включала такие параметры, как IP адрес, user-agent, тип запроса (GET/POST), время запроса в виде UNIX Timestamp, адрес страницы/файла на сервере (URI), и так далее. Также, такая же база запросов была накоплена за время штатной работы сервиса в течение двух месяцев.

За период штатной работы сервиса на графике количества запросов в секунду отчетливо видны суточные колебания посещаемости (см. Рис. 1), а среднее количество запросов в секунду не превышает 5.

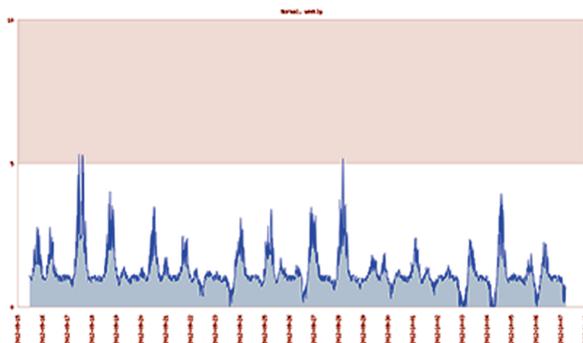


Рис. 1. Запросы в секунду во время штатной работы, масштаб 2 недели

С началом DDoS — атаки был зафиксирован (см. Рис. 2) резкий многократный (на порядки) всплеск количества запросов к веб-сервису в секунду, остававшийся практически на одном уровне в течение нескольких дней, и плавно спадавший по окончании атаки, что говорит о использовании P2P-ботнета, требующего времени на донесение сигнала о конце атаки до всех участников.

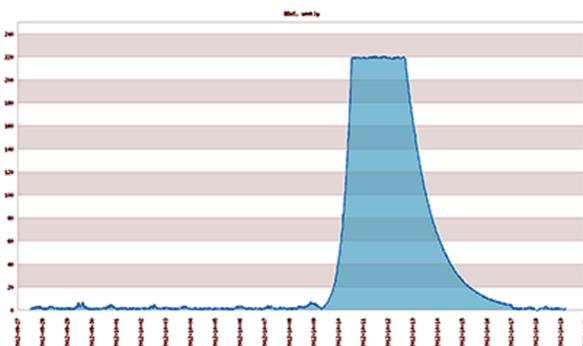


Рис. 2. Запросы в секунду во время атаки, масштаб — месяц

На графиках распределения запросов по объему отдаваемого контента в килобайтах (легкий — скрипты, стили, небольшие картинки; тяжелый — изображения, фотографии) во время штатной работы (см. Рис. 3) и нагрузочной атаки (см. Рис. 4) можно заметить линейную разделимость паразитных запросов и легитимных.

Также была выявлена характерная последовательность запросов для легитимных пользователей — при загрузке страницы в определенном порядке загружается статика, скрипты, и оформление, когда атакующие запросы идут подряд и без закономерностей.

Соответственно, собрав статистику работы сервиса во время штатной работы и построив распределение запросов, можно отсеять IP-адреса, производящие DDoS-атаку, так как они не запрашивают все нужные адреса, а только избранные — те, что либо больше всего нагружают кластер на генерацию, либо те, что отдадут наиболее тяжелый контент, реализуя атаку на истощение пропускной способности канала.

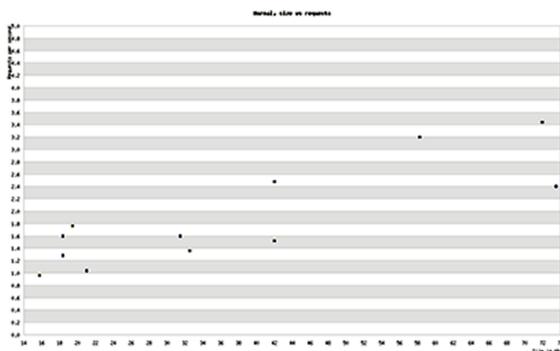


Рис. 3. Распределение количества запросов по весу контента

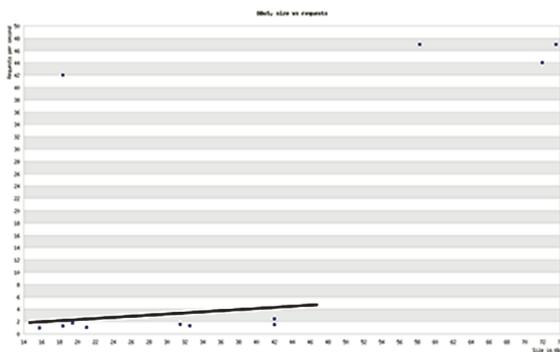


Рис. 4. Распределение запросов по весу контента во время атаки, отмечена возможная линейная разделимость

Выявленные в ходе работы закономерности и статистику запросов можно использовать как входные данные для различных алгоритмов интеллектуального анализа данных (Data mining), что повысит точность определения нештатных ситуаций и аномалий трафика.

Заключение

В работе были рассмотрены закономерности, найденные в результате проведения реального эксперимента и анализа накопленной статистики запросов во время штатной работы сервиса и во время нагрузочной DDoS-атаки, выявлена линейная делимость легитимных запросов и атакующих по весу контента, наличия, и последовательности запросов.

Литература

1. *К.В. Воронцов* «Лекции по линейным алгоритмам классификации» — 19 января 2009, <http://www.machinelearning.ru/wiki/images/6/68/voron-ML-Lin.pdf>
-

Вероятностные графические модели, нечеткие системы и мягкие вычисления



**Юсупов
Рафаэль Мидхатович**

д.т.н., профессор, чл.-корр. РАН
директор СПИИРАН



**Тулупьев
Александр Львович**

д.ф.-м.н., профессор кафедры информатики СПбГУ
заведующий лабораторией ТИМПИ СПИИРАН

МОДЕЛИ КОМПЛЕКСА «ИНФОРМАЦИОННАЯ СИСТЕМА — ПЕРСОНАЛ — КРИТИЧЕСКИЕ ДОКУМЕНТЫ» ПРИ ОЦЕНКЕ ЗАЩИЩЕННОСТИ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННЫХ СИСТЕМ¹

Азаров А.А.

СПбГУ, аспирант; СПИИРАН, м. н. с., artur-azarov@yandex.ru

Аннотация: Одной из основных проблем исследований в области социо-инженерных атак является развитие приемлемых по вычислительной сложности алгоритмов анализа (оценки) защищенности персонала информационных систем. По предварительным оценкам именно применение вероятностно-реляционного алгоритма поможет существенно уменьшить вычислительную сложность программного комплекса. Использование указанного подхода позволит также увеличить гибкость в задании оценок критичности документов, доступных в системе, оценок шансов успешной реализации атак, описанию системы связей и доступа среди собственно компонент комплекса «информационная система — персонал — критичные документы», и среди указанных компонент и злоумышленника.

Введение

Одной из основных проблем исследований в области социо-инженерных атак является, с одной стороны, разработка подходящих для последующего анализа степени защищенности моделей информационных систем, персонала, набора критичных документов, злоумышленника, а также разнообразных связей между указанными сущностями, а с другой стороны — развитие приемлемых по вычислительной сложности алгоритмов анализа (оценки) защищенности персонала информационных систем. Ожидается [5], что дополнение (либо даже замена) системы переборных алгоритмов реляционно-алгебраическим подходом (и впоследствии на его основе — реляционно-вероятностным подходом) к представлению исходных данных и организации вычислений позволит уменьшить время обработки сведений об информационной системе и ее персонале, а также время, затрачиваемое на построение набора возможных атак. Использование указанного подхода позволит также увеличить гибкость в задании оценок критичности документов, доступных в системе, оценок шансов успешной реализации атак,

¹ Исследование поддержано грантом РФФИ на 2010–2012 гг., проект №10-01-00640-а, грантом СПбГУ на 2011–2013 гг., проект № 6.38.72.2011., Грант РФФИ на 2012–2014 гг., проект №12-01-00945-а, стипендия Правительства Российской Федерации (пр. 874 от 29.10.2012).

описанию системы связей и доступа среди собственно компонент комплекса «информационная система — персонал — критичные документы», и среди указанных компонент и злоумышленника. Наконец, реляционно-алгебраический подход позволяет использовать стандартный инструментарий реляционных баз данных для представления данных в автоматизированных системах анализа защищенности, что в свою очередь может открыть путь к вычислению оценки защищенности на основе SQL-запросов, исполнение которых автоматически распараллеливается и оптимизируется встроенной функциональностью современных СУБД.

Цель доклада — представить основы подхода к алгоритмизации анализа защищенности персонала информационной системы от социоинженерных атак, основанные на реляционно-алгебраическом и — более общем — вероятностно-реляционном подходе. Последний, в свою очередь, допускает адаптацию к использованию не только вероятностных, но и нечетких оценок (а также, возможно, оценок, опирающихся на иные подходы к формализации степеней доверия) для представления неопределенности знаний об анализируемом комплексе и атакующем его злоумышленнике.

В качестве «отправной точки» предлагается использовать модель комплекса «информационная система — персонал — критичные документы», в которой информационная система представлена в виде графа, узлы которого отвечают программно-техническим компонентам системы, а ребра — связям между таким компонентами [4, 9]. Причем как узлам, так и дугам сопоставлены достаточно богатые информационные модели соответствующих сущностей. Сложноустроенные угрозообразующие воздействия на информационную систему в рамках рассматриваемой базовой модели представляются в виде деревьев атак. В этом случае, анализ степени защищенности информационной системы сводится к анализу набора возможных деревьев атак [15]. (В краткой формулировке процесс выглядит достаточно просто; что, однако, не упрощает ни само исследование вовлеченных объектов и процессов, ни сбор необходимой для оценки информации, ни разработку комплексов программ, формирующих оценку защищенности.)

Для обобщения указанного подхода на комплекс «информационная система — персонал» потребуется разработать ряд моделей (математических, информационных, визуальных и проч.) позволяющих учесть новую подсистему — «персонал» — и ее компоненты, связи внутри этой новой подсистемы, связи между подсистемами «персонал» и «информационная система», связи этих подсистем с внешней средой (прежде всего, имеются в виду лица, действия которых составляют или могут составлять угрозу для безопасности информации, хранящейся в информационной системе).

Кроме того, данный подход может быть применен к полученным в ходе социологического исследования психологическим уязвимостям пользователя.

Реляционно-алгебраическая модель комплекса «информационная система — персонал»

С точки зрения анализа защищенности от социоинженерных атак, в комплексе «Информационная система — персонал — критические документы» можно выделить следующие компоненты: I — критичная информация, более точно — система документов, каждый из которых, с одной стороны, характеризуется показателем или показателями критичности, а с другой стороны, атрибутами, характеризующим его доступность с такого-то хоста с правами такого-то пользователя; H — хосты, которые характеризуются своими связями с другими хостами, рядом атрибутов, описывающих текущую конфигурацию программно-технического обеспечения (по этим атрибутам определяется успешность реализации атакующих действий программно-технического характера), а также рядом атрибутов, описывающих права пользователей на данном хосте; U — пользователи, которые характеризуются своими отношениями к группам пользователей в отношении которых установлены определенные политики безопасности, допуском в определенные зоны, а также профилем уязвимости (который формируется, в том числе, на основе сведений о психологической защите), причем профиль уязвимости определяет вероятность успеха социо-инженерных атакующих действий, и связями с другими пользователями; A — атакующие действия, предпринимаемые злоумышленником или группой злоумышленников, которые могут быть направлены как на пользователей, так и на хосты, каждому из которых, в свою очередь, задан вес, который выражается отражает требуемые для осуществления данного атакующего действия ресурсы. Поскольку атаки на хосты рассматривались в работах, систематический обзор которых можно найти, например, в [14, 18–20, 26], то в данном докладе мы формализуем лишь атаки на пользователя. Такая модель легко генерализуется и для общего случая.

Отношения между элементами указанных компонент допускают формализацию тем же путем, как это было выполнено в [22], но семантика таких отношений будет иной, отражающей особенности предметной области.

Вероятностно-реляционная модель атакующих действий

Прежде всего, рассмотрим модель атакующих действий для того, чтобы сформулировать соглашения и описать возможности, которые мы будем подразумевать при описании остальных элементов системы.

Каждая атака требует определенное число ресурсов. Ресурсами могут выступать как денежные средства, так и богатый спектр других средств (возможности психологического, социального, технико-инженерного воздействия).

В упрощенной модели мы можем ввести дискретную шкалу R которая измерять количество требуемых ресурсов. В современных условиях подобное до-

пущение вполне отвечает существующей системе, поскольку финансовые ресурсы могут быть в известной степени конвертированы в любой другой ресурс.

Более сложная модель предполагает использование многомерной шкалы $R = R_1 \times \dots \times R_n$, где R_i — дискретная шкала измерения конкретного ресурса. Примерами такой многомерной шкалы могут выступать шкалы, измеряющие финансовые средства и затрачиваемое время, или финансовые средства, усилия эксперта по психологическому воздействию (психолога), усилия эксперта по техническому воздействию (хакера). Однако следует сразу указать на ряд недостатков усложненной модели. Во-первых, в силу того, что ресурсы скорее всего частично конвертируемы, следует выделить классы эквивалентности различных многомерных измерений, что значительно усложняет модель. Во-вторых, из теории управления известно, что сложность управления подобной системой растет очень быстро при увеличении числа моделируемых ресурсов, что заставляет делать выбор между точностью представления и операбельностью модели. Далее мы будем говорить про шкалу R , которую можно рассматривать как одномерной, так и многомерной с учетом сделанных выше предположений.

Рассмотрим отношение $A_U(a, u)$ которое задают возможности проведения атаки a на пользователя u . $(a, u) \in A_U$ говорит о том, что атака будет воздействовать на пользователя u . $R_{AU}: A_U \rightarrow R$ задает число ресурсов, которые необходимо потратить на соответствующую атаку.

Более общим случаем будет ввести на декартовом произведении $A \times U$ функцию p_{AU} которая определяет вероятность успешности атаки. Для тех пар, которые не входят в A_U эта вероятность окажется тождественно равной нулю (либо же меньшей заранее заданной величины).

Наиболее общим случаем будет введение функции p_{AUR} , заданной на декартовом произведении $A \times U \times R$, которая определяется вероятность успешности данной атаки на данного пользователя при использовании данного количества ресурсов.

Вероятностно-реляционная модель комплекса «информационная система — персонал»

$H_I(h, i)$ — подмножество декартового произведения $H \times I$, которое отображает возможность прямого доступа с хоста h к критичному документу i . В графовой модели подобное отношение формализуется через наличие ребра от вершины h до вершины i . Так, например, $(h, i) \in H_I$, когда i хранится на h .

$U_H(u, h)$ — подмножество декартового произведения $U \times H$, которое отображает непосредственную возможность доступа пользователя u к хосту h . В упрощенной (детерминированной) модели мы предполагаем, что пользователь либо имеет, либо не имеет доступ к соответствующему хосту. В графовой модели подобное отношение формализуется через наличие ребра от вершины u до вершины h . Так, например, $(u, h) \in U_H$ когда u работает за компьютером h .

Рассмотрим декартово произведение $H \times H$. Можно ввести либо отношение $A_H(h_1, h_2)$, которое характеризует непосредственный доступ от одного хоста до другого (связь в сеть, и т. д.), либо вероятность $P_H(h_1, h_2)$, задающую вероятность такого доступа.

Сходные A_U или p_U можно задать и для декартового произведения $U \times U$ (например, дружеские контакты, служебные отношения, и т. д.)

Вариантом обобщения модели будет также введение ресурсных функций (т. е., например, финансовые средства могут переданы пользователю, чтобы он подкупил другого пользователя, с которым он состоит в дружеских отношениях), которые будут определять стоимость атак, однако в рамках данной работы мы будем считать, что подобные стратегии уже закладываются в понятие «атака» и описываются множеством A , и ее ресурсная оценка уже включает все дальнейшие перераспределения ресурсов в рамках взаимодействия между другими узлами.

Благодаря введенным выше понятиям мы можем формализовать понятие достижимости.

Так, в детерминированном случае критическая информация i достижима из хоста h , если существует такой упорядоченный набор хостов h_1, \dots, h_n , что $(h, h_1), (h_1, h_2), \dots, (h_{n-1}, h_n) \in A_H$, и $(h_n, i) \in H_I$. Будем обозначать это отношением $ATT_H(h, i)$. Аналогичное отношение для достижимости хоста h пользователем u будем обозначать как $ATT_U(u, h)$ (в этом случае требуется набор пользователей с соответствующими отношениями). В графовой модели подобное отношение характеризуется достижимостью вершины i из вершины h (h и u соответственно для пользователя).

В случае, если была задана функция p_H , вводится функция $P_H(h, i)$, которая определяется следующим образом:

$$P_H(h, i) = \sum_{\{h\}_n} \left(\prod_{\substack{j: h_j, h_{j+1} \in \{h\} \\ (h_n, i) \in H_I}} p_H(h_j, h_{j+1}) \right),$$

т. е. произведение вероятностей всех ребер на всех путях, которые ведут от h до i .

Аналогичным образом можно ввести функцию $P_U(u, h)$.

В детерминированном случае достижимость $ATT_I(u, i)$ информации i пользователем u определяется существованием такой пары $(u_a, h_a) \in U_H$, что $(u, u_a) \in ATT_U$ и $(h_a, i) \in ATT_H$.

В случае, если мы задали функции вероятности, то введем вероятность достижения информации i пользователем u следующим образом:

$$P_I(u, i) = \sum_{\substack{u_a \in U \\ (u_a, h_a) \in U_H}} P_U(u, u_a) \cdot P_H(h_a, i).$$

Заключение

Благодаря введенным определениям и объектам, мы можем (в зависимости от построенной модели) исследовать возможности атак, вероятности их успешной реализации, затраты ресурсов злоумышленника, а также дать агрегированную оценку степени защищенности интересующего нас комплекса от социо-инженерных атак.

Предложенный в работе вероятностно-реляционный подход позволяет не только весьма удачно формализовать «на бумаге» рассматриваемую модель, используя классический инструментарий реляционных алгебр, теории вероятностей, теории нечеткости (и, на самом деле, теорий иных степеней доверия), но и открывает перспективы непосредственно воспользоваться указанной формализацией, сведя модели очень сложных сущностей (комплекса «информационная система — персонал — критичные документы», взаимодействия этого комплекса и злоумышленника, набора возможных атак и др.) к системе таблиц в реляционной базе данных, а вопросы организации и — что особенно важно — ускорения вычислений оценки степени защищенности — к формированию SQL-запросов, которые оптимизируются современными СУБД самостоятельно.

Наконец, необходимо отметить, что хотя представленная вероятностно-реляционная модель сама по себе является гибридной, она допускает свое дальнейшее развитие в сторону применения методов теории нечеткости и теорий, обслуживающих другие степени доверия.

Литература

1. *Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Тулупьев А.Л.* Развитие методов и моделей анализа защищенности информационных систем от социоинженерных атак на основе применения реляционно-алгебраических представлений и алгоритмов // VII Санкт-Петербургская межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2011)» (Санкт-Петербург, 26–28 октября 2011 г.) Материалы конференции. СПб.: СПОИСУ, 2011. С. 160–161.
2. *Азаров А.А., Тулупьева Т.В., Тулупьев А.Л., Пащенко А.Е.* Создание программного комплекса для анализа защищенности информационных систем с учетом человеческого фактора. // Современные информационные технологии и ИТ-образование. Сборник научных трудов VI Международной научно-практической конференции. М: МГУ. 2011. С. 470–477.
3. *Ванюшичева О.Ю.* Прототип комплекса программ для построения профиля психологически обусловленных уязвимостей пользователя. Дипломная работа. СПб.: СПбГУ, 2012.
4. *Зельтерман Д., Суворова А.В., Пащенко А.Е., Мусина В.Ф., Тулупьев А.Л., Тулупьева Т.В., Гро Л.Е., Хаймер Р.* Диагностика регрессионных уравнений в анализе интенсивности рискованного поведения по его последним эпизодам // Труды СПИИРАН. 2011. Вып. 17. С. 33–46.
5. *Котенко И.В., Юсупов Р.М.* Перспективные направления исследований в области компьютерной безопасности. Защита информации. Инсайд. 2006. № 2. С. 46.

6. *Пащенко А.Е., Тулупьев А.Л., Суворова А.В., Тулупьева Т.В.* Сравнение параметров угрозаобразующего поведения в разных группах на основе неполных и неточных данных // Труды СПИИРАН. 2009. Вып. 8. СПб.: Наука, 2009. С. 252–261.
7. *Петренко С.А.* Возможная методика построения системы информационной безопасности предприятия. // URL: <http://bre.ru/security/13985.html> (дата обращения 10.01.12)
8. *Пинский М.Я., Сироткин А.В., Тулупьев А.Л., Фильченков А.А.* Повышение быстродействия алгоритма оценки наблюдаемой последовательности в скрытых марковских моделях на основе алгебраических байесовских сетей // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2011. Вып. 5. С. 69–73.
9. *Сироткин А.В., Тулупьев А.Л., Фильченков А.А., Пащенко А.Е., Тулупьева Т.В., Мусина В.Ф.* Особенности вероятностных графических моделей комплекса «Информационная система — персонал» для оценки его защищенности от социоинженерных атак // Научная сессия НИЯУ МИФИ-2011. (1–5 февраля 2011 г., Москва.) Аннотации докладов. В 3 т. Т. 3: Стратегические информационные технологии в атомной энергетике и промышленности. Проблемы информационной безопасности в системе высшей школы. Экономические и правовые проблемы инновационного развития атомной отрасли. Образование в Национальном исследовательском ядерном университете. М.: НИЯУ МИФИ, 2011. С. 80.
10. *Степашкин М.В.* Модели и методика анализа защищенности компьютерных сетей на основе построения деревьев атак: Дис. канд. техн. наук: СПб.: СПИИРАН, 2002. 196 с.
11. *Суворова А.В., Тулупьев А.Л., Пащенко А.Е., Тулупьева Т.В., Красносельских Т.В.* Анализ гранулярных данных и знаний в задачах исследования социально значимых видов поведения // Компьютерные инструменты в образовании. №4. 2010. С. 30–38.
12. *Суворова А.В., Пащенко А.Е., Тулупьева Т.В.* Оценка характеристик сверхкороткого временного ряда по гранулярным данным о рекордных интервалах между событиями // Труды СПИИРАН. 2010. Вып. 12. С. 170–181.
13. *Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В.* Социально-психологические факторы, влияющие на степень уязвимости пользователей автоматизированных информационных систем с точки зрения социоинженерных атак // Труды СПИИРАН. 2010. Вып. 1 (12). С. 200–214.
14. *Тулупьев А.Л., Азаров А.А., Пащенко А.Е.* Информационные модели компонент комплекса «Информационная система – персонал», находящегося под угрозой социоинженерных атак // Труды СПИИРАН. 2010. Вып. 3 (14). С. 50–57.
15. *Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В.* Генерализация моделей деревьев атак на случай социоинженерных атак // Научная сессия МИФИ-2011. Аннотации докладов. В 3 т. Т. 3. М.: МИФИ, 2011. С. 89.
16. *Тулупьева Т.В., Тулупьев А.Л., Азаров А.А., Пащенко А.Е.* Психологическая защита как фактор уязвимости пользователя в контексте социоинженерных атак // Труды СПИИРАН. 2011. Вып. 18. С. 74–92.
17. *Тулупьев А.Л., Фильченков А.А., Вальтман Н.А.* Алгебраические байесовские сети: задачи автоматического обучения // Информационно-измерительные и управляющие системы. 2011. № 11, т. 9. С. 57–61.

18. *Фильченков А.А., Тулупьев А.Л.* Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. Вып. 2. С. 65–74.
 19. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Структурный анализ клик максимальных графов смежности алгебраических байесовских сетей // Вестн. Тверск. гос. ун-та. Сер.: Прикладная математика. 2011. №20. С. 139–151.
 20. *Фильченков А.А., Тулупьев А.Л.* Анализ циклов в минимальных графах смежности алгебраических байесовских сетей // Труды СПИИРАН. 2011. Вып. 2 (17). С. 151–173.
 21. *Хованов Н.В.* Общая модель измерения ценности экономических благ // Применение математики в экономике. Вып. 18 / Под. ред. Воронцовского А.В. СПб.: «ИПК «КОСТА», 2009. С.108-134
 22. *Юсупов Р., Пальчун Б.П.* Безопасность компьютерной инфосферы систем критических приложений. Вооружение. Политика. Конверсия. 2003. №2. С. 52.
 23. *Dorothy E. Denning* A Lattice Model of Secure Information Flow. // Communications of the ACM, 2008, Vol. 19, No. 5, pp. 236–243
 24. *Balepin, I., Maltsev, S., Rowe, J., Levitt, K.* Using specification-based intrusion detection for automated response. Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection, pp. 135–154 (2003)
 25. *Jahnke, M., Thul, C., Martini, P.* Graph based metrics for intrusion response measures in computer networks. LCN 2007: Proceedings of the 32nd IEEE Conference on Local Computer Networks, Washington, DC, USA, pp. 1035–1042. IEEE Computer Society, LosAlamitos (2007)
 26. *Toth, T., Krugel, C.* Evaluating the impact of automated intrusion response mechanisms. ACSAC 2002: Proceedings of the 18th Annual Computer Security Applications Conference, Washington, DC, USA, p. 301. IEEE Computer Society, Los Alamitos (2002)
-

РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММНЫХ СРЕДСТВ УПРАВЛЕНИЯ ДЕПОЗИТАРИЕМ ЭЛЕКТРОННЫХ НАУЧНЫХ ИЗДАНИЙ

Торопова А.В.

*аспирант кафедры информатики математико-механического ф-та СПбГУ,
alexandra.toropova@gmail.com*

Аннотация: Появление новых информационных технологий способствует переходу многих научных изданий к электронному виду. В связи с этим важной задачей является разработка комплекса программных средств управления депозитарием электронных научных изданий.

Введение

Современные информационные технологии предоставляют ученым массу возможностей для обмена научными и научно-техническими знаниями. С появлением интернета издатели бумажных журналов постепенно теряют свои позиции. Так как сейчас электронная публикация очень популярна благодаря высокой скорости, оперативному доступу, тому, что позволяет обращаться к более широкой аудитории, экономить средства, затрачиваемые на выпуск печатной продукции, автоматизировать процессы издательства и многое другое. Ученый получил возможность одним нажатием кнопки перейти от аннотации, индекса цитирования, фамилии автора или ключевого слова к полному тексту статьи, и его все в меньшей степени интересует, в каком журнале опубликована статья и в какой библиотеке находится этот журнал. В связи с этим происходит перенос деятельности ученых и научных организаций в онлайн. Традиционные печатные формы опубликования научных работ отходят на второй план, и, возможно, в не таком далеком будущем исчезнут вовсе.

Сейчас все большему количеству ученых намного проще воспользоваться электронными источниками, которые можно быстро и не отходя от рабочего места найти с помощью интернета, чем пойти в библиотеку в поисках необходимой для исследования литературы или, например, выписать нужный научный журнал и дожидаться, когда им его пришлют, поэтому они осуществляют подбор материалов для своих исследований с помощью традиционных поисковых механизмов в интернете, а также (у кого есть доступ), с помощью таких электронных платформ как Web of Science [16], SCOPUS [14], MathNet.ru [10] и др.

Многие ученые, в том числе и довольно известные, параллельно обычным публикациям в традиционной форме выкладывают свои работы в электронном виде в интернете. Такое решение обусловлено скоростью публикации, доступностью большому числу читателей, увеличению цитируемости,

возможностью расширения и дополнения материалов, а также защитой от недобросовестных рецензентов, так как уже известны случаи, когда после отклонения или задержки статьи похожие результаты появлялись у других авторов [1].

Для публикации результатов своих исследований и ознакомлением с работами коллег чаще всего используются научные журналы, и в первую очередь электронные [1]. Электронные журналы можно разделить на три группы:

- параллельные электронные журналы — электронные версии традиционных печатных изданий. Печатная и электронная версии — идентичны;
- интегрированные электронные журналы — издаются в двух видах, которые дополняют друг друга;
- оригинальные электронные журналы — издаются только в электронном виде.

Сейчас практически все бумажные издания на западе имеют электронные версии, а большое количество журналов не имеет печатных аналогов. В нашей стране также видна тенденция перехода к электронной публикации научных изданий.

Задача

Одной из задач работы является формализация и стандартизация требований, вызываемых спецификой электронной публикации в России, так как в мире представлено множество решений этой задачи, но не учитываются факторы такой деятельности в нашей стране, например, различные виды апробации научной документации. Под апробацией понимаются самые разные виды одобрения: рекомендации организаций, отзывы научных руководителей, рецензии, протоколы, письма поддержки, решения ученых советов, акты разнообразных экспертиз и проч.

Разрабатываемый депозитарий должен быть удобен в использовании и понятен для всех участников процессов, связанных с изданием электронных научных изданий, а именно содержать удобные инструменты для регистрации, создания, хранения и обработки необходимых документов.

Обзор существующих подходов

Управление электронным журналом, а также процессами, связанными с этим, — это довольно сложная задача, требующая взаимодействия и обмена информацией между многими участниками издательских процессов, поэтому в качестве помощи издателям таких журналов существует большое число средств автоматизации издательских процессов. В мире получает все большее распространение движение за Открытый Доступ (Open Access) [3,

11, 15], смысл которого заключается в предоставлении ученым возможности бесплатного доступа к научным публикациям. Одним из решений для Открытого Доступа является использование программного обеспечения для электронной публикации с открытым кодом. Таких систем довольно много (Digital Publishing System [6], GNU EPrints [8] и др.) [5], одна из самых популярных таких систем Open Journal Systems (OJS) [7], появившаяся в ходе Проекта Общественных Знаний (Public Knowledge Project [13]) по разработкам и исследованиям. OJS была создана для того, чтобы свести к минимуму время и энергию, тратящиеся на канцелярские и управленческие задачи, связанные с ведением журнала, в то же время способствуя эффективности процессов, проходящих в редакции, и улучшению их учета. OJS – это одновременно журнал, веб-сайт и издательство. OJS охватывает все аспекты онлайн-публикации журнала, включая его создание, процесс подачи статей, рецензирование, редактирование, публикацию, архивирование и индексирование журнала. Кроме того, она переведена на большое количество языков (среди которых есть и русский), что позволяет публикацию больше, чем на одном языке. [7, 12]

На западе инфраструктура электронной научной публикации уже во многом сформировалась [2], успешно существуют тысячи электронных научных журналов, в то время как в России она еще на начальных стадиях развития: например, в информационном бюллетене электронных научных изданий Информрегистр [4] насчитывается всего лишь 146 изданий.

Заключение

В данной работе была сформулирована задача по созданию комплекса программных средств управления депозитарием электронных научных изданий. Эта задача крайне важна в связи с тенденцией перехода к электронному виду многих российских журналов, а также появления новых электронных изданий.

Литература

1. *Богданова И.Ф.* Онлайнное пространство научных коммуникаций // Социология науки и технологий. 2010. № 1.
2. *Веселаго В.Г., Елизаров А.М., Сюттюренко О.В.* Российские электронные научные журналы — новый этап развития, проблемы интеграции // Электронные Библиотеки. 2005. Т. 8, № 1.
3. *Полянин, А.Д.* Электронные публикации и основные физико-математические ресурсы Интернета / А.Д. Полянин, А.И. Журов [Электронный ресурс]. – Режим доступа: <http://eqworld.ipmnet.ru/ru/info/sci-edu/PolyaninZhurov2007.htm/>.
4. ФГУП НТЦ Информрегистр — Электронный ресурс: <http://inforeg.ru/eni>.
5. *Szyk M., Choudhury S.* A Survey and Evaluation of Open-Source Electronic Publishing Systems. JScholarship. April 2008.

6. Digital Publishing System — Электронный ресурс: <http://dpubs.org/>.
 7. *Edgar, B.D. & Willinsky, J.* (in press). A survey of the scholarly journals using Open Journal Systems. Scholarly and Research Communication.
 8. GNU EPrints — Электронный ресурс: <http://www.eprints.org/>.
 9. *Johnson R.* Open Access: Unlocking the Value of Scientific Research (2004).
 10. MathNet.ru — Электронный ресурс: <http://www.mathnet.ru/>.
 11. Noorden R. Open access: The true cost of science publishing. Nature News. 2013/.
 12. Open Journal Systems — Электронный ресурс: <http://pkp.sfu.ca/?q=ojs>.
 13. Public Knowledge Project— Электронный ресурс: <http://pkp.sfu.ca/>.
 14. SCOPUS — Электронный ресурс: <http://www.scopus.com/home.url>.
 15. *Smecher A.* The Future of the Electronic Journal // NeuroQuantology. 2008. V. 6. № 1. С. 1–6.
 16. Web of Science — Электронный ресурс: http://thomsonreuters.com/products_services/science/science_products/a-z/web_of_science/.
-

АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ КОЛЛ-ЦЕНТРА: КОМПЛЕКС ХРАНИМЫХ ПРОЦЕДУР

Лепилина А.П.

студентка математико-механического ф-та СПбГУ, lap-7@mail.ru

Аннотация: В докладе рассматривается задача автоматизации деятельности сотрудников колл-центра. Описаны разработанные хранимые процедуры, позволяющие формировать автоматически итоговый отчет по проекту, а также формирующие txt-файл для дальнейшего импорта в ПО Infra Call Center. Кроме того, предложен сценарий для создания окна, в котором отображается информация из разработанной базы данных, соответствующая проекту.

Введение

В настоящее время многие компании размещают информацию о своем продукте или о своих услугах на собственных сайтах в интернете, рассчитывая привлечь как можно больше клиентов. Пользователю, зашедшему на сайт, представляется список товаров и услуг, их описание, а также возможность сделать заказ. Несмотря на то, что технологии всегда двигаются вперед, одним из основных и доверительных способов общения продавец-покупатель остается телефон [3]. Это означает, что телемаркетинг, колл-центры востребованы.

Центры обработки вызовов клиентов развивались в течение 20-го столетия, начав с простых агентов-операторов, которые отвечали на телефонные звонки [3]. Впоследствии центры все больше расширялись, и в них стали использоваться модернизированные средства телекоммуникаций. Сегодня подобным центрам придается особое значение. Они не только оптимизируют работу с клиентами, но и позволяют продвигать компанию. Так, по исследованию университета Пердью, 92% формируют свое мнение о компании на основании работы ее call-центра. При этом, по данным Gartner Group, 68% клиентов могут сменить бренд в связи с плохой работой центра [8].

Несмотря на важность центра обработки вызовов, многие компании предпочитают не иметь в своем штате подобного департамента и пользуются услугами аутсорсинга. Аутсорсинг центров весьма активно развивается сегодня. Так, по данным MetLife, только 20–25% персонала call-центров являются работниками самих компаний, остальная часть рынка — это арендуемые центры. По оценкам агентства Datamonitor, рынок аутсорсинга call-центров в Европе увеличивается на 15–20% ежегодно. Лидером по количеству операторов в компаниях, предлагающих услуги аутсорсинга, выступает Великобритания, на втором месте — Германия, третье место разделяют Испания и Франция. Впрочем, используется не только аутсорсинг, но и

так называемые домашние центры, в которых используется труд надомных работников, отвечающих на звонки. Согласно исследованию Yankee Group, в 350 канадских и американских call-центрах 24% агентов, то есть 672 тыс. человек работают из дома [8].

Программное обеспечение, которое позволяет автоматизировать процессы центров обслуживания клиентов, в последние несколько лет динамично развивается [11]. Кроме того, на сегодняшний день на рынке идет сращение call-центров с контакт-центрами, которые дают более широкий спектр возможностей для обслуживания связи с клиентами.

Контакт-центры дают возможность интегрировать все современные способы связи с клиентом — не только через фиксированные или мобильные телефонные линии, но и посредством SMS, факса, электронной почты, чатов, интернет-пейджеров и т. д.

В докладе рассматриваются разработанные процедуры и функции для автоматизации ряда аспектов работы сотрудников колл-центра.

Описание предметной области

Введем базовое определение. Термин колл-центр (от англ. call center) — может употребляться в следующих значениях [7, 11]:

1. Операторский центр обработки входящих и исходящих звонков.
2. Программно-аппаратный комплекс, позволяющий управлять входящими и исходящими звонками.
3. Контактный центр (контакт-центр) для обработки обращений по всем известным каналам связи.

Мы будем рассматривать колл-центр как операторский центр обработки звонков (п.1 определения).

Центр обслуживания звонков производит администрирование входящего потока запросов клиентов или потенциальных потребителей по поддержке продукта (услуги). Также могут проводиться исходящие звонки по телемаркетингу, клиентские звонки, обслуживание своего продукта (услуги) и коллекторские мероприятия по сбору долгов. Помимо услуг центра обслуживания звонков, можно управлять потоком корреспонденции (писем, факсов, чатов, электронной почты) в одном месте, контакт-центре.

Описание разработки

Разработанные процедуры позволяют автоматизировать ряд действий оператора и/или координатора при работе над проектами в программном обеспечении Infra Call Center [1, 2].

Infra Call Center — это программный комплекс, позволяющий организовывать центры телефонного обслуживания (колл-центры) для обработки большого количества входящих и исходящих звонков [1, 2].

Кроме возможностей автоматической обработки и распределения потоков входящих вызовов, Infra Call Center предоставляет широкие возможности для проведения исходящих кампаний. Исходящая кампания — это обзвон внешних абонентов по списку номеров — например, если нужно уведомить их о предстоящем событии, провести опрос или маркетинговую акцию.

Одним из первых этапов при выполнении дипломного проекта стало проектирование и последующая разработка базы данных для организации и хранения поступающей информации в среде MS SQL Server [4, 6, 9, 13]. База данных включает в себя информацию о юридических и физических лиц, а также результаты, полученные операторами при обзвоне.

Для автоматизации работы операторов было создано на языке программирования PHP [10, 12] окно со сценарием, в котором они видят текст-подсказку для будущего разговора с абонентом и в котором после разговора оператор отмечает конечные результаты в заранее обговоренной форме.

Работа скрипта заключается в следующем:

- получить номер последнего окна (назовем его X),
- создать в базе call_center таблицу reportX+1,
- назвать файл X+1.php,
- в файле X+1.php добавить номера окна.

В окне в самом конце сценария есть выбор результата разговора. Чтобы добавить результат, который в дальнейшем может выбрать оператор, и была написана хранимая процедура. Дальнейшие шаги (открытие окна у оператора, импортирование данных в ПО Infra Call Center) также были автоматизированы, для этого были разработаны соответствующие процедуры.

Формирование отчета по проекту также было автоматизировано. Данные, которые заносит оператор в раннее созданное окно, в результате выполнения написанного скрипта попадают в таблицу в базе данных, а затем формирование итогового отчета по проекту производится разработанной хранимой процедурой. Отчет предоставляется координатору в формате .xls (MS Excel) [5].

Заключение

Каждая компания хочет занять лидирующую позицию на рынке, обогнав своих конкурентов, минимизировав при этом расходы, трудовые ресурсы и, конечно же, время. Перед компаниями стоят разные цели и задачи. Но если речь идет о поиске новых клиентов, маркетинговом исследовании, актуализации баз данных, контроле качества, работе с дебиторской задолженностью, приглашением на мероприятия, предвыборной агитацией, то оправдано обращение в колл-центр.

Чтобы колл-центр был способен обеспечить результативную, стабильную работу и качественное оказание вышеперечисленных услуг, нужно иметь надежное программное обеспечение, квалифицированную техническую поддержку.

В докладе представлен результат работ по автоматизации ряда аспектов деятельности колл-центра. Была разработана база данных для хранения необходимых сведений по проекту, разработан комплекс хранимых процедур, написан сценарий для окна-подсказки оператору, разработана процедура для автоматического формирования отчетов по каждому проекту.

Благодаря разработанной системе автоматизации деятельности колл-центра, операторам не требуется больше вручную набирать телефонный номер, так как звонки приходят автоматически. Координаторам не надо вручную сводить полученные результаты по проекту, так как отчет им предоставляется в результате выполнения хранимой процедуры.

Литература

1. *Infra Call Center: Руководство администратора*. М.: INFRATEL, Inc, 2011. 251 с. URL: <http://www.infratel.ru/> (дата обращения 20.04.2013).
 2. *Infra Call Center: Руководство супервизора*. М.: INFRATEL, Inc, 2011. 251 с. URL: <http://www.infratel.ru/>(дата обращения 20.04.2013).
 3. *Гольдштейн Б.С., Фрейнкман В.А.* Call-центры и компьютерная телефония. СПб.: БХВ – Санкт-Петербург. 2002.
 4. *Бурков А.В.* Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008. URL: <http://www.intuit.ru/department/se/pisqlvs2008> (дата обращения 20.04.2013).
 5. *Васильев А.А., Стоцкий Ю.А., Телина И.С.* Microsoft Office 2010. Самоучитель СПб.: Питер, 2011. 432 с.
 6. *Виэйра Р.* Программирование баз данных Microsoft SQL Server Изд.: «Диалектика», 2008. 1072 с.
 7. *Вольский В.В. Веселова О.В. Золкина Н.К.* Оператор call-центра: от найма до увольнения Тверь: Софитель, 2008. 200 с.
 8. Все о колл-центрах URL: <http://callcenter.vipcentr.ru> (дата обращения 20.04.2013).
 9. *Полякова Л.Н.* Основы SQL. URL: <http://www.intuit.ru/department/database> (дата обращения 03.10.2010).
 10. Почему PHP? Чем и почему PHP лучше? [Электронный ресурс] / oracle, // <http://web-site-lessons.com.ua/php/why_PHP.html>, (по состоянию на 27.03.2013).
 11. Российский рынок аутсорсинговых call-центров: все дальше в регионы М.: Рос-БизнесКонсалтинг, 2012. 50 с.
 12. *Рябов В.А., Несвижский А.И.* Современные веб-технологии // Проектирование баз данных и работа с ними веб-приложений. Введение в БД, SQL Server, ADO.NET. URL: <http://www.intuit.ru/department/internet/mwebtech> (дата обращения 03.10.2012).
 13. *Уолтерс Р., Рей Р., Коулс М., Феррачати Ф., Фармер Д.* SQL Server 2008: ускоренный курс для профессионалов М.:ООО «И.Д. Вильямс», 2009. 768 с.
-

ПРОГРАММА ДЛЯ АНАЛИЗА ЗАЩИЩЕННОСТИ ОТ СОЦИОИНЖЕНЕРНЫХ АТАК КРИТИЧНЫХ ДОКУМЕНТОВ С УЧЕТОМ СТРУКТУРЫ СВЯЗЕЙ МЕЖДУ ПОЛЬЗОВАТЕЛЯМИ ИНФОРМАЦИОННОЙ СИСТЕМЫ¹

Хамаева Э.А.

студентка кафедры информатики СПбГУ, elvirakhamaeva@hotmail.com

Аннотация: Мы живем во время бурного развития технологий безопасности и не секрет, что для информационных грабителей все более заманчивыми становятся именно методы социальной инженерии. Эти методы используют не «дыры» в защитных технологиях, а человеческие слабости. Часто получается так, что компания, которая делает ставку на высокотехнологичные системы, оказывается совершенно беззащитной перед социоинженером. Доклад посвящен программной реализации алгоритма анализа защищенности от социоинженерных атак критичных документов информационной системы с учетом структуры связей между пользователями.

Введение

Мы живем во время бурного развития технологий безопасности и не секрет, что для информационных грабителей все более заманчивыми становятся именно методы социальной инженерии. Эти методы используют не «дыры» в защитных технологиях, а человеческие слабости. Мастера промышленного шпионажа пошли по пути наименьшего сопротивления, находя скрытые и, вместе с тем, простые способы достижения цели. Часто получается так, что компания, которая делает ставку на высокотехнологичные системы, оказывается совершенно беззащитной перед социоинженером [1].

Именно сотрудники являются самым слабым звеном на пути злоумышленника к получению конфиденциальной информации. Поэтому требуется изучать, выявлять и предотвращать социоинженерные атаки, направленные на пользователя, а также обеспечить возможность оценки степени защищенности персонала автоматизированных информационных систем от социоинженерных атак.

Ранее был разработан программный комплекс, анализирующий защищенность пользователей информационных систем от социоинженерных атак на основании профиля уязвимостей пользователя.

Исходя из степени выраженности уязвимостей, содержащихся в профиле уязвимостей пользователя, удалось составить вероятностные оценки устойчивости пользователя к социоинженерным атакующим воздействиям.

¹ Работа поддержана грантом РФФИ 10-01-00640-а и грантом СПбГУ 6.38.72.2011 Мероприятие-2, 2011-2013 гг.

Таким образом, были получены оценки защищенности конкретного пользователя от социоинженерных атак без возможности моделирования общей защищенности информационной системы.

Доклад посвящен разработке программного комплекса, реализующего алгоритм анализа защищенности от социоинженерных атак критичных документов информационной системы с учетом структуры связей между пользователями.

Алгоритм анализа защищенности информационной системы с учетом структуры связей между пользователями

К рассмотрению предлагается подход, построенный на объединенной вероятности успешности элементарных социоинженерных атакующих воздействий злоумышленника.

Рассмотрим комплекс «информационная система — персонал — критические документы». Для того, чтобы анализировать защищенность пользователей информационных систем, рассмотрим связи между пользователями. Под связями понимаются различные отношения между людьми в организации. При имитации социоинженерных атакующих воздействий злоумышленника предполагается использование данных связей в следующем контексте: в случае, если пользователь оказался подвержен СИ-атакующему воздействию злоумышленника, и в то же время на доступных ему устройствах не обнаружилась требуемая злоумышленнику информация, то происходит переход по связям пользователя к другим пользователям. На новых пользователях будет осуществлено новое СИ-атакующее воздействие. При этом рассматривается также вероятность перехода по связям пользователя. Предполагается, что каждый тип связи обладает собственным «весом», то есть вероятностью того, что злоумышленник сможет воспользоваться этой связью и применить СИ-атакующее воздействие к новому пользователю. В качестве связей было рассмотрено следующее множество: Знакомые, Друзья, Любовь, Родственники, Враги. Здесь приведен пример возможных значений. Типов связи может быть больше, и они могут быть иные.

В разработанном алгоритме происходит поиск пути наиболее простого развития СИ-атаки с точки зрения злоумышленника. На очередном шаге алгоритма, когда осуществляется переход к новому пользователю, поиск пользователя производится путем поиска максимальной общей вероятности успешности элементарных СИ-атакующих действий злоумышленника. Поиск осуществляется до того момента, как будет получен пользователь, свойства которого указывают на наличие требуемой злоумышленнику конфиденциальной информации. Если такой пользователь не найдет, то считается, что данная конфиденциальная информация защищена от СИ-атак, направленных на пользователей информационных систем.

Программная реализация рассмотренного алгоритма

Внедрение данного алгоритма в разработанный ранее программный комплекс (его рассмотрению были посвящены статьи [2–8]) позволило обеспечить автоматизацию анализа защищенности пользователей информационных систем от СИ-атакующих действий злоумышленника. Реализовано текстовое и графическое представление итогов СИ-атаки. Текстовое меню представлено на рисунке 1.

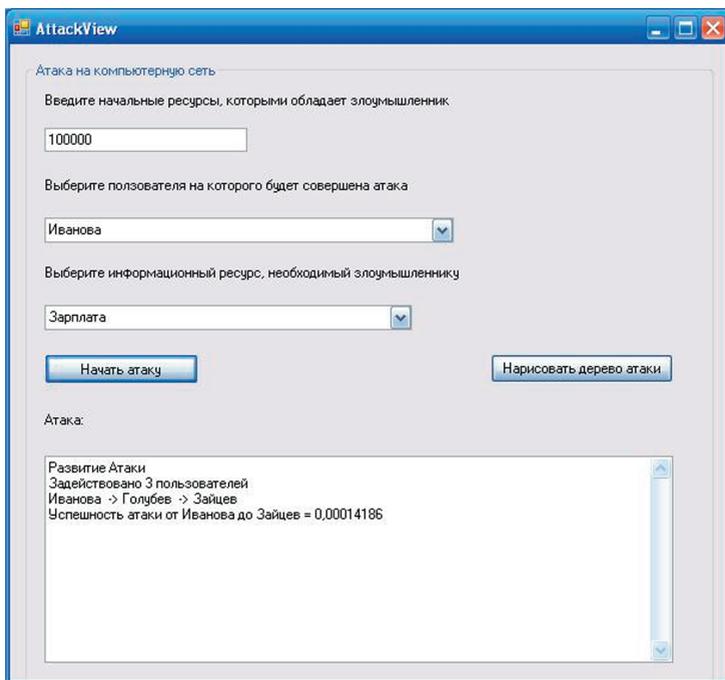


Рис. 1. Вывод результатов СИ-атаки на пользователей информационных систем

Результат СИ-атаки находится в текстовом поле. Надпись «Задействовано 3 пользователей» означает, что конфиденциальная информация, требуемая злоумышленнику, найдена и для ее получения будут задействованы 3 человека. Первым звеном в СИ-атаке злоумышленника является пользователь с никнеймом «Иванова», далее по связям переходим к «Голубеву», информация была найдена на устройствах, доступных пользователю с никнеймом «Зайцев». Надпись «Успешность атаки от Иванова до Зайцев = 0,00014186» означает, что злоумышленник получит доступ к необходимой ему информации с вероятностью 0,00014186.

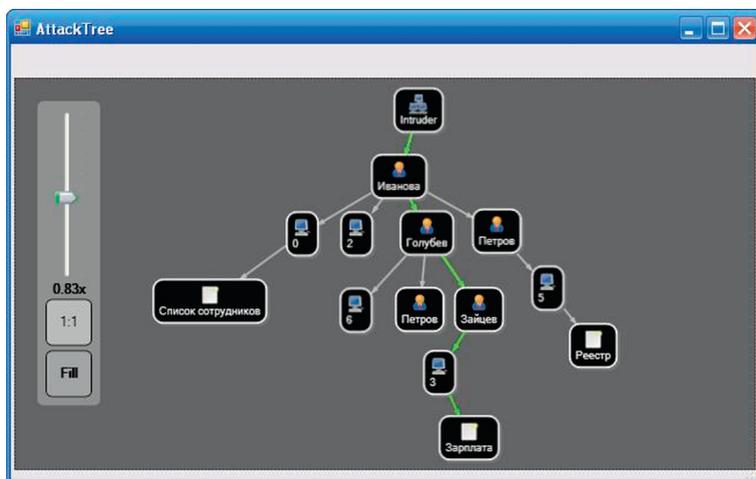


Рис. 2. Визуализация результатов СИ-атаки на пользователей информационных систем

На рисунке 2 представлена визуализация данной СИ-атаки с выводом всех цепочек связанных с злоумышленником пользователей. Зеленым цветом выделен путь от Злоумышленника до необходимой ему конфиденциальной информации. Для каждого пользователя информационной системы можно из контекстного меню вызвать подробное описание уязвимостей и степень проявления этих уязвимостей (Рисунок 3).

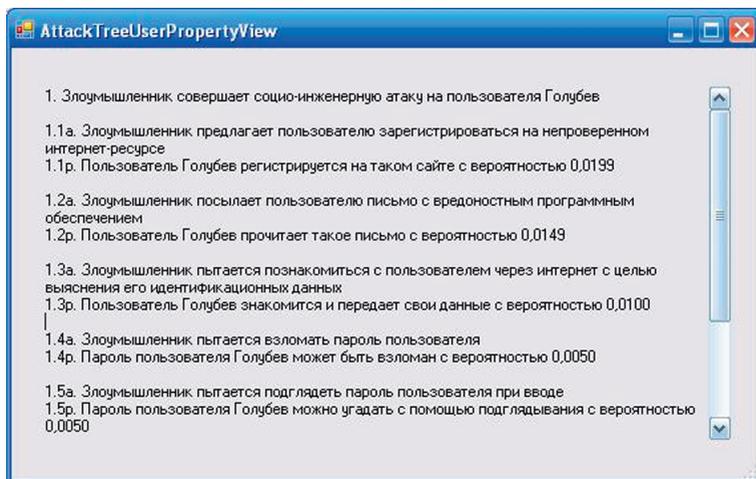


Рис. 3. Степень проявления уязвимостей для пользователя «Голубев»

Заключение

В докладе рассмотрены ключевые шаги алгоритма анализа резистентности пользователей от элементарных СИ-атакующих воздействий злоумышленника. В данном подходе учитывается степень выраженности уязвимостей пользователя, а также веса связей (вероятности успешного перехода) между пользователями.

Полученный программный продукт позволяет автоматизировать анализ защищенности пользователей информационных систем от СИ-атак с применением анализа степени выраженности уязвимостей пользователя, содержащихся в частичном профиле уязвимостей пользователя, построенном на психологических особенностях личности, учитывающий структуру связей между пользователями.

В то же время данный программный продукт позволит получить временные оценки работы алгоритма, что позволит адаптировать его работу ради ускорения вычислений всего программного комплекса.

Литература

1. Митник К.Д., Саймон В.Л. Искусство обмана. М.: Компания АйТи, 2004. 360 с.
2. Азаров А.А., Тулупьева Т.В., Фильченков А.А., Тулупьев А.Л. Вероятностно-реляционный подход к представлению модели комплекса «Информационная система — персонал — критичные документы». // Труды СПИИРАН. 2012. Вып. 20. С. 57–71.
3. Азаров А.А., Тулупьева Т.В., Тулупьев А.Л. Прототип комплекса программ для анализа защищенности персонала информационных систем построенный на основе фрагмента профиля уязвимостей пользователя. // Труды СПИИРАН. 2012. Вып. 21. С. 21–40.
4. Ванюшичева О.Ю. Прототип комплекса программ для построения профиля психологически обусловленных уязвимостей пользователя. Дипломная работа. СПб.: СПбГУ, 2012.
5. Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В. Социально-психологические факторы, влияющие на степень уязвимости пользователей автоматизированных информационных систем с точки зрения социоинженерных атак // Труды СПИИРАН. 2010. Вып. 1 (12). С. 200–214.
6. Тулупьев А.Л., Азаров А.А., Пащенко А.Е. Информационные модели компонент комплекса «Информационная система — персонал», находящегося под угрозой социоинженерных атак // Труды СПИИРАН. 2010. Вып. 3 (14). С. 50–57.
7. Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В. Генерализация моделей деревьев атак на случай социоинженерных атак // Научная сессия МИФИ-2011. Аннотации докладов. В 3 т. Т. 3. М.: МИФИ, 2011. С. 89.
8. Тулупьева Т.В., Тулупьев А.Л., Азаров А.А., Пащенко А.Е. Психологическая защита как фактор уязвимости пользователя в контексте социоинженерных атак // Труды СПИИРАН. 2011. Вып. 18. С. 74–92.

ВИЗУАЛИЗАЦИЯ КОНТРОЛИРУЕМЫХ ЗОН ПРИ АНАЛИЗЕ СТЕПЕНИ ЗАЩИЩЕННОСТИ КРИТИЧНЫХ ДОКУМЕНТОВ ОТ СОЦИО-ИНЖЕНЕРНЫХ АТАК С УЧЕТОМ СТРУКТУРЫ СВЯЗЕЙ МЕЖДУ ПОЛЬЗОВАТЕЛЯМИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Михайлов Н.Ю.

студент кафедры информатики СПбГУ, nikmih87@gmail.com

Аннотация: В статье представлен обзор двух основных подходов к разрешению проблемы избыточности прав пользователей в информационной системе, а также описание программного модуля визуализации контролируемых зон, разработанного в качестве компоненты программного комплекса для анализа степени защищенности критичных документов от социо-инженерных атак.

Введение

В наше время функционирование предприятий, как государственных, так и частных, немыслимо без автоматизации бизнес-процессов. Каждая организация имеет собственную информационную систему, собранную на основе комплексных решений, широко представленных на информационно-технологическом рынке, или разработанную по индивидуальному заказу под конкретные задачи предприятия. Вместе с тем, привлечение подобных ресурсов выдвигает ряд требований к обеспечению безопасности конфиденциальных данных.

На сегодняшний день, выявление достаточных для работы с критичной информацией прав пользователя — наиболее распространенный среди специалистов в информационной безопасности метод противодействия инсайдерским и социо-инженерным атакам. Избыточность прав пользователей, работающих с критичной информацией, является основным источником злоупотребления пользователями ИС своими правами с целью получения конфиденциальной информации компании и передачи ее злоумышленнику.

Целью доклада является обзор двух основных подходов к разрешению проблемы избыточности прав пользователей в информационной системе, а также описание программного модуля визуализации контролируемых зон, разработанного для программного комплекса для анализа степени защищенности критичных документов от социо-инженерных атак.

Разграничение прав доступа в информационной системе

Все подходы к разрешению проблемы избыточности прав связаны с анализом текущей деятельности пользователя и выявлением аномального

поведения пользователя относительно его предшествующего поведения в информационной системе. Таким образом, предлагаемые подходы позволяют выявить совершаемое деструктивное поведение пользователя в информационной системе и прекратить его. Процедуры выявления деструктивного поведения пользователя и прекращения подобных действий требуют больших информационных и организационных ресурсов. В то же время можно уменьшить количество затрачиваемых ресурсов, если проводить профилактику подобных действий.

Проблема предотвращения несанкционированного использования защищаемых ресурсов критически важных инфраструктур решается двумя основными путями: разграничением доступа и организационными мерами. На основе учёта информационных профилей пользователей и в соответствии с возложенными на них задачами среди пользователей выделяются группы по правам доступа к информационным ресурсам. Территория предприятия делится на контролируемые зоны. Оговаривается круг лиц, имеющих доступ к аппаратным средствам, находящимся внутри контролируемой зоны. Применение такого подхода приводит к снижению возможностей пользователя по совершению атипичных действий. Кроме того, данные подходы позволяют сравнительно легко выявить пользователей, замешанных в утечке информации.

Визуализация контролируемых зон информационной системы

Для наглядного представления контролируемых зон и визуализации дифференциации пользователей по правам доступа к ресурсам информационной системы, в рамках более широких научно-исследовательских проектов, поддержанных РФФИ, СПбГУ и Правительством Санкт-Петербурга, был создан программный модуль визуализации контролируемых зон информационной системы.

Программный модуль позволяет визуализировать контролируемые зоны информационной системы для наглядного представления структуры компьютерной сети компании и дифференциации пользователей по правам доступа к аппаратным средствам.

Данные об аппаратных средствах и пользователях, а также о связях между компонентами информационной системы извлекаются из базы данных, разработанной в рамках проекта по созданию программного комплекса для анализа защищённости критичных документов от социо-инженерных атак.

Пользователю программного модуля доступны инструменты для визуализации отдельных контролируемых зон в виде графа, в котором вершинами являются аппаратные средства и пользователи информационной системы, а дугами — связи между ними. Визуально различимы вершины, соответствующие пользователям из различных групп по правам доступа к информационным ресурсам.

Описание работы программы

При запуске программы появляется интерфейс, который позволяет пользователю начать работу (рисунок 1).

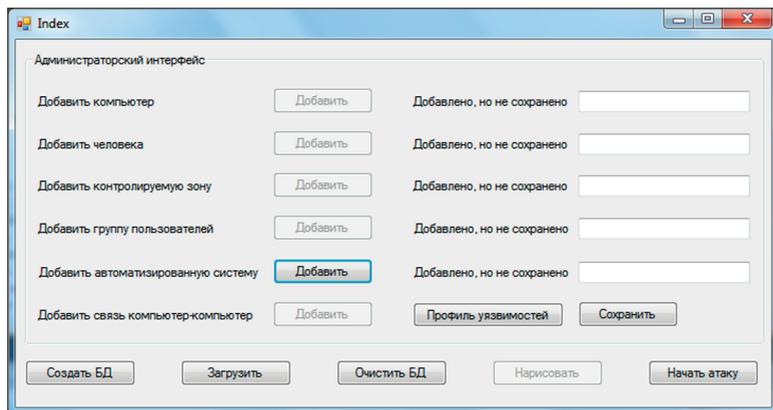


Рис. 1. Основной интерфейс приложения

Реализованы три варианта ввода данных об информационной системе:

- 1) начать ввод характеристик автоматизированной информационной системы, нажав доступную кнопку «Добавить». Для обеспечения целостности информации, передаваемой в базу данных, кнопки добавления зависимых сущностей (в данном случае это все объекты, так как объект «автоматизированная информационная система» является главным по отношению к ним) остаются недоступными до тех пор, пока не будет введена информация о главном объекте;
- 2) создать пустую базу данных и постепенно добавлять в нее информацию, нажав «Создать БД»;
- 3) загрузить ранее созданную БД.

При выборе кнопки «Загрузить» на первом шаге в программу загружаются данные из уже существующей базы данных. Впоследствии к этой базе данных можно добавлять новую информацию и записывать ее с помощью кнопки «Сохранить».

Когда с программой связана конкретная база данных, становится доступна кнопка «Очистить БД», при нажатии на которую из всех таблиц базы удаляются данные, но сами таблицы не удаляются.

При добавлении хотя бы одной платформы также становится доступна кнопка «Нарисовать». При нажатии на нее появляется новое окно (рисунок 2), в котором визуализированы текущие данные, характеризующие информационную систему.

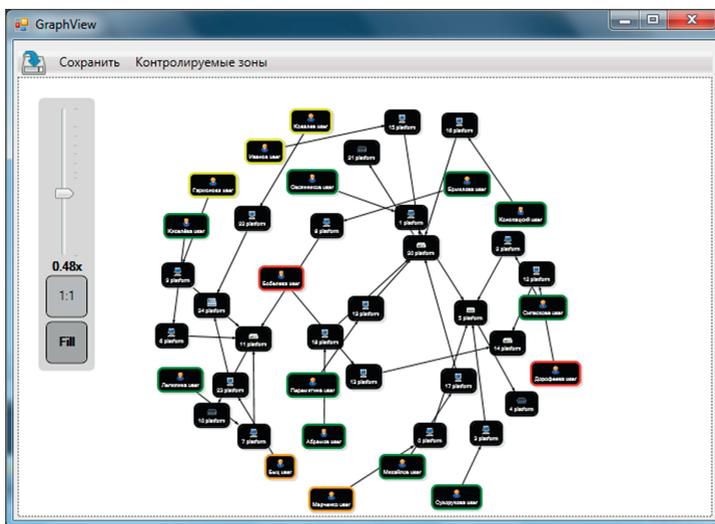


Рис. 2. Визуализация модели информационной системы

Данные представляются в виде графа, вершинами которого являются компьютеры и пользователи. Каждая вершина изображается в виде прямоугольника. Внутри прямоугольника содержится изображение, характеризующее тип вершины (пользователь это или компьютер). При этом для компьютера отображается тип платформы (L2Commutator, L3Commutator, Firewall, Workstation, ApplicationServer, WorkstationGroup или неизвестный тип). Кроме того отображается текстовое значение, которое в случае пользователя является его фамилией, а в случае компьютера — порядковым номером. Рамки прямоугольников для пользователей раскрашены в различные цвета, в соответствии с принадлежностью пользователя к определённой группе по правам доступа в информационной системе.

При помощи ребер, изображенных стрелками, указывается доступ пользователей к платформам.

Пользователь может выбирать масштаб изображения, используя меню в левой части формы, перетаскивать объекты на экране по своему усмотрению (рисунок 3). Связи при этом не будут потеряны.

При нажатии правой кнопки мыши в свободной от объектов области экрана, появляется контекстное меню (рисунок 4).

При выборе пункта «Контролируемые зоны» открывается окно управления визуализацией контролируемых зон (рисунок 5). То же самое происходит при нажатии кнопки «Контролируемые зоны» на верхней панели.

В этом окне активны кнопки для визуализации контролируемых зон, добавленных пользователем в базу данных. При нажатии на кнопку откры-

вается окно, в котором визуализирована соответствующая контролируемая зона (рисунок 6).

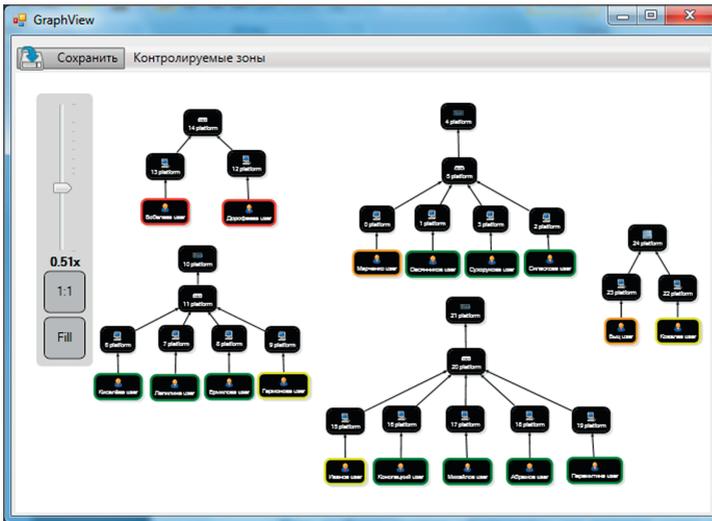


Рис. 3. Пользовательское редактирование

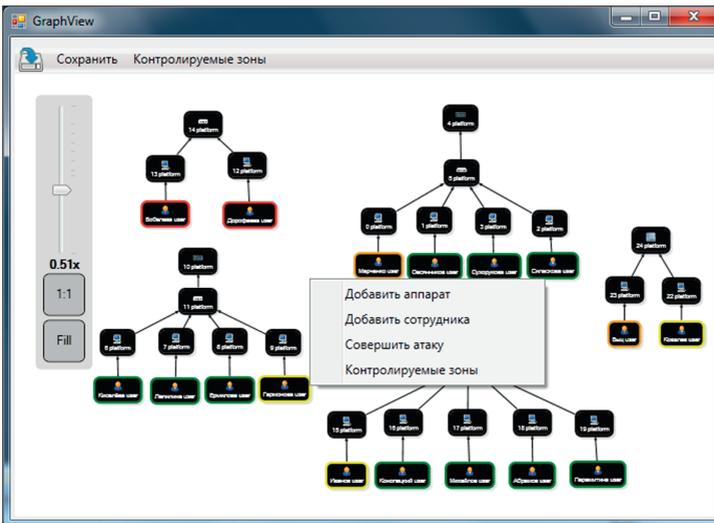


Рис. 4. Контекстное меню

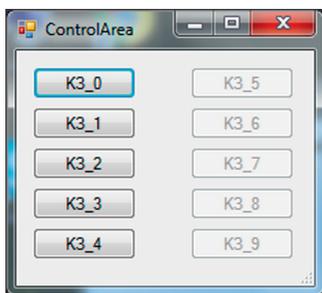


Рис. 5. Окно управления визуализацией контролируемых зон

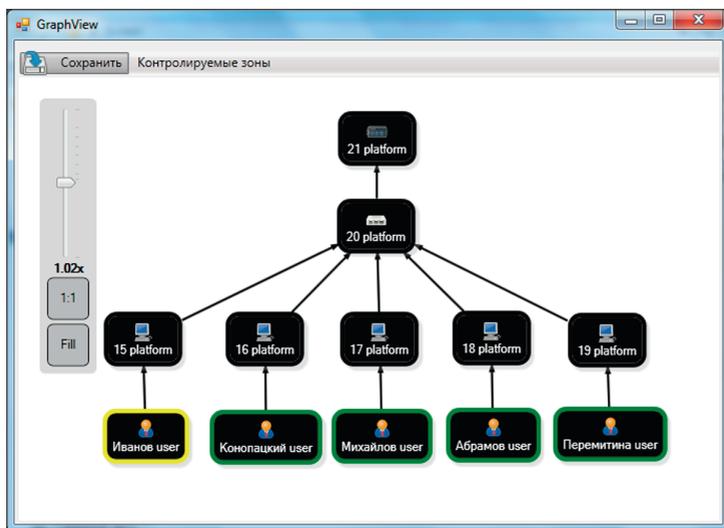


Рис. 6. Визуализация отдельной контролируемой зоны

Заключение

Программный модуль обеспечивает наглядное представление контролируемых зон информационной системы. Наряду с визуализацией модели всей информационной системы стало доступно отображение отдельных контролируемых зон, что необходимо при работе с информационными системами с большим числом пользователей и аппаратных средств. При использовании для визуализации библиотеки Graph# удалось уйти от статических изображений. Доступно ручное редактирование положения объектов информационной системы на схеме, добавление, удаление, объектов информационной

системы и связей между ними непосредственно на схеме. Визуально различимы пользователи с различными правами доступа.

Специалисты по обеспечению безопасности могут использовать данное программное обеспечение для выявления слабых мест в архитектуре информационных систем.

Литература

1. *Фролова А.Н., Пащенко А.Е., Тулупьева Т.В., Тулупьев А.Л.* Анализ уровня защищенности информационных систем в контексте социоинженерных атак: постановка проблемы // Труды СПИИРАН. 2008. Вып. 7. СПб.: Наука, 2008. С. 170–176.
 2. *Тулупьева Т.В., Тулупьев А.Л., Пащенко А.Е., Степашкин М.В.* Подход к оценке защищенности персонала автоматизированной информационной системы от социо-инженерных атак // Региональная информатика-2008 (РИ-2008). XI Санкт-Петербургская международная конференция. Санкт-Петербург, 22–24 октября, 2008 г.: Материалы конференции / СПОИСУ. СПб., 2008. С. 113–114.
 3. *Фролова А.Н., Тулупьева Т.В., Пащенко А.Е., Тулупьев А.Л.* Возможный подход к анализу защищенности информационных систем от социоинженерных атак // Информационная безопасность регионов России (ИБРР-2007). V Санкт-Петербургская региональная конференция. Санкт-Петербург, 23–25 октября 2007 г.: Труды конференции / СПОИСУ. СПб., 2008. С. 195–199.
 4. *Тулупьева Т.В., Тулупьев А.Л., Пащенко А.Е., Азаров А.А., Степашкин М.В.* Социально-психологические факторы, влияющие на степень уязвимости пользователей информационных систем, с точки зрения социоинженерных атак // Труды СПИИРАН. 2010. Вып. 1(12).
 5. *Тулупьев А.Л., Пащенко А.Е., Азаров А.А.* Информационная модель пользователя, находящегося под угрозой социоинженерной атаки // Труды СПИИРАН. 2010. Вып. 2(12).
 6. *Тулупьев А.Л., Пащенко А.Е., Азаров А.А.* Информационные модели компонент комплекса «информационная система — персонал», находящегося под угрозой социоинженерных атак. // Труды СПИИРАН. 2010. Вып. 3(13).
 7. *Eric Shaw, Ph.D., Keven G. Ruby, M.A. and Jerrold M. Post, M.D.* The Insider Threat to Information Systems.
-

ГРАНУЛЯЦИЯ НЕТОЧНЫХ ДАННЫХ В СИНТЕЗЕ БАЙЕСОВСКОЙ СЕТИ ДОВЕРИЯ ПРИ ИССЛЕДОВАНИИ ЧАСТИЧНО НАБЛЮДАЕМЫХ РЕАЛИЗАЦИЙ ПУАССОНОВСКОГО ПРОЦЕССА¹

Суворова А.В.

*м. н. с. лаборатории ТuМПИ СПИИРАН,
аспирант кафедры информатики СПбГУ, suvalv@mail.ru*

Аннотация: В работе предложен подход к построению моделей социально-значимого поведения в виде байесовской сети доверия на основе данных об эпизодах поведения, рассматриваемого как пуассоновский случайный процесс. Кроме того, приведена процедура получения выводов об интенсивности поведения на основе численного примера.

Введение

При исследовании многих процессов необходимо характеризовать поведение индивидов, вовлеченных в этот процесс, влияющих на него, оценивать параметры такого поведения. Одним из важных показателей, позволяющих дать количественное описание поведения, является интенсивность изучаемого поведения. В качестве примера можно привести рискованное поведение, связанное с риском передачи или получения неизлечимой инфекции (например, ВИЧ-инфекции) [1].

Одним из способов получения данных о поведении является опрос респондента об эпизодах социально-значимого поведения [1–4]. Такой опрос позволяет судить об интервалах между последними эпизодами угрожающего поведения, о минимальном и максимальном интервалах между эпизодами. Однако из-за особенностей припоминания таким образом можно получить сведения лишь о рекордных интервалах эпизодами поведения и о небольшом числе (3–5) последних эпизодов, что приводит к необходимости анализа неполных данных.

Описанный способ получения данных обладает рядом достоинств [1], но им невозможно собрать сведения о большом количестве эпизодов поведения одного респондента, в отличие задач, возникающих в технических системах, позволяющих производить измерения с требуемой периодичностью. Кроме того, данные об эпизодах поведения респондентов поступают на естественном языке, т. е. являются в значительной степени неточными и

¹ Доклад содержит материалы исследований, частично поддержанных грантами РФФИ 09-01-00861-а, 10-01-00640-а, 12-01-00945-а, субгрантом № M13A11589 (A06995) гранта NIH № 5 R01AA017389-04 «Alcohol and HIV Risk Reduction in St. Petersburg, R.F.»

неполными. Например, высказывание «некоторое событие произошло неделю назад» в очень редких случаях означает именно «семь дней назад»; чаще всего при таком ответе респондент подразумевает «семь дней плюс–минус два дня». Таким образом, интерпретация ответа зависит от выбранной респондентом единицы измерения, полученное значение следует рассматривать не как точку на временной оси, а как гранулу [3, 5], размер которой зависит от единицы измерения (рисунок 1).

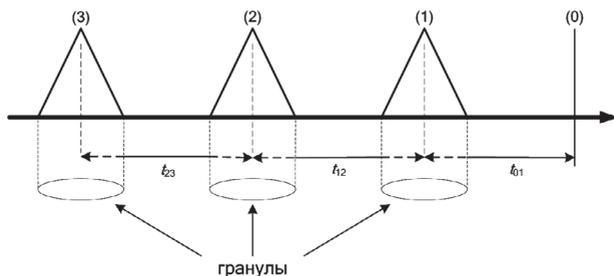


Рис. 1. Грануляция ответов

Отметим, что социально-значимое поведение рассматривается как случайный процесс определенного класса. Цель доклада — предложить подход к получению оценок интенсивности социально-значимого поведения, учитывающий особенности исходных данных, в частности, их гранулярный характер.

Описание модели

Для получения необходимых оценок интенсивности и производных характеристик социально-значимого поведения респондентов предлагается свести исходную задачу к построению байесовской сети доверия [6, 7], что позволит воспользоваться существующим алгоритмическим аппаратом теории байесовских сетей доверия. В этом случае появление большего числа наблюдений, скрытых переменных, необходимость отобразить более сложные связи сведется к частичной модификации построенной ранее модели: к появлению большего числа узлов, установлению новых связей, пересчету некоторых условных вероятностей. На рисунке 2 показана модель социально-значимого поведения, основанная на сведениях о последних эпизодах поведения (t_{ij}), а также минимальном (t_{\min}) и максимальном (t_{\max}) интервалах между последовательными эпизодами.

Таким образом, на рисунке 2 представлены случайные элементы, входящие в модель, и связи между ними, другими словами, связи между исходными данными и интересующими нас параметрами. Отметим, что в данной модели все элементы дискретны, то есть непрерывные величины, соответ-

ствующие интенсивности и длинам интервалов между эпизодами преобразовываются к дискретному виду.

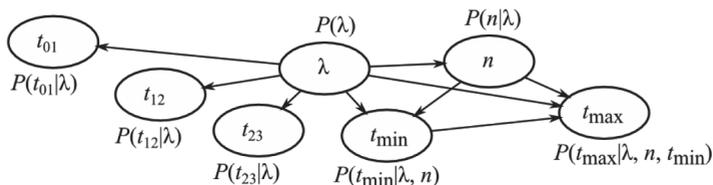


Рис. 2. Модель социально-значимого поведения

Численный пример

Пусть на вопрос о трех последних эпизодах получен ответ вида «последний эпизод произошел 4 дня назад, предпоследний — 9 дней назад, а третий с конца — четыре недели назад». Переведя такой ответ в длины интервалов (в днях), получим $T_{01}=4$, $T_{12}=5$, $T_{23}=19$.

В качестве примера рассмотрим дискретизацию следующего вида: для интенсивности $[0; 0,01)$, $[0,01; 0,05)$, $[0,05; 0,1)$, $[0,1; 1)$, $[1; 10)$, $[10; +\infty)$; для длин интервалов между эпизодами $[0; 0,1)$, $[0,1; 1)$, $[1; 7)$, $[7; 30)$, $[30; 180)$, $[180; +\infty)$.

Отметим, что $T_{01} \in [1; 7)$, $T_{12} \in [1; 7)$, $T_{23} \in [7; 30)$. После включения этих данных в рассматриваемую модель производится пересчет вероятностных распределений случайных элементов (заметим, что осуществление вероятностного вывода в данной модели выполняется программным обеспечением GeNIe&SMILE [8]). Апостериорное распределение интенсивности поведения, в рассматриваемом примере имеет следующий вид (таблица 1):

λ	$[0; 0,01)$	$[0,01; 0,05)$	$[0,05; 0,1)$	$[0,1; 1)$	$[1; 10)$	$[10; +\infty)$
$p(\lambda)$	0,004	0,267	0,728	<0,001	0	0

Таблица 1. Апостериорное распределение интенсивности

Полученное распределение позволяет делать выводы вида «интенсивность поведения меньше 0,1 с вероятностью больше 0,999» или эквивалентно «с вероятностью больше 0,999 эпизоды поведения происходят не чаще 1 раза в 10 дней».

Заключение

В работе предложен подход к построению моделей социально-значимого поведения в виде байесовской сети доверия на основе данных об эпизо-

дах поведения, рассматриваемого как пуассоновский случайный процесс. Кроме того, приведена процедура получения выводов об интенсивности поведения на основе численного примера.

Литература

1. Тулупьева Т.В., Пащенко А.Е., Тулупьев А.Л., Красносельских Т.В., Казакова О.С. Модели ВИЧ-социально-значимого поведения в контексте психологической защиты и других адаптивных стилей. СПб.: Наука, 2008. 140 с.
 2. Степанов Д.В., Мусина В.Ф., Суворова А.В., Тулупьев А.Л., Сироткин А.В., Тулупьева Т.В. Функция правдоподобия с гетерогенными аргументами в идентификации пуассоновской модели рискованного поведения в случае информационного дефицита // Труды СПИИРАН. 2012. Вып. 23. С. 157–184.
 3. Суворова А.В., Тулупьев А.Л., Пащенко А.Е., Тулупьева Т.В., Красносельских Т.В. Анализ гранулярных данных и знаний в задачах исследования социально значимых видов поведения // Компьютерные инструменты в образовании. №4. 2010. С. 30–38.
 4. Суворова А.В., Тулупьева Т.В., Тулупьев А.Л., Сироткин А.В., Пащенко А.Е. Вероятностные графические модели социально-значимого поведения индивида, учитывающие неполноту информации // Труды СПИИРАН. 2012. Вып. 22. С. 101–112.
 5. Заде Л.А. Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии информационных/интеллектуальных систем // Новости Искусственного Интеллекта. 2001. №2–3. С.7–11.
 6. *Perl J.* Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. NY etc.: Morgan Kaufmann Publ., 1994. P. 552.
 7. *Perl J.* Causality: Models, Reasoning, and Inference. Cambridge: Cambridge University Press, 2000. 400
 8. GeNIe& SMILE // Decisions systems laboratory. School of Information Sciences. University of Pittsburg. URL: <http://genie.sis.pitt.edu/> (доступ 05.05.2012).
-

РАНДОМИЗАЦИЯ НЕОПРЕДЕЛЕННОСТИ ИСХОДНЫХ ДАННЫХ ПРИ АНАЛИЗЕ УГРОЗООБРАЗУЮЩЕГО ПОВЕДЕНИЯ¹

Пащенко А.Е.

н. с. лаборатории ТuМПИ СПИИРАН, aep@iias.spb.su

Суворова А.В.

*м. н. с. лаборатории ТuМПИ СПИИРАН,
аспирант кафедры информатики СПбГУ, suvalv@mail.ru*

Тулупьев А.Л.

*зав. лаборатории ТuМПИ СПИИРАН,
профессор кафедры информатики СПбГУ, alt@iias.spb.su*

Тулупьева Т.В.

*с. н. с. лаборатории ТuМПИ СПИИРАН,
доцент кафедры информатики СПбГУ, tvt100a@mail.ru*

Аннотация: В докладе представлена процедура рандомизации неопределенности ответа, позволяющая обработать особенности естественно-языковых формулировок ответов о последних эпизодах угрозообразующего поведения.

Разработан программный комплекс, реализующий данную процедуру и позволяющий проводить вычислительные эксперименты с разными значениями параметров.

Введение

В связи с задачами своевременного обнаружения изменений в поведении отдельных индивидов и групп, науки социогуманитарного цикла испытывают потребность в математических моделях и алгоритмах, которые бы позволили получать оценки интенсивности угрозообразующего (то есть приводящего к возникновению угрозы) поведения. В качестве примера можно привести угрозообразующее поведение пользователя информационной системы, которое может привести к раскрытию критичной информации [1]. При этом существующие методы прямого измерения интенсивности (круглосуточный мониторинг, дневниковый метод, длительное сопровождение когорты индивидов и пр.) часто не применимы из-за их дороговизны, а также из-за ряда проблем этического характера.

Отметим, что наиболее доступными исходными данными для анализа поведения выступают самоотчеты респондентов об их поведении, то есть

¹ Доклад содержит материалы исследований, частично поддержанных грантами РФФИ 09-01-00861-а, 10-01-00640-а, 12-01-00945-а.

ответы в анкете на блок вопросов или результаты проведения интервью. На данный момент разработаны и применяются в опросах два подхода к оцениванию интенсивности поведения: прямые вопросы и Лайкерт-шкалы — каждый из которых имеет недостатки [2]. Одной из возможных альтернатив представляется опрос респондента о нескольких последних эпизодах его поведения (рисунок 1). Однако ограниченное число и неточность, фактически, нечеткость естественно-языковых формулировок ответов (например, «на прошлой неделе») требуют новых методов для обработки таких данных и получения количественной оценки интенсивности угрожающего поведения.

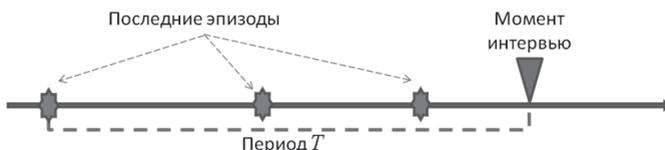


Рис. 1. Последние эпизоды поведения

В результате все более актуальной становится междисциплинарная фундаментальная научная проблема — развитие методологии поиска, представления, агрегирования и обработки данных и знаний (полученных из самоотчетов респондентов) в условиях информационного дефицита для последующего формирования и расчета косвенных оценок интенсивности социально-значимого поведения. Эта проблема требует развития моделей и алгоритмов в рамках специфических математических и компьютерных дисциплин: теории принятия решений, искусственного интеллекта, мягких вычислений, теории вероятностей и математической статистики.

Цель данного доклада — описать подход в обработке особенностей естественно-языковых формулировок ответов о последних эпизодах угрожающего поведения.

Рандомизация ответов

Ответы на вопросы об эпизодах поведения поступают на естественном языке, т. е. являются в значительной степени нечеткими и неполными. Отметим, что респонденты используют в своих высказываниях разные единицы измерения: часы, дни, недели, месяцы, полугодия, года. Причем использованная единица измерения несет в себе информацию о точности измерения. Поясним это на примере двух, на первый взгляд равнозначных, высказываний: «семь дней назад» и «неделю назад». Когда респондент использует формулировку «семь дней назад», это свидетельствует о его уверенности в том, что событие произошло именно семь дней назад. В то время как «неделю назад» — это может быть и пять, и восемь дней назад.

Для учета указанной неточности каждый ответ рассматривается не как точка на временной оси, а как интервал, длина которого зависит от единицы измерения (рисунок 2). Значение каждого ответа рассматривается, таким образом, не как константа, а как случайная величина с заранее заданным распределением [2]. Введенная случайная величина за счет рандомизации [3] неопределенности ответа, обусловленной нечеткостью его формулировки, позволяет рассмотреть интенсивность как случайную величину и вычислить характеристики последней.

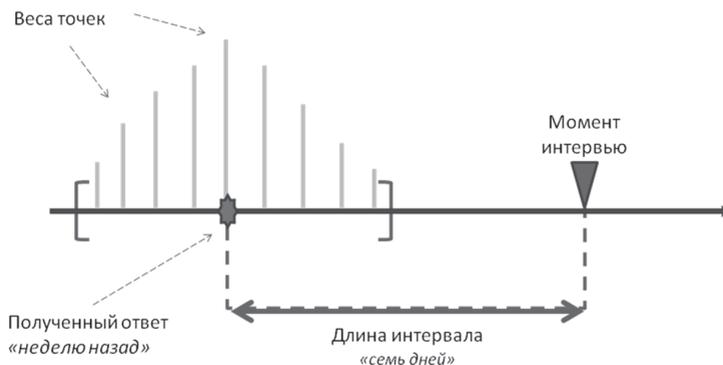


Рис. 2. Рандомизация ответа о последнем эпизоде

Поясним более подробно. Применяя идею метода анализа и синтеза показателей при информационном дефиците Н.В. Хованова [3], получим следующую процедуру обработки естественно-языковых ответов. Пусть известны данные о последних эпизодах поведения $t_1, t_2, t_3, \dots, t_N$ а T — общий временной промежуток, за который произошли эпизоды. Тогда интенсивность поведения λ оценивается по формуле: $\lambda = N/T$ [4–6].

Для каждого эпизода со значением t_i , $1 \leq i \leq N$ (где N — число рассматриваемых эпизодов поведения) через характеристику разброса δ определяется интервал (возможных значений) в днях: $[t_i - \delta x, t_i + \delta x]$, где x — коэффициент перевода рассматриваемой единицы измерения в дни [35]. Заметим, что любая точка из интервала $[t_i - \delta x, t_i + \delta x]$ возможна в качестве значения оценки t_i ; что, однако, не означает, что точки из этого интервала равновероятны в качестве такого. Сведения о такого рода отношениях между допустимыми значениями можно задать с помощью их распределения вероятностей [7]. В зависимости от предположений о характере ответов респондента для задания случайной величины t_i оценки t_i используется равномерное, биномиальное или какое-либо другое вероятностное распределение.

Введенная случайная величина t_i за счет рандомизации [3] неопределенности ответа позволяет рассмотреть интенсивность как случайную величину и вычислить характеристики последней.

Расчет среднего значения для случая трех последних эпизодов производится по следующей формуле:

$$\lambda_{\text{среднее}} = \sum_{i,j,k} (\lambda_{ijk} p_i p_j p_k),$$

где p_i — вес i -ой точки из первого интервала, p_j — вес j -ой точки из первого интервала, p_k — вес k -ой точки из первого интервала, λ_{ijk} — оценка интенсивности для соответствующего сочетания точек, т.е. $\lambda_{ijk} = N/\tau_{ijk}$, где τ_{ijk} — соответствующая точкам i, j, k оценка величины рассматриваемого интервала.

Среднее квадратичное отклонение σ для рассчитываемого среднего значения:

$$\sqrt{\sigma = \sum_{i,j,k} \lambda_{ijk}^2 p_i p_j p_k - \lambda_{\text{среднее}}^2}.$$

Программный комплекс

Процедуры расчета средних оценок были реализованы в программном комплексе [7], позволяющем задавать значения необходимых параметров рандомизации полученного ответа о последнем эпизоде угрожающего поведения. К таким параметрам относятся:

- характеристика разброса, определяющая оценку неопределенности ответа;
- число точек разбиения интервала, используемое при расчетах средней оценки интенсивности;
- вероятностное распределение, характеризующее рандомизацию — равномерное, треугольное, трапециевидное, биномиальное, бета-распределение, синусоидальное, полиномиальное, семиэллиптическое. Также имеется возможность указать параметр смещения для этих распределений.

Кроме того, можно указать алгоритм обработки неопределенности, применяющийся при расчете средней оценки интенсивности — либо на основе весов, либо на основе квантилей.

Заключение

Предложенная процедура рандомизации неопределенности ответа позволяет обработать особенности естественно-языковых формулировок ответов о последних эпизодах угрожающего поведения, связанные с тем, что ответ выражается в терминах «бытовой» речи, предполагающей определенную неточность ответа.

Литература

1. *Тулупьева Т.В., Тулупьев А.Л., Азаров А.А., Пащенко А.Е.* Психологическая защита как фактор уязвимости пользователя в контексте социоинженерных атак // Труды СПИИРАН. 2011. Вып. 18. С. 74–92.
2. *Суворова А.В., Тулупьев А.Л., Пащенко А.Е., Тулупьева Т.В., Красносельских Т.В.* Анализ гранулярных данных и знаний в задачах исследования социально значимых видов поведения // Компьютерные инструменты в образовании. №4. 2010. С. 30–38.
3. *Хованов Н.В.* Анализ и синтез показателей при информационном дефиците. СПб.: Изд-во СПбГУ, 1996. 196 с.
4. *Тулупьева Т.В., Пащенко А.Е., Тулупьев А.Л., Красносельских Т.В., Казакова О.С.* Модели ВИЧ-рискованного поведения в контексте психологической защиты и других адаптивных стилей. СПб.: Наука, 2008. 140 с.
5. *Пащенко А.Е., Тулупьев А.Л., Николенко С.И.* Моделирование заражения ВИЧ-инфекцией на основе данных о последних эпизодах рискованного поведения. // Известия высших учебных заведений: Приборостроение. 2006. № 8. 33–34 с.
6. *Тулупьева Т.В., Тулупьев А.Л., Пащенко А.Е.* Оценка интенсивности поведения респондента в условиях информационного дефицита // Труды СПИИРАН. Вып. 7. СПб.: Наука, 2008. С. 239–254.
7. *Пащенко А.Е., Суворова А.В.* Программный комплекс для экспертного оценивания интенсивности поведения респондента в условиях дефицита информации // Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте. Научно-практическая конференция студентов, аспирантов, молодых ученых и специалистов. Научные доклады. В 2-х т. Т. 2. М.: Физматлит, 2009. С. 220–241.

ПРОТОТИП ИНТЕРНЕТ-ПОРТАЛА РЕДАКЦИИ ЭЛЕКТРОННОГО НАУЧНОГО ИЗДАНИЯ

Абрамов М.В.

студент кафедры информатики СПбГУ, abramoff90@mail.ru

Ковалев В.С.

студент кафедры информатики СПбГУ

Аннотация: Интернет-портал редакции электронного научного издания представляет собой инструмент для автоматизации процессов редакции, публикации и рецензирования периодических научных изданий.

В докладе содержится информация о том, каких результатов удалось добиться, что уже сделано и в каком направлении планируется дальнейшее развитие портала.

Введение

В последнее время подписка на научные журналы становится все менее и менее популярна [1]. С развитием информационных технологий печатные издания стали пользоваться небольшим спросом, так как появилась возможность автоматизировать процесс работы со статьями в электронных научных изданиях.

Целью доклада являлось описание и демонстрация разработанного комплекса программ, автоматизирующего деятельность авторов, редакторов, редакции и контрагентов редакции.

Реализация поставленной цели предопределила необходимость решения следующих задач:

1. Создание и проектирование базы данных для хранения информации о всех зарегистрированных пользователях, хранимых ими статьях, публикациях и других файлах;
2. Разработка web-портала для взаимодействия пользователей и системы;
3. Реализация функции регистрации пользователей (ввод первичных данных, подтверждение регистрации через e-mail);
4. Реализация функций интерфейса автора (изменение личных данных, отправка публикаций, слежение за статусом своей публикации);
5. Реализация функций модератора (просмотр заявок на регистрацию, утверждение запросов пользователей на получение прав модератора, редактора или рецензента);
6. Реализация функций редактора (просмотр отправленных статей, утверждение или же отклонение их, добавление комментариев);
7. Реализация функции рецензента (написание рецензий для публикаций);
8. Визуальное оформление сайта.

По итогам работы был получен web-портал, на котором реализованы интерфейсы автора статьи, редактора, рецензента и модератора. Автор, авторизовавшись в личном кабинете, имеет возможность отслеживать статус уже отправленного проекта, автором или соавтором которого является, имеет доступ к чтению замечаний к проекту, обладает интерфейсом для ответа на замечания и возможностью редактирования ещё не отправленного проекта, при условии, что он является ответственным лицом. Редактор имеет возможность править загруженные статьи, создавать замечания по уже отредактированным статьям и отвечать на вопросы авторов по замечаниям к их проектам. Интерфейс рецензента предполагает доступ к отправленным авторами проектам и возможность создавать рецензии на данные проекты. Интерфейс модератора необходим для поддержки работоспособности сайта, подтверждения статуса регистрируемых пользователей.

Роль автора

Первоначально все пользователи делятся на тех, кто еще не зарегистрирован и на тех, кто уже прошел процесс регистрации.

Любой пользователь видит главную страницу, может просматривать содержимое страниц «О сайте», «Техническая поддержка», «Вакансии» и «Контакты».

Всех зарегистрированных пользователей можно разделить на четыре группы: авторы, модераторы, редакторы и рецензенты. Необходимый функционал нужно выбрать в процессе регистрации.

Автор — единственный функционал, который не требует подтверждения модератора. Авторы могут отправлять публикации, заполняя информацию о себе, а также следить за статусом своих отправленных публикаций. Более того имеется возможность читать комментарии редактора и отвечать на них.

На рисунке 1 изображена главная страница личного кабинета автора, на рисунке 2 — меню такого личного кабинета.

Сайт Главная Университетская страница Контакты Биокарт

Электронная редакция СВВИАРАН

Приветствуем Вас, Ковалев Владимир Сергеевич, на сайте svvian.ru

Ваши персональные данные:

Фамилия:

Имя:

Отчество:

Научные звание и степень:

Должность:

Наименование подразделения СВВИАРАН:

Адрес электронной почты:

Прямой городской телефон:

Мобильный телефон:

Местный телефон:

Рис. 1. Главная страница личного кабинета автора

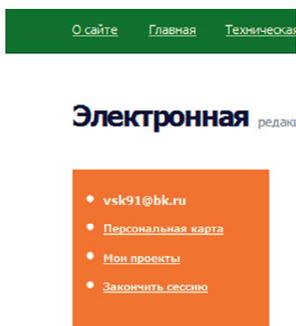


Рис. 2. Меню личного кабинета автора

Роль модератора

Модератор сайта нужен для того, чтобы следить за всеми пользователями сайта, подтверждать заявку на регистрацию редакторов, рецензентов или других модераторов. После успешной регистрации и получения роли модератора в меню его личного кабинета появляется пункт «Заявки на регистрацию». В этом разделе модератор увидит регистрационные данные потенциальных пользователей, которые желают выступать на портале в роли модераторов, редакторов или рецензентов. Модератор может принять или отклонить любую заявку.

На рисунке 3 показана главная страница личного кабинета пользователя с правами модератора, на рисунке 4 — меню такого личного кабинета.

Номер	ФИО	Степень	Должность	Подразделение СПИРАН	E-mail	Городской тел.	Моб.	Местный тел.	Подтверждение
Пользователи, которые подали заявки на то, чтобы стать модераторами:									
5	Пупкин Василий Иванович	нет	крутая	нет	po-cold@yandex.ru	666-66-66	777-77-77	888-88-88	Подтвердить Отклонить
6	Михайлов Владимир Сергеевич	нет	есть	нет	abramoff90@yandex.ru	555-55-55	111-11-11	333-33-33	Подтвердить Отклонить
Пользователи, которые подали заявки на то, чтобы стать редакторами:									
5	Пупкин Василий Иванович	нет	крутая	нет	po-cold@yandex.ru	666-66-66	777-77-77	888-88-88	Подтвердить Отклонить
6	Михайлов Владимир Сергеевич	нет	есть	нет	abramoff90@yandex.ru	555-55-55	111-11-11	333-33-33	Подтвердить Отклонить
Пользователи, которые подали заявки на то, чтобы стать рецензентами:									
5	Пупкин Василий Иванович	нет	крутая	нет	po-cold@yandex.ru	666-66-66	777-77-77	888-88-88	Подтвердить Отклонить

Рис. 3. Главная страница личного кабинета модератора

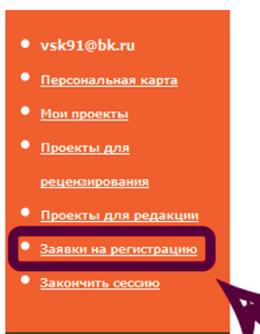


Рис. 4. Меню личного кабинета модератора

Роль редактора

Основные обязанности редактора связаны с решением дальнейшей судьбы статьи, направленной автором. Он отслеживает поступление в редакцию новых проектов, просматривает собранные автором материалы и, в случае полного соответствия установленным требованиям, принимает её, одновременно отправляя рецензенту при необходимости. В противном случае редактор может отклонить статью, либо же принять с дополнениями, в этом случае он оставляет свои замечания, на которые автор должен соответствующим образом отреагировать.

Номер	Авторы	Емэй	Название	Время загрузки/обновления	Статус
1	Абрамов Максим Викторович	abr@mail90@mail.ru	Протокол интернет-портала редакции электронного научного издания: автоматизация ролей редактора и рецензента Internet portal prototype for scientific e-journal release edition: editor and reviewer role automation	2013-05-11 18:35:18/2013-05-11 20:14:03	Готова к редакции
2	Абрамов Максим Викторович	abr@mail90@mail.ru	Протокол интернет-портала редакции электронного научного издания: автоматизация ролей редактора и рецензента Internet portal prototype for scientific e-journal release edition: editor and reviewer role automation	2013-05-11 19:08:27/2013-05-11 20:15:03	Готова к редакции

Рис. 5. Главная страница личного кабинета редактора

Роль рецензента

Если потенциальный пользователь регистрируется как рецензент и успешно проходит проверку модератором, то в его меню слева появляется пункт «Проекты для рецензирования». Здесь расположены направленные

данному рецензенту редакцией статьи с целью получения на них отзыва. Написанная рецензия отправляется обратно в редакцию, где формируется выпуск из полностью готовых к публикации статей. На рисунке 6 показан личный кабинет пользователя, который выполняет на портале все заявленные роли: автора, модератора, редактора и рецензента.

Необходимо поставить оценку по пятибалльной шкале.

Степень знакомства автора с современным состоянием исследований:	1
Соответствие содержания статьи тематике журнала:	1
Новизна и оригинальность результатов:	1
Значимость вклада, вносимого в развитие теории и практики:	1
Соответствие названия статьи ее содержанию:	1
Адекватность аннотации и ключевых слов, достаточность списка цитируемой литературы:	1
Логичность представления материала, ясность языка и стиль изложения:	1
Оформление и техническое представление текста:	1
Адекватность и обоснованность выводов:	1
Рекомендации (при выборе пунктов 1-4 обязательно прокомментируйте в поле ниже):	Принять без доработки (5)

Комментарии к рекомендациям 1-4. Иные комментарии, которые требуется довести до сведения автора статьи.

Комментарии (если требуются), адресованные редакции, которые не доводятся до сведения автора.

Отклоняются статьи:

- не носящие научного или образовательного характера
- носящие антинаучный характер
- прочтение которых затруднено или невозможно из-за ошибок в тексте и/или несогласованности его элементов
- публикации которых может привести к нарушению действующего законодательства Российской Федерации или авторских и/или смежных прав
- содержащие утверждения, суждения или заключения, не подтверждаемые ни собственными исследованиями авторов, ни ссылками на внешние источники
- подготовленные с нарушением норм научной этики

Загрузить рецензию

Рис. 6. Личный кабинет рецензента

Заключение

В результате проведенной работы был создан прототип интернет-портала редакции электронного научного издания и разработаны пользовательские интерфейсы редактора и рецензента.

Воспользовавшись разработанной системой, пользователь сможет зарегистрироваться на портале, автоматически получить права автора и сразу же начать работу со своими статьями. Те пользователи, которые запросили права редактора и получили подтверждение от модератора, смогут просматривать отправленные авторами проекты, редактировать их, принимать или отклонять. Также есть возможность писать комментарии к проектам и принимать их с дополнениями. После принятия статьи редактор может выбрать рецензента для написания рецензии к данному проекту.

Пользователи, получившие права рецензента, после авторизации в системе, через панель управления смогут обозревать проекты, отправленные редакторами для написания рецензии. Также рецензенты могут скачивать все приложенные автором файлы. Для загрузки рецензии на странице управления проектами для рецензирования есть специальная кнопка «Загрузить».

Модераторы сайта, после авторизации в личном кабинете, смогут подтверждать или отклонять пользовательские заявки на привилегии редактора, рецензента или модератора.

Литература

1. Современные проблемы науки и образования, [Электронный ресурс] / science-education, // URL: <http://www.science-education.ru/prospects/> (по состоянию на 15.03.2013).
-

МЕТОДЫ ПОСТРОЕНИЯ И ОБРАБОТКИ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ, ПРЕДСТАВЛЕННЫХ В ВИДЕ АЛГЕБРАИЧЕСКИХ БАЙЕСОВСКИХ СЕТЕЙ¹

Алексеев А.М.

студент кафедры информатики СПбГУ, anton.m.alexeyev@math.spbu.ru

Аннотация: Вероятностные графические модели (ВГМ) — один из способов представления знаний с неопределённостью. Они успешно применяются для решения задач машинного обучения, в том числе благодаря тому, что есть развитые методы обработки этих моделей. Все ВГМ задают вероятностные распределения; часто удобно представлять одни ВГМ в виде других.

В докладе будут представлены методы декодирования скрытых состояний и обучения бинарной скрытой марковской модели, представленной в виде алгебраической байесовской сети.

Введение

Вероятностные графические модели (ВГМ) [2] задают распределения (семейства распределений) случайных величин с разреженными вероятностными зависимостями. К их числу принадлежат скрытые марковские модели (СММ), байесовские сети доверия, условные случайные поля, марковские сети и др. Ввиду близкой вероятностной природы некоторые модели можно представлять друг в виде друга. Известны примеры представления СММ в виде байесовских сетей доверия [1], классов байесовских сетей доверия в виде алгебраических байесовских сетей (АБС; логико-вероятностные графические модели) [11], СММ в виде АБС [6–9]. Так, обучив одну модель, можно обрабатывать полученные знания методами другой.

На данный момент для бинарных скрытых марковских моделей в виде алгебраических байесовских сетей (СММ-АБС) существуют:

1. Алгоритм построения по скрытой марковской модели.
2. Алгоритм решения задачи оценки правдоподобия последовательности наблюдений (методами логико-вероятностного вывода в АБС).
3. Алгоритм декодирования последовательности скрытых состояний по последовательности наблюдений.
4. Метод обучения по корпусу вида $\{(s_i, o_i)\}_{i=0}^T$, где s_i — скрытое состояние в момент i , а o_i — соответствующее наблюдение.

Цель предлагаемого доклада — изложить методы 3 и 4.

¹ Работа выполнена при финансовой поддержке РФФИ, гранты № 12-01-00945-а, 12-01-31202-мол_а.

Скрытые марковские модели

Скрытая марковская модель — темпоральная вероятностная графическая модель. Задан конечный набор состояний, текущее состояние в данный момент времени «порождает» наблюдение из конечного множества наблюдений. Предполагается, что последовательность состояний удовлетворяет марковскому свойству, а вероятность «порождения» наблюдения зависит только от текущего состояния.

Таким образом, скрытую марковскую модель можно задать как следующую тройку:

- вектор вероятностей для первого состояния,
- условные вероятности «переходов» из состояния в состояние,
- вероятности «порождения» наблюдения состоянием.

В теории скрытых марковских моделей выделяют три ключевых задачи [4]: оценка правдоподобия данной последовательности наблюдений при данной модели (вычисление вероятности «порождения» наблюдений моделью), декодирование скрытых состояний по данной цепочке наблюдений (поиск наиболее вероятной последовательности состояний, которая могла «породить» данную последовательность наблюдений), обучение данной модели (настройка параметров модели таким образом, чтобы вероятность данной последовательности наблюдений была максимальна).

Алгебраические байесовские сети

В основе теории АБС лежит вероятностная логика Н. Нильссона [3]. Для задания вероятностных оценок истинности всех формул над n высказываниями потребуется задать вероятностные оценки 2^{n-1} конъюнктов (элементарных конъюнкций без отрицаний). АБС — набор моделей фрагментов знаний (то есть таких наборов конъюнктов с оценками) над некоторыми подмножествами множества заданных высказываний. Наиболее развитый на данный момент подход к обработке знаний в модели — представление АБС как графа, вершины которого соответствуют фрагментам знаний. Существуют методы осуществления априорного вывода (вычисления оценок вероятностей формул над высказываниями из АБС) и апостериорного (байесовского) вывода (распространение свидетельств, то есть изменений оценок вероятностей, по сети). С перечисленным можно ознакомиться в работах [10–16].

Декодирование скрытой последовательности

Рассказ о второй задаче скрытых марковских моделей редко обходится без упоминания алгоритма Витерби [5]. Предлагаемый алгоритм также основан на методе динамического программирования.

Постановка задачи: по последовательности наблюдений восстановить наиболее вероятную последовательность скрытых состояний.

Обозначения:

- Propagate(abn, notation) — функция, пропагирующая детерминированное свидетельство notation;
- Probability(abn, notation) — функция, возвращающая вероятностную оценку истинности формулы, заданной notation, в сети abn;
- w — двумерный массив, в котором будут храниться величины, которые нам понадобятся в процессе вычислений;
- path — коллекция-словарь наиболее вероятных «префиксных траекторий» модели;
- buffer — вспомогательная коллекция-словарь.

Алгоритм:

Require abn, O

```

1: for  $i \rightarrow 0 \dots T$  do
2:   Propagate(abn, ' $O_i = V_i$ ')
3: end for
4: for  $i \leftarrow 0 \dots n - 1$ 
5:    $w[0, j] \leftarrow \text{Probability}(\text{abn}, 'X_0 = s_j')$ 
6: end for
7: for  $t \leftarrow 1 \dots T$ 
8:   buffer  $\leftarrow \emptyset$ 
9:   for  $j \leftarrow 0 \dots n - 1$ 
10:     $k_{\text{opt}} \leftarrow \text{argmax}_k \left\{ \frac{w[t-1, k] \text{Probability}(\text{abn}, 'X_t = s_j \wedge X_{t-1} = s_k')}{\text{Probability}(\text{abn}, 'X_{t-1} = s_k')} \right\}$ 
11:    buffer[  $j$  ]  $\leftarrow \text{path}[k_{\text{opt}}] : k_{\text{opt}}$ 
12:     $w[t, j] \leftarrow \left\{ \frac{w[t-1, k_{\text{opt}}] \text{Probability}(\text{abn}, 'X_t = s_j \wedge X_{t-1} = s_{k_{\text{opt}}}')}{\text{Probability}(\text{abn}, 'X_{t-1} = s_{k_{\text{opt}}}')} \right\}$ 
13:   end for
14:   path  $\leftarrow$  buffer
15: end for
16:    $k_{\text{last}} \leftarrow \text{argmax}_k \{ w[T, k] \}$ 
17: return path[  $k_{\text{last}}$  ] :  $k_{\text{last}}$ 

```

Шаги (1–3) — распространение свидетельств $\langle O_t = v_t \rangle, t = 0 \dots T$.

Шаги (4–6) — записываем в нулевую строку матрицы w апостериорные вероятности $p_a(X_0 = s_j)$ того, что на первом шаге мы о соответствующем состоянии.

В цикле (7–15) для каждого шага t и состояния j находим состояние k_{opt} предыдущего шага, на котором максимальна величина

$$\frac{w_{t-1,k} p_a(X_t = s_j \wedge X_{t-1} = s_k)}{p_a(X_{t-1} = s_k)},$$

записываем её в матрицу, а траекторию $\text{path}[j]$ процесса, оказавшегося в момент t в состоянии j , задаём как $\text{path}[k_{\text{opt}}]:k_{\text{opt}}:j$.

На шаге 14 находим состояние k_{last} , максимизирующее величину $w[T, k]$ (наиболее вероятная траектория заканчивается в состоянии k_{last}).

В качестве наиболее вероятной последовательности скрытых состояний возвращаем траекторию, которую мы нашли для k_{last} .

Обучение СММ-АБС

Постановка задачи:

- $S = \{s_0, s_1\}$ и $V = \{v_0, v_1\}$ — множества состояний модели и наблюдений соответственно;
- $B = \{(x_t, o_t)\}_{t=0}^T$ — корпус данных для обучения, то есть набор «иных» пар «состояние – наблюдение» во времени.

Требуется построить СММ-АБС, эквивалентную СММ, обученной по этим данным.

Как и в случае СММ, будем считать, что условные вероятности «перехода» из состояния в состояние и «порождения» данным состоянием данного наблюдения постоянны.

Метод: будем считать, что оценки для $p(o|x)$ и $p(x|y) \forall x, y \in S \forall o \in O$, где S — множество состояний, а O — множество наблюдений, нами уже получены (посредством P_{xx}, P_{ox}, P_x ; см. ниже).

Обозначения:

- abn — переменная «АБС»,
- $p(\text{abn}, \text{notation})$ — «свойство», дающее доступ к вероятности истинности формулы notation в abn ,
- $p(\text{abn}, \text{notation}, \text{bool} \dots)$ — «свойство», дающее доступ к вероятности истинности формулы notation с означиваниями высказываний, заданными списком bool ,
- $\text{glue}(\text{abn}, \text{notation})$ — метод, синтезирующий фрагмент знаний с максимальным конъюнктом, заданным notation ,
- P_{xx}, P_{ox}, P_x — массивы с соответствующими заданными заранее оценками вероятностей,
- s_0, s_1, v_0, v_1 — константы, причём v_0, s_0 — ложь, v_1, s_1 — истина,
- $T+1$ — число шагов.

Алгоритм:

Require: $T, P_x[2], P_{xx}[2, 2], P_{ox}[2, 2]$

Ensure: $\sum_{x \in \{0,1\}} P_x[x] = \sum_{x,y \in \{0,1\}} P_{xx}[x,y] = \sum_{x,y \in \{0,1\}} P_{ox}[x,y] = 1, T > 0$

- 1: $s_0 = 0, s_1 = 1$
- 2: $v_0 = 0, v_1 = 1$
- 3: $abn \leftarrow \emptyset$
- 4: $p(abn, 'x_0') \leftarrow P_x[s_1]$
- 5: $p(abn, 'o_0x_0') \leftarrow P_{ox}[v_1, s_1]$
- 6: $p(abn, 'o_0') \leftarrow P_{ox}[s_1, s_1] + P_{ox}[s_1, s_0]$
- 7: $glue(abn, 'o_0x_0')$
- 8: $p(abn, 'x_0x_1') \leftarrow P_{xx}[s_1, s_1]$
- 9: for $t \leftarrow 1 \dots T$ do
- 10: $p(abn, 'x_t') \leftarrow \sum_{s \in \{s_0, s_1\}} p(abn, 'x_{t-1}', s) \frac{P_{xx}[s, s_1]}{P_x[s]}$
- 11: $glue(abn, 'x_{t-1}x_t')$
- 12: $p(abn, 'o_t') \leftarrow \sum_{s \in \{s_0, s_1\}} p(abn, 'x_t', s) \frac{P_{ox}[v_1, s]}{P_x[s]}$
- 13: $p(abn, 'o_t') \leftarrow \sum_{s \in \{s_0, s_1\}} p(abn, 'x_t', s) \frac{P_{ox}[v_1, s]}{P_x[s]}$
- 14: $glue(abn, 'o_tx_t')$
- 15: if $t < T$ then
- 16: $p(abn, 'x_tx_{t+1}') \leftarrow p(abn, 'x_t') \frac{P_{xx}[s_1, s_1]}{P_x[s_1]}$
- 17: end if
- 18: end for

return abn

Комментарий:

- в строках 1 и 2 задаются константы;
- в строках 3–8 АБС abn инициализируется известными нам оценками, которые задают фрагмент знаний над алфавитом $\{x_0, o_0\}$; также задается вероятность конъюнкта x_0x_1 ;
- в строках 9–18 производится оценивание новых конъюнктов.

В качестве примера оценок вероятностей P_{xx} , P_{ox} , P_x будут приведены относительные частоты соответствующих паттернов в корпусе.

Заключение

В докладе изложены методы декодирования скрытых состояний и обучения для бинарных СММ в виде АБС. Они демонстрируют выразительную силу алгебраических байесовских сетей и позволяют говорить о будущей «интервализации» скрытых марковских моделей.

Литература

1. *Cowell R.G., Dawid A.P., Lauritzen S.L., Spiegelhalter D.J.* Probabilistic Networks and Expert Systems. NY.: Springer-Verlag, 1997. 370 p.
2. *Koller D., Friedman N.* Probabilistic Graphical Models. Principles and Techniques. Cambridge, Massachusetts, London: MIT Press, 2009. 1231 pp.
3. *Nilsson N.J.* Probabilistic Logic // Artificial Intelligence. 1986. Vol. 28. P. 71–87.
4. *Rabiner L.* A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // IEEE Proceedings. 1989. Vol. 77. No. 2. Pp. 257–286.
5. *Viterbi A.J.* Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. // Information Theory, IEEE Transactions on. 1967. Vol. 13. No. 2. Pp. 260–269.
6. *Момзикова М.П., Великодная О.И., Пинский М.Я., Сироткин А.В., Тулупьев А.Л., Фильченков А.А.* Оценка вероятности наблюдаемой последовательности в бинарных линейных по структуре скрытых марковских моделях с помощью апостериорного вывода в алгебраических байесовских сетях // Труды СПИИРАН. СПб: Наука, 2010. Вып. 2. С. 122–142.
7. *Момзикова М.П., Великодная О.И., Пинский М.Я., Сироткин А.В., Тулупьев А.Л., Фильченков А.А.* Представление бинарных линейных по структуре скрытых марковских моделей в виде алгебраических байесовских сетей // Труды СПИИРАН. СПб: Наука, 2010. Вып. 1. С. 134–150.
8. *Ревзин Л.М.* Комплекс программ для представления и обработки ациклических скрытых марковских моделей на основе алгебраических байесовских сетей: Дипломная работа. Математико-механический факультет СПбГУ. 2012. 39 с. (Санкт-Петербургский государственный университет).
9. *Ревзин Л.М., Фильченков А.А., Тулупьев А.Л.* Представление многозначных линейных по структуре скрытых марковских моделей в виде алгебраических байесовских сетей // Труды СПИИРАН. 2012. Вып. 20. С. 186–199.
10. *Тулупьев А.Л., Николенко С.И., Сироткин А.В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
11. *Тулупьев А.Л.* Преобразование ациклических байесовских сетей доверия в алгебраические байесовские сети. // Известия высших учебных заведений: Приборостроение. 2009. Вып. 3, С. 21–23.
12. *Тулупьев А.Л., Сироткин А.В.* Алгебраические байесовские сети: принцип декомпозиции и логико-вероятностный вывод в условиях неопределенности // Информационно-измерительные и управляющие системы. 2008. № 10, Т. 6. С. 85–87.
13. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петербург. ун-та, 2009, 400 с.
14. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. Вып. 11. С. 104–129.
15. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Особенности анализа вторичной структуры алгебраической байесовской сети // Труды СПИИРАН. 2010. Вып. 1 (12). С. 97–118.
16. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Структурный анализ клик минимальных графов смежности // Вестн. Тверск. гос. ун-та. Сер.: Прикладная математика. 2011. № 20. С. 139–151.

ЗАДАЧА ВЫБОРА РАСПРЕДЕЛЕНИЯ, ОТРАЖАЮЩЕГО ВЕРОЯТНОСТНУЮ СЕМАНТИКУ АЛГЕБРАИЧЕСКОЙ БАЙЕСОВСКОЙ СЕТИ¹

Фроленков К.В.

аспирант кафедры информатики СПбГУ, м. н. с. СПИИРАН, frolenk@mail.ru

Аннотация: В работе рассматриваются различные подходы к конструированию распределения, задаваемого алгебраической байесовской сетью. Для сети, представленной ациклическим графом, показаны подходы, опирающиеся на понятие объемлющего фрагмента знаний для подсети и предположения об условной независимости. Поскольку последний подход не применим для циклической сети, предложен метод выбора распределения с максимальной энтропией и алгоритм для его конструирования.

Введение

Настоящий доклад посвящен одной из задач теории алгебраических байесовских сетей (АБС). АБС может быть использована для моделирования вероятностных взаимосвязей в сложных системах [4]. Такие системы представляются как наборы зависимых булевых случайных элементов, разбитые на поднаборы, образующие фрагменты знаний. АБС формализованы с помощью вероятностной логики [7], что позволяет говорить об оценках вероятностей пропозициональных формул, составленных на основе случайных элементов сети и о вероятностной семантике сети, то есть наборе совместных распределений данных элементов, задаваемом АБС.

Алгоритмы логико-вероятностного вывода [1], предоставленные инструментарием АБС, позволяют модифицировать параметры модели для её применения в конкретной ситуации, что осуществляется посредством передачи свидетельства в сеть. Для того чтобы ответить на вопрос, каким образом поступающее свидетельство меняет вероятностную семантику сети, необходимо предоставить метод её определения для случаев циклической и ациклической сети.

Описание модели

Фрагментом знаний (ФЗ) будем называть пару (C, p) , где C — множество всех возможных положительных элементарных конъюнкций некоторого набора пропозициональных переменных (данный набор будем называть

¹ Работа выполнена при финансовой поддержке РФФИ, гранты № 12-01-00945-а, 12-01-31202-мол_а.

нагрузкой ФЗ и обозначать $W((C, p))$, и $p: C \rightarrow I(R)$ (через $I(R)$ обозначим множество интервалов $[a; b]: 0 \leq a \leq b \leq 1$), отображение, задающее нижние и верхние оценки вероятности для каждой конъюнкции из C . p может быть представлено парой функций p^- и p^+ , удовлетворяющими $p(c) = [p^+(c); p^-(c)]$ для всех конъюнкций c из C .

Частным случаем для такого определения является фрагмент знаний, у которого нижние и верхние границы $p(c)$ совпадают. Такие ФЗ будем называть ФЗ со скалярными оценками и обозначать (C, p) .

АБС определяется набором ФЗ $\{(C_i, p_i)\}_{i=1}^n$. В данной работе рассматриваются только сети, включающие ФЗ со скалярными оценками и обладающие свойством непротиворечивости, то есть такие, что существует распределение $p: \forall c \in c_i \ p_i(c) = p(c)$.

Подходы к выбору распределения

Случай ациклической сети

Для набора ФЗ $\{(C_i, p_i)\}_{i=1}^n$, задающего непротиворечивую сеть можно построить [2] объемлющий ФЗ (C, p) со следующими свойствами: $W((C, p)) = \bigcup_i W(C_i, p_i)$, $\forall i \forall c \in c_i \ p(c) = p_i(c)$. Такой ФЗ обладает интервальными оценками вероятности и задаёт семейство всех возможных распределений, задаваемых сетями, содержащими заданный набор ФЗ.

Поскольку в теории АБС разработаны методы обработки интервальных вероятностей [3], возможно не разрешать неопределенность, присущую АБС, и определять её вероятностную семантику как набор всех возможных распределений, построенных с помощью алгоритма построения объемлющего ФЗ. Недостатком такого подхода является сложность операций по обработке интервальных вероятностей по сравнению с их эквивалентами для скалярного представления.

Другим подходом к определению семантики АБС является типичное [6] для вероятностных графических моделей использование предположений об условной независимости. Наборы таких предположений представляются с помощью графа. Данный граф в теории АБС называется графом смежности [5] и отражает структуру пересечений ФЗ. В вершинах такого графа располагаются ФЗ, а каждому из ребер сопоставлен сепаратор — множество переменных, содержащихся в ФЗ, инцидентных данному ребру. Для АБС набор таких предположений определяется правилом: если для двух случайных элементов a и b , не лежащих в одном ФЗ, и набора сепараторов S верно $a \perp b | S$, если любой путь из ФЗ, содержащего a , до ФЗ, содержащего b , проходит по ребру, соответствующему одному из сепараторов из S . В случае ациклической сети построенный таким образом набор предположений позволяет полностью задать совместное распределение для всех переменных, входящих в сеть.

Случай циклической сети

Описанный выше подход не может быть применен без изменений к сети, содержащей циклы. Для иллюстрации приведем пример АБС, состоящей из трех ФЗ с нагрузками $\{a, b\}$, $\{a, c\}$ и $\{b, c\}$. Граф смежности такой сети является полным и не определяет ни одного соотношения условной независимости, таким образом, неопределенность значения вероятности формула $a \& b \& c$ не может быть разрешена. Следует отметить, что циклические АБС могут задавать предположения независимости, но их недостаточно для обоснования выбора единственного распределения.

Несмотря на приведенные выше рассуждения, данный подход всё же может быть использован и для циклических сетей. Так как среди всех возможных распределений то, которое построено с применением набора предположений об условной независимости, в случае ациклической сети, обладает максимальной энтропией, будем искомым распределение по такому же принципу.

Алгоритм поиска распределения

Подход, основанный на максимизации энтропии, позволяет выбирать единственное распределение, определяемое сетью, но требует решения оптимизационных задач с большим количеством переменных (экспоненциальным от размера цикла). Для его поиска для циклических сетей с большим числом ФЗ предложим следующий метод. Для каждого цикла:

1. Из набора ФЗ $\{P_i\}$, содержащихся в цикле, случайным образом выберем один.
2. Так как сеть, состоящая из всех остальных ФЗ данного набора, является ациклической, применим алгоритм выбора распределения на основе предположений об условной независимости, ФЗ — результат применения такого алгоритма — обозначим через P .
3. До тех пор, пока данное действие изменяет $P = (C, p)$, передадим в P один из фрагментов P_i знаний в качестве свидетельства.

Данный алгоритм является адаптацией алгоритма IPFP [8], в котором фрагмент знаний АБС рассматривается как набор ограничений на совместное распределение.

Для проверки правильности работы описанного алгоритма была реализована программа на языке java. Для всех автоматически сгенерированных примеров циклов АБС верны свойства:

1. Алгоритм завершает работу, т.е. конструируется результирующий ФЗ P , который не изменяется при подаче в него произвольного ФЗ из $\{P_i\}$ в качестве свидетельства, из чего следует $\forall i \forall c \in C_i \ p(c) = p_i(c)$.
2. Результат не зависит от выбора фрагмента знаний на шаге 1.
3. Результат выполняет ограничения независимости, накладываемые сетью.

4. Результат совпадает с ФЗ, полученным непосредственным решением оптимизационных задач.

Заключение

В докладе были рассмотрены различные методы определения вероятностной семантики как ациклической, так и циклической АБС, продемонстрирована неприменимость подхода, основанного на представленном графом набора предположений об условной независимости переменных сети в случае наличия в АБС обособленного цикла. Данный метод обобщен на такие сети путём использования принципа максимизации энтропии. Предложен алгоритм выбора распределения, задаваемого сетью, сформулированы его свойства.

Литература

1. *Тулупьев А.Л.* Алгебраические байесовские сети: глобальный логико-вероятностный вывод в деревьях смежности. СПб.: СПбГУ; ООО Издательство «Анатолия», 2007. 40 с. (Элементы мягких вычислений).
2. *Тулупьев А.Л., Николенко С.И., Сироткин А.В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
3. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петербур. ун-та. 2009. 400 с.
4. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Синтез согласованных оценок истинности утверждений в интеллектуальных информационных системах // Известия высших учебных заведений: Приборостроение. 2006. № 7. 20–26 с.
5. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. № 11. С. 104–129.
6. *Koller D., Friedman N.* Probabilistic Graphical Models: Principles and Techniques. MA:MIT Press. 2009. 1231 p.
7. *Nilson N.J.* Probabilistic logic // Artificial Intelligence. N. 28(1). 1986. P. 71–87.
8. *Vomlel. J.* Methods of Probabilistic Knowledge Integration. PhD Thesis. Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University. December 1999.

СИНТЕЗ ВТОРИЧНОЙ СТРУКТУРЫ АЛГЕБРАИЧЕСКОЙ БАЙЕСОВСКОЙ СЕТИ: КОМПРОМИСС МЕЖДУ СЛОЖНОСТЬЮ И АЦИКЛИЧНОСТЬЮ¹

Фильченков А.А.

аспирант СПбГУ, м.н.с. лаб. ТуМПИ СПИИПАН aaafil@mail.ru

Аннотация: Ацикличность вторичной структуры алгебраической байесовской сети является необходимым требованием для осуществления алгоритмов логико-вероятностного вывода, однако при достижении возможности построить такую вторичную структуру возрастает и сложность осуществления алгоритмов.

В работе предлагается компромисс, который может быть достигнут между требованием ацикличности и ростом сложности алгоритмов логико-вероятностного вывода.

Введение

Алгебраическая байесовская сеть (АБС) — это логико-вероятностная графическая модель баз фрагментов знаний с неопределенностью [6, 7]. Другими словами, алгебраическая байесовская сеть, как и любая другая вероятностная графическая модель, декомпозирует многомерный случайный элемент [13] на совокупность случайных элементов меньшей размерности, причем задаваемые ими распределения вероятности за счет предположений условной независимости позволяют выразить случайное распределение исходного многомерного случайного элемента [8, 16]. Отличие АБС от других вероятностных графических моделей состоит в том, что они позволяют работать также и с интервальными оценками вероятности. В этом случае приходится говорить уже не просто о случайных элементах, а о семействах случайных элементов, поскольку интервальные оценки соотносятся с семейством вероятностных распределений.

Важную роль играет «топология» сети, т. е. граф, связывающий случайные элементы и отображающий условные независимости. В теории АБС такой граф называется ее *вторичной структурой*. Для поддержания целого ряда алгоритмов такой сети ее вторичная структура должна быть ациклична [4, 5]. Однако ацикличную вторичную структуру можно построить не над любой совокупностью случайных элементов, которые составляют *первичную структуру* АБС — такие первичные структуры называются *цикличными* [10]. Известны алгоритмы, которые позволяют преобразовывать цикличную первичную структуру АБС к ацикличной, однако их применение приводит

¹ Работа выполнена при финансовой поддержке РФФИ, гранты № 12-01-00945-а, 12-01-31202-мол_а.

к росту размерности случайных элементов. От размерности случайного элемента (в случае интервальных оценок размерности всех случайных элементов из семейства совпадают) экспоненциально зависит сложность работы в том числе и тех алгоритмов, ради которых требуется ацикличность вторичной структуры [3] — алгоритмов глобального логико-вероятностного вывода. Соответственно, требования ацикличности вторичной структуры и ограничения сложности работы алгоритмов логико-вероятностного вывода не могут быть удовлетворены.

В работе будет предложен компромисс, между двумя этими требованиями.

Первичная и вторичная структуры алгебраической байесовской сети

Вместо того, чтобы рассматривать случайные элементы (или семейства таких элементов), достаточно будет рассуждать лишь (упрощенно) о наборах переменных (в случае АБС речь идет об атомарных пропозициональных формулах), над которыми они построены².

Пусть задан некоторый алфавит $A = \{x_1, \dots, x_n\}$. Первичной структуре АБС будем сопоставлять набор подмножеств (подалфавитов) этого множества $PS_I = \{A_i\}_{i \in I}$, где I — некоторое подмножество множества индексов подалфавитов A , причем элементы этого набора не поглощают друг друга и вместе покрывают A :

$$\forall i, j \in I \quad A_i \not\subseteq A_j; \quad \bigcup_{i \in I} A_i = A.$$

Набору подалфавитов PS_I соответствует гиперграф [1, 14] H_I , вершинами которого являются элементы A , а ребрами — элементы PS_I . Такой гиперграф является *минимальным* — каждая его вершина входит в какое-либо ребро, каждое ребро содержит какую-либо вершину и не содержит никакого другого ребра.

Теперь рассмотрим ненаправленный граф, в котором каждой вершине приписан подалфавит из PS_I . Подалфавит, соответствующий вершине, будем называть ее *нагрузкой*. *Сепаратором* двух вершин называется пересечение их нагрузок. Две вершины называются *сочлененными*, если их сепаратор непуст. *Графом смежности* называется граф, в котором каждое ребро проведено между сочлененными вершинами и между каждой парой сочлененных вершин существует путь (называемый *магистралью*

² На самом деле речь идет не «переменных», поскольку АБС использует логико-вероятностный подход и вероятности задаются над пропозициональными формулами, однако для формализации этого пояснения потребовалось ввести целую систему понятий, приведенную, например, в [6].

ным путем), в котором нагрузка каждой вершин содержит сепаратор его концов. Вторичной структурой АБС может выступать только граф смежности [11].

Вторичная структура требуется в частности для осуществления глобального логико-вероятностного апостериорного вывода, который заключается в распространении влияния поступившего в сеть свидетельства [7] (т.е. последовательного осуществления апостериорного логико-вероятностного вывода). При этом сложность распространения влияния свидетельства (его пропагации) в каждый узел экспоненциально зависит от размера этого узла [3].

Преобразование первичной структуры алгебраической байесовской сети к ациклической

Гиперграф называется *ациклическим*, если для него существует ациклический граф смежности [15].

Первичным графом гиперграфа называется граф, в котором множество вершин совпадает с множеством вершин гиперграфа и две вершины соединены ребром тогда и только тогда, когда они входят в какое-либо ребро гиперграфа. Первичный граф гиперграфа, соответствующего первичной структуре АБС, называется ее *протоструктурой*.

Цикл в графе называется *простым*, если в нем нет хорд, т.е. ребер между несмежными по циклу вершинами. Граф называется *хордальным*, если в нем нет простых циклов длины больше трех. *Максимальным кликами* графа называется максимальные по включению полный подграфы графа. Гиперграф называется *конформным*, если множество его ребер совпадает с множеством вершин максимальных клик его первичного графа. Гиперграф называется *хордальным*, если хордален его первичный граф.

Теорема [15]. Гиперграф ацикличесок тогда и только тогда, когда он гиперграф конформен и хордален.

Для преобразования первичной АБС структуры к ациклической необходимо последовательно триангулировать протоструктуру, то есть за счет добавления ребер устранить все простые циклы длины больше трех, а затем из множеств вершин, входящих в максимальные клики полученной протоструктуры, построить подалфавиты, которые составляют новую первичную структуры АБС. Такая первичная структура по построению будет ациклическа.

Поскольку триангуляция осуществляется за счет добавления ребер, размер максимальных клик может увеличиться, причем неограниченно. Соответственно, максимальный размер максимальной клики также может увеличиться, причем неограниченно. Поскольку размер максимальной клики равен размеру подалфавита, то он определяет размерность случайного элемента из первичной структуры.

Цикличность первичной структуре алгебраической байесовской сети

В предыдущем разделе было показано, что преобразование первичной структуры к ациклической ведет к росту сложности алгоритмов логико-вероятностного вывода, поэтому допустимо не всегда. Соответственно, возникает потребность в работе с циклической первичной структурой, для которой пока не существует алгоритмов пропагации свидетельств по сети.

Разработка таких алгоритмов является открытым вопросом, однако уже на текущий момент исследовательским коллективом были получены (но не опубликованы) частичные результаты, основанные на многократной пропагации по циклу поступившего свидетельства и свидетельств, соответствующих элементам, стоящим в узлах цикла. Полученные результаты свидетельствуют, что последовательная пропагация таких свидетельств сходится.

Соответственно, актуальным является вопрос о построении вторичной структуры такой сети, и он решается за счет построения минимальных по числу ребер графов смежности. По определению такие графы можно построить над любой первичной структурой АБС, и было показано, что множество таких графов совпадает с множеством деревьев смежности в случае ациклическости первичной структуры АБС [10]. Также были предложены алгоритмы синтеза таких графов [2, 9].

Таким образом, компромисс между ациклическостью и связностью может быть найден в фиксации верхней границы размера элементов первичной структуры и устранением некоторых циклов из первичной структуры таким образом, чтобы размер элементов первичной структуры не превысил зафиксированный. Затем над преобразованной первичной структурой следует в качестве вторичной структуры построить минимальный граф смежности.

В работе [12] были описаны методы устранения циклов, которые основаны на последовательном добавлении к первичной структуре новых элементов, которые устраняют особые циклы — небратские полусиблинговые простые циклы. Метод позволяет устранить все циклы из первичной структуры. Если учесть ограничение на размер элемента, то можно добавлять лишь те элементы, размер которых не превосходит установленный. Таким образом, задача при поставленном ограничении будет решена.

Заключение

В работе была описаны вступающие друг с другом в противоречие требования к первичной структуре алгебраической байесовской сети, одно из которых состоит в возможности построить над ней ациклическую вторич-

ную структуру, а второе — в ограничении размера элементов первичной структуры. Поскольку преобразование первичной структуры к ациклической ограничено применимо, возникает необходимость как в разработке алгоритмов логико-вероятностного вывода на цикличной вторичной структуре, так и решений, связанных с построением вторичной структуры над цикличной первичной структурой.

В работе был предложен подход, основанный на устранении лишь на устранении лишь некоторого числа циклов таким образом, чтобы размер элементов первичной структуры АБС не превысил зафиксированный, и последующим построением минимального по числу ребер графа смежности.

Литература

1. *Зыков А.А.* Гиперграфы // Успехи математических наук. 1974. №6 (180). С. 86–154.
2. *Опарин В.В., Тулупьев А.Л.* Синтез графа смежности с минимальным числом ребер: формализация алгоритма и анализ его корректности // Труды СПИИРАН. СПб.: Наука, 2009. Вып. 11. С. 142–157.
3. *Сироткин А.В.* Вычислительная сложность алгоритмов локального апостериорного вывода в алгебраических байесовских сетях // Труды СПИИРАН. 2011. Вып. 3(18). С. 188–214.
4. *Тулупьев А.Л.* Ациклические алгебраические байесовские сети: логико-вероятностный вывод // Нечеткие системы и мягкие вычисления: Научный журнал Российской ассоциации нечетких систем и мягких вычислений. 2006. Том 1, № 1. С. 57–93.
5. *Тулупьев А.Л.* Байесовские сети: логико-вероятностный вывод в циклах. СПб.: Изд-во С.-Петербургского ун-та, 2008. 140 с. (Элементы мягких вычислений).
6. *Тулупьев А.Л., Николенко С.И., Сироткин А.В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
7. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петерб. ун-та, 2009, 400 с.
8. *Фильченков А.А.* Меры истинности и вероятностные графические модели для представления знаний с неопределенностью // Труды СПИИРАН. 2012. Вып. 4(23). С. 254–295.
9. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. Вып. 11. С. 104–127.
10. *Фильченков А.А., Тулупьев А.Л.* Связность и ацикличность первичной структуры алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2013. Вып. 1. С. 110–119.
11. *Фильченков А.А., Тулупьев А.Л.* Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. Вып. 2. С. 65–74.
12. *Фроленков К.В., Фильченков А.А.* Методы устранения циклов вторичной структуры алгебраической байесовской сети // «Гибридные и синергетические интел-

- лектуальные системы: теория и практика» (29 июня — 2 июля 2012 г., Калининград). Материалы 1–го международного симпозиума. Т. 2. Калининград: БФУ им. Канта. 2012. С. 282–292.
13. *Ширяев А.Н.* Вероятность. М.: Наука, 1989. 640 с.
 14. *Щербина О.А.* Локальные алгоритмы и древовидная декомпозиция для задач дискретной оптимизации // *Динамические системы: межвед. науч. сб. ТНУ*, 2006. Вып. 20. С. 89–103.
 15. *Beeri C., Fagin R., Maier D., Yannakakis M.* On the Desirability of Acyclic Database Schemes // *Journal of the ACM*. 1983. Vol. 30, iss. 3. P. 479–513.
 16. *Daphne Koller, Nir Friedman.* Probabilistic Graphical Models: Principles and Techniques. Cambridge: The MIT Press. 2009.
-

АЛГОРИТМ РАСЧЕТА ОЦЕНКИ ЗАЩИЩЕННОСТИ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ЗА СЧЕТ ЭЛИМИНАЦИИ МАЛОВЕРОЯТНЫХ ТРАЕКТОРИЙ СОЦИО-ИНЖЕНЕРНЫХ АТАК

Соловцов Н.Б.

*студент кафедры информатики,
математико-механический факультет СПбГУ, Nekit.tg@gmail.com.*

Аннотация: Для обеспечения деятельности специалистов в области информационной безопасности необходимо разработать научно-обоснованные и отражающие специфику предметной области математические методы и модели, позволяющие автоматизировать анализ защищенности пользователей информационных систем от социо-инженерных атак. Целью настоящей работы является рассмотрение метода поиска вероятности успеха социо-инженерного атакующего воздействия на каждого пользователя в комплексе «персонал информационная система — критичные документы», пользователи которого и связи между ними представлены в виде графа. Алгоритм предполагает поиск всевозможных ациклических путей между двумя пользователями.

Введение

Защита конфиденциальной информации неразрывно связана как с программно-технической защищенностью информационных систем, так и с защищенностью пользователей таких систем от негативного влияния социоинженеров извне. Вопросу анализа защищенности программно-технической составляющей информационных систем посвящено немало внимания [6, 21, 22–28], в то время как анализ защищенности пользователей информационных систем от социо-инженерных (социотехнических) атак находится на ранней стадии исследований [1–4, 13–16]. Для обеспечения деятельности специалистов в области информационной безопасности необходимо разработать научно-обоснованные и отражающие специфику предметной области математические методы и модели, позволяющие автоматизировать анализ защищенности пользователей информационных систем от социо-инженерных (социотехнических) атак.

Целью настоящей работы является рассмотрение метода поиска вероятности успеха социо-инженерного атакующего воздействия на каждого пользователя в комплексе «персонал информационная система — критичные документы» [1–4, 13–16], пользователи которого и связи между ними представлены в виде графа [5.7.9–12, 17–20]. Алгоритм предполагает поиск всевозможных ациклических путей между двумя пользо-

вателями. Также будет приведена иллюстрация метода расчета полной вероятности системы на упрощенном (для доступности и краткости изложения) примере.

Вероятность успешной реализации атаки на пользователя.

Подсчет вероятности ответных действий определенных пользователей сети на социо-инженерные атакующие воздействия злоумышленника позволяет судить о защищенности данного «узла» системы, то есть пользователя, но не о защищенности системы в целом. Для подсчета совокупной вероятности защищенности информационной системы от социо-инженерных атак злоумышленника можно пользоваться несколькими эвристиками. В настоящей работе предложен подход к подсчету такой вероятности, сводящийся к поиску всевозможных путей в графе и последующей комбинации весов их рёбер. Общая схема ключевых шагов подобного алгоритма выглядит следующим образом. Изначально у злоумышленника есть один кандидат для атаки. Для удобства, и, не умаляя общности, присвоим ему номер один. После этого предполагаем, что у любого из пользователей может оказаться доступ к требуемому злоумышленнику файлу, и ищем вероятности успешности социо-инженерной атаки злоумышленника на каждого пользователя, учитывая при этом веса перехода (веса рёбер) от пользователя к пользователю; причём веса вероятности успеха установления контакта злоумышленника со вторым пользователем.

На рис. 1 представлен пример графа, представляющего сложившиеся социальные связи персонала информационной системы. Каждому узлу графа соответствует пользователь информационной системы. Поэтому вес узла — вероятность успешности социо-инженерного атакующего воздействия злоумышленника на того пользователя, которому соответствует данный узел графа. В данном случае, под вероятностью успешности социо-инженерного атакующего воздействия злоумышленника понимается совокупная вероятность успешности социо-инженерной атаки на пользователя в случае применения всех атакующих воздействий. Ребра графа соответствуют взаимоотношения между пользователями. Каждое ребро графа, то есть связь между пользователями, имеет собственный вес, который соответствует вероятности успешного перехода по этой связи в случае социо-инженерного воздействия злоумышленника. В данном графе мы рассматриваем двунаправленные связи ради упрощения рассматриваемой модели. Вообще говоря, в реальной ситуации, при анализе защищенности пользователей, связи между двумя пользователями односторонни и, переходя к графовой структуре, каждая из них имеет свою вероятность успешного перехода в случае социо-инженерного воздействия злоумышленника.

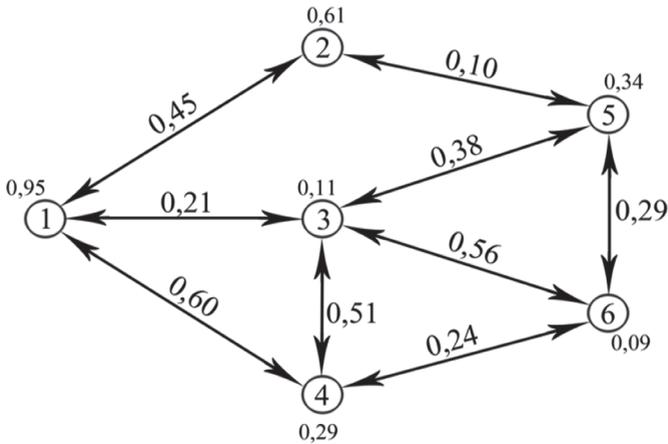


Рис. 1. Пример графа социальных связей персонала информационной системы

Пусть P_i — вероятность успешности атаки на i -го сотрудника, если у злоумышленника есть на него выход. $P_{i,j}$ — вероятность выхода злоумышленника на пользователя j через пользователя i если пользователь i уже успешно атакован. Тогда вероятность успешного выхода злоумышленника на пользователя j через путь, проходящий через пользователей i_k и начинающийся с пользователя m : $\tilde{P}_{m \dots i_k \dots j} = \prod_{k=1}^{n-1} P_m(P_{i_k i_{k+1}} P_{i_{k+1}})$, где $i_1 = m$, $i_n = j$. Теперь, когда у нас есть вероятность успешного выхода злоумышленника на пользователя j через цепь пользователей, вычислим формулу атаки через несколько цепей пользователей \tilde{P}_i : $\tilde{P}_{\text{sum}j} = 1 - \sum_{i=1 \dots k} (\tilde{P}_i)$. Соответственно чтобы $\tilde{P}_{\text{sum}j}$ была полной вероятностью успеха атаки на пользователя j , нужно, чтобы цепи были полным дизъюнктивным (с точки зрения теории вероятностей) множеством всевозможных путей, либо аппроксимировали его.

Преимущества и недостатки рассматриваемого подхода

В настоящей работе предложен эвристический подход, при котором учитываются всевозможные пути в графе отношений пользователей, не содержащие циклов. Преимуществами такого подхода является то, что он позволяет рассчитать именно искомую вероятность успеха социо-инженерной атаки злоумышленника на пользователя информационной системы через сеть пользователей, а также прост в понимании, поскольку, фактически, сводится к формуле суммы вероятностей дизъюнктивных событий. В то же время при больших объемах данных, возникает огромное число цепей

с небольшим весом (то есть соответствующих ничтожно малой вероятности успешного выхода злоумышленника на атакуемого пользователя). Кроме того, для каждого из пользователей вероятность его «поражения», «успеха социо-инженерной атаки» нужно рассчитывать отдельно.

Подчеркнем пару особенностей предложенного подхода.

Свойство 1. Вероятность успеха атаки на пользователя зависит в большей степени от минимальной длины цепочки, которую можно построить между первой жертвой злоумышленника и атакуемым пользователем.

Свойство 2. Вероятности успешных переходов по длинным цепочкам ничтожно малы по сравнению с вероятностями, вычисленными для более коротких цепочек, соответственно, они вносят малозаметный вклад в итоговую оценку.

Определение. Длинными будем называть цепочки, вероятность успешного социо-инженерного атакующего воздействия по которым меньше, скажем, 1 %, а также получающиеся из них путем добавления новых узлов.

На основании данных свойств можно предложить критерий отброса длинных цепочек, имеющих ничтожно малый вес в конечном результате.

Критерий. Рассматривать можно только цепочки минимальной длины и цепочки, длина которых больше этой длины на два – три звена. При этом из рассмотрения следует исключить более длинные цепочки.

При применении данного критерия работа алгоритма на больших объемах данных значительно ускорится в силу рассмотрения меньшего числа вариантов развития социо-инженерных атак.

Заключение

В настоящей работе рассмотрен подход к вычислению оценки вероятности успеха социо-инженерного атакующего воздействия на каждого пользователя в комплексе «персонал – информационная система – критичные документы», представленного в виде графа (для изложения основных принципов было достаточно ограничиться графом социальных связей пользователей). Предложенный подход сводится к поиску всевозможных ациклических путей между двумя пользователями в графе. Выделены особые свойства подхода, на основе которых предложен критерий, позволяющий уменьшить вычислительную сложность поиска полной вероятности успеха социо-инженерного атакующего воздействия на пользователя информационной системы. Следует отметить, что на практике может потребоваться критерий, более тонко характеризующий «длинные» цепочки и «малые вероятности»; однако такая «настройка» критерия будет в значительной степени определяться конкретной ситуацией. Принцип же оптимизации вычислений за счет отброса особо длинных цепочек с особо малыми вероятностями успеха реализации атакующих действий в таком случае останется неизменным.

Литература

1. *Азаров А.А., Тулупьева Т.В., Фильченков А.А., Тулупьев А.Л.* Вероятностно-реляционный подход к представлению модели комплекса «Информационная система — персонал — критичные документы». // Труды СПИИРАН. 2012. Вып. 20. С. 57–71.
2. *Азаров А.А., Тулупьева Т.В., Тулупьев А.Л.* Прототип комплекса программ для анализа защищенности персонала информационных систем построенный на основе фрагмента профиля уязвимостей пользователя. // Труды СПИИРАН. 2012. Вып. 21. С. 21–40.
3. *Азаров А.А., Тулупьев А.Л., Тулупьева Т.В.* SQL-представление реляционно-вероятностных моделей социо-инженерных атак в задачах расчета агрегированных оценок защищенности персонала информационной системы // Труды СПИИРАН. 2012. Вып. 22. С. 31–44.
4. *Ванюшичева О.Ю.* Прототип комплекса программ для построения профиля психологически обусловленных уязвимостей пользователя. Дипломная работа. СПб.: СПбГУ, 2012.
5. *Зельтерман Д., Суворова А.В., Пащенко А.Е., Мусина В.Ф., Тулупьев А.Л., Тулупьева Т.В., Гро Л.Е., Хаймер Р.* Диагностика регрессионных уравнений в анализе интенсивности рискованного поведения по его последним эпизодам // Труды СПИИРАН. 2011. Вып. 17. С. 33–46.
6. *Котенко И.В., Юсупов Р.М.* Перспективные направления исследований в области компьютерной безопасности. Защита информации. Инсайд. 2006. №2. С. 46.
7. *Пащенко А.Е., Тулупьев А.Л., Суворова А.В., Тулупьева Т.В.* Сравнение параметров угрозообразующего поведения в разных группах на основе неполных и неточных данных // Труды СПИИРАН. 2009. Вып. 8. СПб.: Наука, 2009. С. 252–261.
8. *Петренко С.А.* Возможная методика построения системы информационной безопасности предприятия. // URL: <http://bre.ru/security/13985.html> (дата обращения 10.01.12)
9. *Пинский М.Я., Сироткин А.В., Тулупьев А.Л., Фильченков А.А.* Повышение быстродействия алгоритма оценки наблюдаемой последовательности в скрытых марковских моделях на основе алгебраических байесовских сетей // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2011. Вып. 5. С. 69–73.
10. *Сироткин А.В., Тулупьев А.Л., Фильченков А.А., Пащенко А.Е., Тулупьева Т.В., Мусина В.Ф.* Особенности вероятностных графических моделей комплекса «Информационная система — персонал» для оценки его защищенности от социоинженерных атак // Научная сессия НИЯУ МИФИ-2011. (1–5 февраля 2011 г., Москва.) Аннотации докладов. В 3 т. Т. 3: Стратегические информационные технологии в атомной энергетике и промышленности. Проблемы информационной безопасности в системе высшей школы. Экономические и правовые проблемы инновационного развития атомной отрасли. Образование в Национальном исследовательском ядерном университете. М.: НИЯУ МИФИ, 2011. С. 80.
11. *Суворова А.В., Тулупьев А.Л., Пащенко А.Е., Тулупьева Т.В., Красносельских Т.В.* Анализ гранулярных данных и знаний в задачах исследования социально значимых видов поведения // Компьютерные инструменты в образовании. №4. 2010. С. 30–38.
12. *Суворова А.В., Пащенко А.Е., Тулупьева Т.В.* Оценка характеристик сверхкороткого временного ряда по гранулярным данным о рекордных интервалах между

- событиями // Труды СПИИРАН. 2010. Вып. 12. С. 170–181.
13. Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В. Социально-психологические факторы, влияющие на степень уязвимости пользователей автоматизированных информационных систем с точки зрения социоинженерных атак // Труды СПИИРАН. 2010. Вып. 1 (12). С. 200–214.
 14. Тулупьев А.Л., Азаров А.А., Пащенко А.Е. Информационные модели компонент комплекса «Информационная система — персонал», находящегося под угрозой социоинженерных атак // Труды СПИИРАН. 2010. Вып. 3 (14). С. 50–57.
 15. Тулупьев А.Л., Азаров А.А., Тулупьева Т.В., Пащенко А.Е., Степашкин М.В. Генерализация моделей деревьев атак на случай социоинженерных атак // Научная сессия МИФИ-2011. Аннотации докладов. В 3 т. Т. 3. М.: МИФИ, 2011. С. 89.
 16. Тулупьева Т.В., Тулупьев А.Л., Азаров А.А., Пащенко А.Е. Психологическая защита как фактор уязвимости пользователя в контексте социоинженерных атак // Труды СПИИРАН. 2011. Вып. 18. С. 74–92.
 17. Тулупьев А.Л., Фильченков А.А., Вальтман Н.А. Алгебраические байесовские сети: задачи автоматического обучения // Информационно-измерительные и управляющие системы. 2011. № 11, т. 9. С. 57–61.
 18. Фильченков А.А., Тулупьев А.Л. Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. Вып. 2. С. 65–74.
 19. Фильченков А.А., Тулупьев А.Л., Сироткин А.В. Структурный анализ клик максимальных графов смежности алгебраических байесовских сетей // Вестн. Тверск. гос. ун-та. Сер.: Прикладная математика. 2011. № 20. С. 139–151.
 20. Фильченков А.А., Тулупьев А.Л. Анализ циклов в минимальных графах смежности алгебраических байесовских сетей // Труды СПИИРАН. 2011. Вып. 2 (17). С. 151–173.
 21. Юсупов Р., Пальчун Б.П. Безопасность компьютерной инфо сферы систем критических приложений. Вооружение. Политика. Конверсия. 2003. № 2. С. 52.
 22. Dorothy D.E. A Lattice Model of Secure Information Flow // Communications of the ACM. 2008. Vol. 19.No. 5. p. 236–243.
 23. Balepin I., Maltsev S., Rowe, J., Levitt K. Using specification-based intrusion detection for automated response //Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection. 2003. p. 135–154.
 24. Jahnke M., Thul C., Martini P. Graph based metrics for intrusion response measures in computer networks //LCN 2007: Proceedings of the 32nd IEEE Conference on Local Computer Networks. IEEEComputerSociety, LosAlamitos. 2007. Washington. DC. USA. p. 1035–1042.
 25. National Institute of Standards and Technology. URL: <http://www.nist.gov/index.html> (датаобращения 24.06.2012)
 26. Siemens. The total information security toolkit. URL: <http://www.cramm.com/>(дата обращения 24.06.2012)
 27. Software Engineering Institute. URL: <http://www.cert.org/octave/> (датаобращения 24.06.2012)
-

ЭЛИМИНАЦИЯ И ИДЕНТИФИКАЦИЯ ЦИКЛОВ ВО ВТОРИЧНОЙ СТРУКТУРЕ АЛГЕБРАИЧЕСКОЙ БАЙЕСОВСКОЙ СЕТИ НА ОСНОВЕ МЕТОДОВ ТЕОРИИ ДРЕВОВИДНОЙ ДЕКОМПОЗИЦИИ¹

Вяткин А.В.

*студент 542 группы математико-механического факультета СПбГУ,
dewshick@gmail.com*

Аннотация: Алгебраическая байесовская сеть — вероятностная графическая модель, позволяющая работать с интервальными оценками вероятности и вероятностной логикой.

В докладе рассмотрены алгоритмы идентификации и элиминации циклов во вторичной структуре алгебраической байесовской сети, необходимых для осуществления однозначного вывода на вторичной структуре.

Введение

Задача извлечения информации из данных становится всё более актуальной. Для решения таких задач применяются вероятностные графические модели. Алгебраическая байесовская сеть [1] (АБС) — вероятностная графическая модель, позволяющая работать с интервальными оценками вероятности и вероятностной логикой.

АБС обладает вторичной структурой, представляющей собой неориентированный граф со случайными элементами особого вида в узлах. Для того, чтобы корректно осуществлять апостериорный вывод на вторичной структуре, необходимо, чтобы вторичная структура АБС была ациклической [1]. *Цель работы* — применить методы теории древовидной декомпозиции к устранению циклов во вторичной структуре.

Основные понятия теории АБС

Алфавитом называется множество различных атомарных пропозициональных формул. *Фрагментом знаний* над алфавитом называется пара из идеала конъюнктов над алфавитом и функции, сопоставляющей конъюнкту из идеала интервальную оценку истинности конъюнкта. *Первичной структурой АБС* над алфавитом, называется множество фрагментов знаний, построенных над взаимно непоглощающими подалфавитами алфавита. *Вторичная структура АБС* представляет собой неориентированный граф над первичной

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проекты № 12-01-00945-а и 12-01-31202-мол_а.

структурой АБС с особыми свойствами. *Гиперграф, соответствующий первичной структуре АБС* — это гиперграф, множество вершин которого совпадает с алфавитом, над которым построена АБС, а ребра совпадают с подалфавитами, над которыми построены фрагменты знаний, входящие в первичную структуру. *Протоструктурой* называется первичный граф гиперграфа, соответствующего первичной структуре.

Свойства цикличности вторичной структуры АБС

Ранее было показано [2], что все вторичные структуры, построенные над заданной первичной, одновременно цикличны или ацикличны. Таким образом, можно говорить о цикличности первичной структуры, которая в свою очередь эквивалентна цикличности гиперграфа, соответствующего ей [2]. Минимальный гиперграф цикличен, если его первичная структура триангулярна и он конформен первичной структуре [10]. *Триангулярным* называется граф, у которого в каждом цикле длины, большей четырех, есть *хорда* — ребро, соединяющее несмежные в цикле вершины.

Алгоритмы определения и устранения циклов во вторичной структуре АБС

Обоснование алгоритма

Для того, чтобы определить цикличность вторичной структуры АБС, достаточно построить по первичной структуре АБС её протоструктуру и определить триангулярность последней. Также необходимо проверить конформность гиперграфа первичной структуры.

В случае обнаружения циклов, для их устранения во вторичной структуре достаточно построить протоструктуру и *триангулировать* её, т. е. добавить ребра в протоструктуру таким образом, чтобы получившийся граф стал триангулярным.

Алгоритмы теории древовидной декомпозиции, применяемые для идентификации циклов вторичной структуры АБС

Как уже было сказано, для решения задачи идентификации цикличности вторичной структуры АБС достаточно построить её протоструктуру и определить коформность её гиперграфу и её триангулярность. Задача построения протоструктуры по первичной структуре решается за время, квадратичное от размера первичной структуры. Для задачи определения триангулярности существует алгоритм, работающий за линейное от размера протоструктуры время [11]. Проверка конформности гиперграфа требует построения конформного протоструктуры графа и сравнения этого графа с графом пер-

вичной структуры. Для задачи построения гиперграфа, конформного протоструктуре, также требуется линейное от размера протоструктуры время благодаря предположению о триангулярности протоструктуры (если она не триангулярна, то гиперграф цикличен и нет необходимости проверять его конформность). Сравнить на равенство гиперграфы можно за полиномиальное время.

Таким образом, алгоритмы для решения задачи определения цикличности первичной структуры работают за полиномиальное время.

Алгоритмы теории древовидной декомпозиции, применяемые для элиминации циклов вторичной структуры АБС

Задача устранения циклов во вторичной структуре АБС аналогично задаче определения цикличности полиномиально сводима к задаче теории древовидной декомпозиции. Но хотя задача триангуляции графа в общем случае полиномиально сложна, на полученных после такого изменения вторичных структур алгоритмы апостериорного вывода могут работать недопустимо медленно. Это связано с тем, что время выполнения существующих алгоритмов апостериорного вывода экспоненциально зависит от максимального размера фрагмента знаний, входящего в получаемую первичную структуру. В данном случае необходимо искать триангуляцию протоструктуры *минимальной ширины* — триангуляцию, минимизирующую максимальный размер полного подграфа протоструктуры. Задача построения триангуляции минимальной ширины является NP-полной и для её решения на практике используются эвристики [4, 5, 6, 8].

Один из наиболее популярных подходов к задаче триангуляции графа заключается в том, что триангуляция строится по последовательности вершин графа. Такой подход называется триангуляцией по *элиминирующей последовательности*. Принцип этого подхода заключается в том, что поочередно для каждой вершины последовательности выполняются следующие действия: подграф, индуцируемый смежными с ней вершинами дополняется до клики.

Благодаря применению подхода триангуляции по элиминирующей последовательности, задача триангуляции сводится к задаче поиска последовательности, которая решается многими эвристическими алгоритмами поиска, как то генетические алгоритмы или метод отжига.

Заключение

В работе были представлены подходы к идентификации и элиминации циклов во вторичной структуре АБС. Важность изложенных методов обусловлена возможностью их применения при построении первичной структуры, для улучшения общей производительности АБС.

Литература

1. *Тулупьев А.Л., Николенко С.И., Сироткин А.В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
 2. *Фильченков А.А.* Приложение теории древовидной декомпозиции в машинном обучении глобальных структур алгебраических байесовских сетей // Труды СПИИРАН. 2013. Вып. 4 (27) [в печати].
 3. *Фильченков А.А., Тулупьев А.Л.* Анализ циклов в минимальных графах смежности алгебраических байесовских сетей. // Тр. СПИИРАН. 2011. Вып. 17. С. 151–173.
 4. *Berry A., Heggernes P., Simonet G.* The minimum degree heuristic and the minimal triangulation process // Proceedings WG 2003 — 29th Workshop on Graph Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, Springer Verlag, 2003. . Vol. 2880. P. 58–70.
 5. *Bodlaender H.L., Koster A., van den Eijkhof F., van der Gaag L.C.* Pre-processing for Triangulation of Probabilistic Networks // UAI'01 Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, 2001, pp. 32–39
 6. *Cano A., Moral S.* Heuristic Algorithms for the Triangulation of Graphs // Proceedings of the Fifth IPMU Conference. Springer, 1995. P. 166–171.
 7. *Golumbic M.C.* Algorithmic Graph Theory and Perfect Graphs. NY: Academic Press, 1980. 286 p.
 8. *Kjærulff U.* Optimal decomposition of probabilistic networks by simulated annealing // Statistics and Computing. 1992. Vol. 2. P. 7–17.
 9. *Larranaga P., Kuijpers C., Poza M., Murga R.H.* Decomposing Bayesian Networks: Triangulation of Moral Graph with Genetic Algorithms // Statistics and Computing (UK), 1997, 7(1). P. 19–34.
 10. *Maier D.* The theory of relational databases // Computer science press, 1983, 656 pp.
 11. *Tarjan R.E., Yannakakis M.* Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs // SIAM Journal on Computing. 1984. Vol. 13(2). P. 566–579.
-

РАВНОВЕРОЯТНОСТНЫЙ СИНТЕЗ РЕАЛИЗАЦИИ СЛУЧАЙНОГО МИНИМАЛЬНОГО ГРАФА СМЕЖНОСТИ¹

Фильченков А.А.

*аспирант кафедры информатики СПбГУ, м. н. с. лаб. ТуМПИ СПИИРАН,
aafil@mail.ru*

Мусина В.Ф.

*магистрант экономического факультета СПбГУ,
м. н. с. лаб. ТуМПИ СПИИРАН, valery.musina@gmail.com*

Аннотация: Цель исследования — предложить алгоритм, который по заданному набору подалфавитов, соответствующих первичной структуре алгебраической байесовской сети, случайным образом строит минимальный граф смежности, причем все возможные реализации минимальных графов смежности строятся равновероятно.

Введение

Алгебраические байесовские сети (АБС) относятся к классу логико-вероятностных графических моделей [11, 19] и используются для представления сложных систем знаний с неопределенностью. Узлами алгебраической байесовской сети [2, 3] выступают сложные случайные элементы [16] (и даже семейства таких случайных элементов, единых по структуре, но различающихся по значениям параметров распределений) — математические модели фрагментов знаний (фрагменты знаний) [2, 3]. Набор максимальных по включению фрагментов знаний (МФЗ) формирует первичную структуру алгебраической байесовской сети, а граф, построенный над ними — ее вторичную структуру [4].

Проблема синтеза вторичной структуры АБС по ее первичной структуре сводится к задаче построения минимального графа смежности (МГС) по набору подалфавитов, над которыми построены фрагменты знаний [4, 5, 13, 18]. Для исследования как свойств минимальных графов смежности, так и численных характеристик осуществления алгоритмов глобального логико-вероятностного вывода [1] можно было бы изучать все минимальные графы смежности, однако это неэффективно, поскольку множество минимальных графов смежности велико [14, 15] и алгоритм синтеза этого множества затратен [6–9]. В работе [12] было предложено исследовать случайные выборки из этого множества. В ней же была поставлена задача построения минимального графа смежности случайным образом и предложен алгоритм рандомизированного синтеза такого графа с ненулевой вероятностью. Од-

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проекты № 12-01-00945-а и 12-01-31202-мол_а.

нако просто рандомизированного синтеза недостаточно: необходимо управлять распределением, которое задается работой алгоритма.

Интерес представляет, в первую очередь, равномерное распределение: алгоритм, работа которого обеспечивает реализацию такого вероятностного распределения, позволит статистически корректно оценивать частоту встречаемости тех или иных свойств у минимальных графов смежности, а также позволит привлечь генетические алгоритмы для синтеза минимального графа смежности, который минимизирует время работы алгоритмов пропации или же уменьшает потерю точности оценок при пропации в случае интервальных оценок вероятности.

Цель статьи — предложить алгоритм, который по заданному набору подалфавитов, соответствующих первичной структуре АБС, строит случайным образом минимальный граф смежности, причем все возможные реализации минимальных графов смежности строятся равновероятно.

Глобальные структуры алгебраических байесовских сетей

Граф — пара $\langle V, E \rangle$, где V — множество вершин графа, а E — множество ребер. Через V и E будем обозначать функции от графа, возвращающие множество его вершин и множество его ребер соответственно. Деревом называется связный граф без циклов.

В дальнейших рассуждениях будем опираться на алгоритм Прюфера [20] для построения алгоритма равновероятного синтеза минимального графа смежности. Рассмотрим ненаправленный граф из n вершин, которым произвольным образом присвоены номера с 0 по $n-1$. Для удобства дальнейшей работы введем определение бирки.

Биркой размера n будем называть $n-2$ -местный кортеж из чисел от 0 до $n-1$. Код *Прюфера* для дерева из n пронумерованных вершин называется бирка размера n , которая кодирует соответствующее дерево.

Так как алгоритм Прюфера восстанавливает по коду Прюфера соответствующее коду дерево, а кодом может быть любая бирка, то перебирая каждую бирку и восстанавливая по ней дерево, мы переберем все деревья на данном множестве вершин.

Теорема [15]. Любая бирка Прюфера $p_1 p_2 \dots p_{n-2}$ задает дерево, степени вершин которого равны числу вхождений номера вершины в кортеж $(p_1 p_2 \dots p_{n-2})$, увеличенному на один.

В дальнейших рассуждениях будем обозначать степень вершины графа как $d(v)$. Если обозначить $\alpha(i)$, $i=1 \dots n$ количество повторений индекса i в коде Прюфера, то было показано [15], что

$$\alpha(i) + 1 = d(v_i), \quad (1)$$

где v_i — вершина дерева, занумерованная индексом i .

Полная формализация глобальных структур алгебраической байесовской сети может быть найдена в работах [10, 13, 16, 17]. В дальнейшем будем полагать, что первичная структура зафиксирована.

Схема алгоритма равновероятного синтеза минимального графа смежности

Алгоритм равновероятного синтеза минимального графа смежности строит по первичной структуре АБС случайный минимальный граф смежности, причем каждый элемент множества МГС выбирается из множества МГС равновероятно. Он состоит из следующих шагов:

1. Построить множество $\text{Separator} = \{U_1, U_2, \dots, U_n\}$ всех сепараторов первичной структуры АБС.
2. Для каждого $U_a \in \text{Separator}$ построить сильное сужение $\Downarrow U_a$.
3. Разложить сужения $\Downarrow U_a$ на компоненты связности (владения) $\{c_1^a, c_2^a, \dots, c_{m_a}^a\}$. Пронумеровать компоненты связности числами $\{1, 2, 3, \dots, m_a\}$.
4. Задать код Прюфера диаметра $m_a : \{p_1^a, p_2^a, \dots, p_{m_a-2}^a\}$, определяющий однозначно дерево $H_{U_a}^{b_a}$ на элементах $\{c_1^a, c_2^a, \dots, c_{m_a}^a\}$, причем сам код Прюфера генерируется с вероятностью

$$P_a = \frac{\prod_{i=1}^{m_a} (v(c_i^a))^{\alpha(i)+1}}{\left(\prod_{i=1}^{m_a} v(c_i^a)\right) \left(\sum_{i=1}^{m_a} v(c_i^a)\right)^{m_a-2}}.$$

5. Для каждого ребра дерева $H_{U_a}^{b_a}$, соединяющего владения c_i^a и c_j^a , построить ребро между вершиной $g \in c_i^a$ и вершиной $g' \in c_j^a$, причем в каждом владении c_k^a вершина выбирается равновероятно с вероятностью $\frac{1}{v(c_k^a)}$. Набор ребер и их концы образуют жилу $S_{U_a}^{l_a}$.
6. Объединить жилы $S_{U_1}^{l_1}, S_{U_2}^{l_2}, \dots, S_{U_n}^{l_n}$.

Граф, полученный на последнем шаге работы алгоритма, является минимальным графом смежности по теореме о множестве минимальных графов смежности. Предложенный алгоритм генерирует минимальный граф смежности, согласно утверждению в работе [12]. Проверим, что любой граф из множества минимальных графов смежности может быть получен равновероятно при помощи предложенного алгоритма.

Теорема. Каждый элемент множества минимальных графов смежности выбирается равновероятно из множества всех минимальных графов смежности в результате работы алгоритма.

Доказательство. Пусть G является элементом множества минимальных графов смежности. Результатом алгоритма является реализация случайного минимального графа смежности G .

В дальнейшем верхним индексом * будем обозначать конкретную реализацию случайного элемента.

Покажем, что конкретный граф G выбирается из множества всех минимальных графов смежности равновероятно в результате работы алгоритма. Теорема о минимальных графах смежности [13, 16] утверждает, что, каждый минимальный граф смежности представляет собой пучок, т. е. объединение жил, выбранных по одной для каждого сепаратора. Обозначим через S_a случайный элемент, отвечающий выбору жилы для сильного сужения $\Downarrow U_a$ при реализации алгоритма.

$$P(G=G) = P\left(\bigcup_{a=1}^n S_a = \bigcup_{a=1}^n S_a^*\right) = \prod_{a=1}^n P(S_a = S_a^*).$$

Согласно предложенной схеме, жилы генерируются независимо. Таким образом, необходимо проверить, что каждая жила S_a^* , $a = 1, \dots, n$ выбирается из множества всех жил сильного сужения $\Downarrow U_a$ равновероятно.

Выбор конкретной жилы зависит от построения конкретного оммажа. Обозначим через H_a случайный элемент, отвечающий выбору оммажа для сепаратора U_a при реализации алгоритма. Таким образом,

$$P(S_a = S_a^*) = P(S_a = S_a^* \mid H_a = H_a^*) \cdot P(H_a = H_a^*).$$

Вероятность получить конкретный оммаж при реализации алгоритма равна вероятности выбрать конкретный код Прюфера, т. е. $P(H_a = H_a^*) = P_a$. При построении конкретной жилы мы обращаемся к владению c_i^a и выбираем из него вершину с вероятностью $\frac{1}{v(c_i^a)}$ столько раз, какова степень вершины в дереве H_a^* , таким образом, вероятность построить конкретную жилу при условии уже выбранного дерева есть

$$P(S_a = S_a^* \mid H_a = H_a^*) = \frac{1}{\prod_{i=1}^{m_a} (v(c_i^a))^{d(i)}}.$$

На шаге 4 алгоритма выписан вид вероятности P_a :

$$\begin{aligned} P(S_a = S_a^* \mid H_a = H_a^*) \cdot P(H_a = H_a^*) = \\ = \frac{\prod_{i=1}^{m_a} (v(c_i^a))^{\alpha(i)+1}}{\left(\prod_{i=1}^{m_a} v(c_i^a)\right) \left(\sum_{i=1}^{m_a} v(c_i^a)\right)^{m_a-2}} \cdot \frac{1}{\prod_{i=1}^{m_a} (v(c_i^a))^{d(i)}}. \end{aligned}$$

Согласно свойствам кода Прюфера, получим

$$P(S_a = S_a^*) = \frac{1}{\left(\prod_{i=1}^{m_a} v(c_i^a)\right) \left(\sum_{i=1}^{m_a} v(c_i^a)\right)^{m_a-2}}.$$

Таким образом, введенный алгоритм генерирует каждую жилу равномерно из всего множества жил для сильного сужения $\Downarrow U_a$. Доказательство окончено.

Заключение

Для практической реализации алгоритма необходимо разработать математическую модель реализации кода Прюфера с конкретной, установленной на шаге 4 алгоритма, вероятностью.

Алгоритм равномерностного синтеза минимального графа смежности, представленный в статье, позволяет получить реализацию случайного минимального графа смежности, выбранную равномерно из множества всех минимальных графов смежности. Возможность равномерного синтеза минимального графа смежности позволит оптимизировать подходы к изучению свойств вторичной структуры алгебраической байесовской сети и логико-вероятностного вывода на ней, а также будет полезна при генерации тестовых выборок.

Литература

1. Тулупьев А.Л. Апостериорные оценки вероятностей в алгебраических байесовских сетях // Вестник Санкт-Петербургского университета. Серия 10. Прикладная математика. Информатика. Процессы управления. 2012. № 2. С. 51–59.
2. Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
3. Тулупьев А.Л., Сироткин А.В., Николенко С.И. Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петербур. ун-та, 2009, 400 с.
4. Тулупьев А.Л., Столяров Д.М., Ментюков М.В. Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях // Труды СПИИРАН. 2007. Вып. 5. СПб.: Наука, 2007. С. 71–99.
5. Тулупьев А.Л., Фильченков А.А., Вальтман Н.А. Алгебраические байесовские сети: задачи автоматического обучения // Информационно-измерительные и управляющие системы. 2011. № 11, т. 9. С. 57–61.
6. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик // Труды СПИИРАН. 2010. Вып. 1 (12). С. 119–133.
7. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик-собственников // Труды СПИИРАН. 2010. Вып. 3 (14) С. 150–169.

8. *Фильченков А.А.* Алгоритм построения множества минимальных графов смежности при помощи клик владений // Труды СПИИРАН. 2010. Вып. 2 (13). С. 119–133.
 9. *Фильченков А.А.* Алгоритм построения множества минимальных графов смежности при помощи клик-собственников владений // Труды СПИИРАН. 2010. Вып. 4 (15). С. 193–212.
 10. *Фильченков А.А.* Иерархия глобальных структур алгебраической байесовской сети как система графов и гиперграфов // Научно-технический вестник информационных технологий, механики и оптики. 2013. Вып. 1(83). С. 75–80.
 11. *Фильченков А.А.* Меры истинности и вероятностные графические модели для представления знаний с неопределенностью // Труды СПИИРАН. 2012. Вып. 4(23). С. 254–295.
 12. *Фильченков А.А., Мусина В.Ф., Тулупьев А.Л.* Алгоритм рандомизированного синтеза минимального графа смежности // Тр. СПИИРАН. 2013. Вып. 2(25). С. 221–234.
 13. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. Вып. 11. С. 104–127.
 14. *Фильченков А.А., Тулупьев А.Л.* Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. Вып. 2. С. 65–74.
 15. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Мощность множества минимальных графов смежности // Труды СПИИРАН. 2010. Вып. 15. С. 136–161.
 16. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Особенности анализа вторичной структуры алгебраической байесовской сети // Труды СПИИРАН. 2010. Вып. 1 (12). С. 97–118.
 17. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Структурный анализ клик минимальных графов смежности // Вестн. Тверск. гос. ун-та. Сер.: Прикладная математика. 2011. №20. С. 139–151.
 18. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Управление глобальной структурой знаний в интеллектуальных системах, основанных на алгебраических байесовских сетях // Материалы конференции «Информационные технологии в управлении» (ИТУ–2012) (9–11 октября 2012 г., Санкт-Петербург). СПб.: ОАО «Концерн «ЦНИИ «Электроприбор». 2012. С. 25–33.
 19. *Koller D., Friedman N.* Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 2009. 1208 p.
 20. *Prufer H.* Neuer Beweis eines Satzes Uber Permutationen // Arch. Math. Phys. 1918. No. 27. P. 742–744.
-

ОЦЕНКИ СЛОЖНОСТИ АЛГОРИТМОВ СИНТЕЗА МИНИМАЛЬНЫХ ПО ЧИСЛУ РЕБЕР ГРАФОВ СМЕЖНОСТИ¹

Сироткин А.В.

*к. ф.-м. н., лаборант-исследователь лаб. исследования операций СПбГУ,
н. с. лаб. ТуМПИ СПИИРАН, alexander.sirotkin@gmail.com*

Тулупьев А.Л.

*д. ф.-м. н., доц., проф. кафедры информатики СПбГУ,
зав. лаб. ТуМПИ СПИИРАН, alexander.tulupyev@gmail.com*

Фильченков А.А.

*аспирант кафедры информатики СПбГУ, м. н. с. лаб. ТуМПИ СПИИРАН,
aafil@mail.ru,*

Фроленков К.В.

*аспирант кафедры информатики СПбГУ, м. н. с. лаб. ТуМПИ СПИИРАН,
frolenk@mail.ru*

Аннотация: В работе рассмотрены алгоритмы построения некоторых подмножеств множества минимальных графов смежности (множество таких графов, множество локально звездчатых графов смежности и одноэлементное подмножество, содержащее реализацию случайного минимального графа смежности), а также приведены оценки их сложности.

Введение

Вторичной структурой алгебраических байесовских сетей выступает граф с определенными свойствами, а именно — граф смежности [1–3]. Ацикличность такого графа является также желаемым свойством, однако не всегда может быть достижима [10]. Поэтому актуальными являются исследования относительно множества минимальных по числу ребер графов смежности — *минимальных графов смежности* (МГС) [4–7].

В работе будут рассмотрены алгоритмы построения выделенных подмножеств таких графов, а именно: всего множества МГС, подмножества МГС, являющихся локально звездчатыми графами, а также одноэлементных подмножеств, содержащих реализацию случайного минимального графа смежности. Все такие алгоритмы состоят из двух шагов, и первый совпадает: необходимо по множеству нагрузок, соответствующих первичной структуре, построить три других множества, по которым строятся подмножества минимальных графов смежности. Работа посвящена именно вторым

¹ Работа выполнена при финансовой поддержке РФФИ, гранты № 12-01-00945-а, 12-01-31202-мол_а.

шагам алгоритмов для построения указанных подмножеств. Цель работы — дать оценки сложности построения указанных подмножеств.

Основные понятия

Графы смежности

В дальнейшем изложении мы будем опираться на систему терминов, изложенную в работах [11–16]. *Нагруженным графом* называется тройка $\langle G, A, W \rangle$, где G — ненаправленный граф, A — алфавит, W — функция нагрузки, заданная на вершинах и ребрах G , принимающая значения из множества 2^A . *Сепаратором* двух вершин в нагруженном графе называется пересечение нагрузок соответствующих вершин. *Согласованным нагруженным графом* называется нагруженный граф, для которого нагрузка каждого ребра совпадает с сепаратором его концов.

Две вершины называются *сочлененными*, если их сепаратор непуст. *Граф максимальных фрагментов знаний (граф МФЗ)* — согласованный лексический граф, в котором ребра возможны только между сочлененными вершинами (т. е. нагрузка любого ребра непуста). *Магистральный путь* между двумя вершинами u и v в согласованном нагруженном графе — такой путь что нагрузка любой его вершины содержит сепаратор u и v . Согласованный лексический граф *магистрально связан*, если между каждой парой сочлененных вершин существует магистральный путь.

Граф смежности — магистрально связный граф МФЗ. В графе смежности возможны ребра только между сочлененными вершинами и между любой парой сочлененных вершин существует магистральный путь. *Минимальный граф смежности (МГС)* — граф смежности, число ребер которого минимально. *Максимальный граф смежности G_{\max}* — граф смежности, число ребер которого максимально, такой граф единственен [13].

Сужение $G \downarrow U$ согласованного лексического графа G на нагрузку U — это ненаправленный граф, в который входят только те вершины и ребра исходного графа G , нагрузки которых содержат или равны U . *Сильное сужение $G \Downarrow U$* — сужение $G \downarrow U$, из которого удалили все ребра с нагрузкой U . Сильное сужение графа $G \Downarrow U$ разбивается на компоненты связности, которые называются *владениями*.

Жила с нагрузкой U — набор ребер с нагрузкой U , число которых равно числу компонент сильного сужения $G \Downarrow U$ уменьшенного на единицу, а граф, полученный добавлением к $G \Downarrow U$ ребер жилы, является связным.

Структура ModelKit

Стереосепаратор — сепаратор, которому соответствует более одной жилы. *Обязательное ребро* — ребро, которое встречается в каждом минимальном графе смежности.

Через ModelKit будет обозначать четверку $\langle \text{Workloads}, \text{StereoSeparators}, \text{StereoHolding}, \text{NecessaryEdge} \rangle$, где Workloads — множество нагрузок, составляющих первичную структуру, StereoSeparators — множество стереосепараторов, построенное по данной первичной структуре, StereoHolding — множество компонент связности сильных сужений на каждый стереосепаратор, NecessaryEdge — множество обязательных ребер.

ModelKit используется как структура, по которой осуществляется синтез как элементов или подмножеств множества МГС, так и самого такого множества. Далее в этом разделе будут описаны алгоритмы синтеза каждого из указанных объектов, а в следующих разделах приведены алгоритмы синтеза уже самой ModelKit. Алгоритмы синтеза подмножеств МГС по ModelKit будем называть *алгоритмами сборки*.

Алгоритмы сборки минимальных графов смежности

Общая схема сборки подмножеств минимальных графов смежности

Приведем общую схему алгоритмов сборки (листинг 1).

```

MJGASSEMBLING
Require: Workloads, StereoSeparators, StereoHolding, NecessaryEdge, T
Ensure: StereoSeparators  $\cup$  NecessaryEdge  $\neq \emptyset$ 
1: for all  $u \in \text{StereoSeparators}$  do
2:    $\text{Sinews}[u] \leftarrow \text{SINEWS}(T, \text{StereoHolding}[u])$ 
3: end for
4:  $\text{EdgeSets} \leftarrow \text{UNITINGALGEBRAICFOLDING}(\text{Sinews})$ 
5:  $\text{MJGSubset} \leftarrow \emptyset$ 
6: for all  $E \in \text{EdgeSets}$ 
7:    $\text{MJGSubset} \leftarrow \text{MJGSubset} \cup \{ \langle \text{Workloads}, \text{UNITEDISJOINTSETS}(E, \text{NecessaryEdge}) \rangle \}$ 
8: end for
9: if  $\text{EdgeSets} \neq \emptyset$  do
10:    $\text{MJGSubset} \leftarrow \{ \langle \text{Workloads}, \text{NecessaryEdge} \rangle \}$ 
11: end if
12: return MJGSubset

```

Листинг 1. Алгоритм сборки подмножества МГС

В приведенном выше алгоритме используются вспомогательные функции. UNITEDISJOINTSETS объединяет два пересекающихся множества, сложность ее работы равна $O(1)$. UNITINGALGEBRAICFOLDING для заданного набора множеств множеств возвращает объединяющую алгебраическую свертку этого набора. Пускай всего n множеств, в которых c_1, c_2, \dots, c_n эле-

ментов соответственно, тогда сложность построения объединяющей алгебраической свертки для такого набора равна $O\left(n \prod_{i=1, \dots, n} c_i\right)$, а всего элементов в свертке ровно $\prod_{i=1, \dots, n} c_i$. SINEWS — абстрактная функция, которая для заданного типа жил и множества владений одного сепаратора (и, возможно, дополнительных аргументов) строит множество жил указанного типа соответствующего сепаратора. Варианты реализации функции SINEWS будут приведены далее.

Сложность работы алгоритма сборки подмножества МГС зависит от конкретной реализации SINEWS. Будем говорить, что реализация SINEWS *эффективна*, если ее сложность для каждого сепаратора равна $O(hM)$, где h — число владений этого сепаратора, а M — мощность синтезируемого подмножества. В случае эффективности алгоритма в указанном смысле, сложность синтеза подмножества равна $O(HM)$, где H — число владений всех сепараторов, то есть линейно зависит от размера множества.

Реализация SINEWS для построения множества минимальных графов смежности

Алгоритм построения множества всех жил по заданному множеству владений строит все возможные жилы соответствующего сепаратора. Алгоритм приведен на листинге 2.

```

SINEWSALL
Require: Holdings
Ensure: Holdings={C0, ..., Ch-1}; h>0; |ci|>1 ∀ i∈0, ..., h-1.
1: Sinews←∅
2: Homages←PRÜFERTREES(Hodings)
3: for all H∈Homages do
4:   for i←0 to |Holdings|-2
5:     TreeRealisationEdge[i]←∅
6:   end for
7:   i←0
8:   for all (U, V)∈E(H) do
9:     for all u∈Holdings[U], v∈Holdings[V] do
10:      TreeRealisationEdge[i]←TreeRealisationEdge[i]∪{{u, v}}
11:    end for
12:    i←i+1
13:   end for
14:   Sinews←Sinew ∪ UNITINGALGEBRAICFOLDING(TreeRealisationEdge)
15: end for
16: return Sinews

```

Листинг 2. Алгоритм построения множества всех жил для заданного сепаратора

В приведенном алгоритме функция PRÜFER TREES для заданного набора вершин возвращает множество возможных деревьев над этими вершинами.

Сложность работы алгоритма SINEWS ALL принадлежит классу $O(h^{h-2} \cdot (h + c^{2h-2}))$, где h — число владений сепаратора, а c — максимальное число вершин во владении. Поскольку для одного сепаратора можно построить $O(h^{h-2} c^{2h-2})$ жил [14], алгоритм построения всего множества эффективен.

Реализация SINEWS для построения подмножества локально звездчатых графов смежности

Локально звездчатым графом смежности называется граф смежности, в котором каждая жила является звездой. *Звездчатым графом смежности* называется локально звездчатый граф смежности, в котором для каждого сепаратора вершины соответствующей жилы являются центрами звезд жил, соответствующих сыновьям этого сепаратора.

Множество звездчатых графов смежности содержит минимальные графы смежности с минимальным диаметром [8], соответственно, локально звездчатые графы также содержат минимальные графы смежности с минимальным диаметром. Локально звездчатые графы смежности отличаются от звездчатых тем, что выбор жил производится независимо, тогда как выбор жил при построении звездчатых графов осуществляется последовательно и зависит от выбора предыдущих жил. Соответственно, в рамках настоящей работы мы рассмотрим алгоритмы синтеза всего множества локально звездчатых графов смежности. Алгоритм построения звездчатых графов смежности несложно получить за счет добавления таблицы, в которой для каждой жилы хранится ее центр и последовательным перебором стереосепараторов.

Алгоритм построения множества жил-звезд по заданному множеству владений строит все возможные жилы соответствующего сепаратора, являющиеся звездами. Алгоритм приведен на листинге 3.

```

SINEWSSTARS
Require: Holdings
Ensure: Holdings = {C0, ..., Ch-1}; h > 0; |ci| > 1 ∀ i ∈ 0, ..., h-1.
1: Sinews ← ∅
2: for all C ∈ Holdings do
3:   for all u ∈ C do
4:     EdgeRealisation ← ∅
5:     for all C' ∈ Holdings, C ≠ C' do
6:       EdgeRealisation|C'| ← ∅
7:       for all v ∈ C' do
8:         EdgeRealisation|C'| ← EdgeRealisation|C'| ∪ {(u, v)}
9:       end for
10:    end for

```

```

11:   Sinews ←
        Sinews ∪ UNITINGALGEBRAICFOLDING(TreeRealisationEdge)
12:   end for
13: end for
14: return Sinews

```

Листинг 3. Алгоритм построения множества жил-звезд для заданного сепаратора

Сложность работы алгоритма `SINEWSSTARS` принадлежит классу $O(h^2c^h)$, где h — число владений сепаратора, а c — максимальное число вершин во владении. Полученная оценка, оставаясь экспоненциальной, значительно меньше, чем оценка для всех жил вообще. Кроме того, поскольку жил-звезд всего $O(hc^h)$ алгоритм эффективен.

Реализация `SINEWS` для рандомизированного синтеза минимального графа смежности

Алгоритм рандомизированного синтеза минимального графа смежности по множеству владений сепаратора строит реализацию случайного минимального графа смежности, причем каждая реализация может быть построена с ненулевой вероятностью. Алгоритм основан на работе [9] и приведен на листинге 4.

```

SINEWSRANDOMONE
Require: Holdings
Ensure: Holdings = {c0, ..., ch-1}; h > 0; |ci| > 1 ∀ i ∈ 0, ..., h-1.
1: C ← CHOOSERANDOM(Hodings)
2: NotYetConnectedH ← Hodings \ {C}
3: AlreadyConnectedV ← V(C)
4: Edges ← ∅
5: while NotYetConnectedH ≠ ∅ do
6:   C ← CHOOSERANDOM(Hodings)
7:   v ← CHOOSERANDOM(AlreadyConnectedV)
8:   u ← CHOOSERANDOM(C)
9:   Edges ← Edges ∪ {v, u}
10:  NotYetConnectedH ← NotYetConnectedH \ {C}
11:  AlreadyConnectedV ← AlreadyConnectedV ∪ V(C)
12: end while
13: return {Edges}

```

Листинг 4. Алгоритм случайного построения жилы для заданного сепаратора

Функция `CHOOSERANDOM` равновероятно возвращает элемент заданного непустого множества. Будем считать, что сложность работы равна $O(1)$.

Сложность работы алгоритма `SINERANDOMONE` принадлежит классу $O(ch)$, где h — число владений сепаратора, а c — максимальное число вершин во владении. Поскольку строится всего один минимальный граф смежности, а сложность выше $O(h)$, алгоритм неэффективен в введенном смысле, однако поскольку само подмножество одноэлементно, это не является критичным для скорости синтеза такого подмножества.

Заключение

В работе рассмотрены три алгоритма построения трех подмножеств минимальных графов смежности по модели для сборки — четверки, состоящей из множеств нагрузок первичной структуры, множества стереосепараторов, жил таких стереосепараторов и обязательных ребер. Первое подмножество — само множество минимальных графов смежности; второе — множество локально звездчатых графов смежности, которые позволяют искать минимальные графы смежности с минимальным диаметром; третье — одноэлементное подмножество, состоящее из реализации случайного минимального графа смежности (причем любая реализация может быть получена с ненулевой вероятностью).

Сложность приведенных алгоритмов линейно зависит от мощности подмножества.

Литература

1. Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
2. Тулупьев А.Л., Сироткин А.В., Николенко С.И. Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петерб. ун-та, 2009. 400 с.
3. Тулупьев А.Л., Столяров Д.М., Ментюков М.В. Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях // Труды СПИИРАН. 2007. Вып. 5. СПб.: Наука, 2007. С. 71–99.
4. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик // Труды СПИИРАН. 2010. Вып. 1 (12). С. 119–133.
5. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик-собственников // Труды СПИИРАН. 2010. Вып. 3 (14) С. 150–169.
6. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи клик владений // Труды СПИИРАН. 2010. Вып. 2 (13). С. 119–133.
7. Фильченков А.А. Алгоритм построения множества минимальных графов смежности при помощи клик-собственников владений // Труды СПИИРАН. 2010. Вып. 4 (15). С. 193–212.
8. Фильченков А.А. Субоптимальная звездчатая структура алгебраической байесовской сети // Информационно-управляющие системы. 2013. Вып. 2. С. 13–17.

9. *Фильченков А.А., Мусина В.Ф., Тулупьев А.Л.* Алгоритм рандомизированного синтеза минимального графа смежности // Тр. СПИИРАН. 2013. Вып. 2(25). С. 221–234.
 10. *Фильченков А.А., Тулупьев А.Л.* Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. Вып. 2. С. 69–78.
 11. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Компаративный анализ клик минимальных графов смежности алгебраических байесовских сетей // Труды СПИИРАН. 2010. Вып. 2(13). С. 87–105.
 12. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Ребра графов смежности в контексте компаративного анализа клик минимальных графов смежности алгебраических байесовских сетей // Труды СПИИРАН. 2010. Вып. 3(14). С. 132–149.
 13. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. Вып. 11. С. 104–127.
 14. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Мощность множества минимальных графов смежности // Труды СПИИРАН. 2010. Вып. 15. С. 136–161.
 15. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Особенности анализа вторичной структуры алгебраической байесовской сети // Труды СПИИРАН. 2010. Вып. 1(12). С. 97–118.
 16. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Структурный анализ клик минимальных графов смежности // Вестн. Тверск. гос. ун-та. Сер.: Прикладная математика. 2011. №20. С. 139–151.
-

Содержание

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	5
Овчинников А.А. Восстановление памяти виртуальной машины из расширенного образа памяти базовой системы на платформе Windows	7
П.Ю. Свицерский. Общий подход к восстановлению адресного пространства процесса из образа памяти ОС Android	12
Карбушев Д.О., Волошин С.В. Корпоративная система организации информации и событий	17
Агапова Татьяна. Обзор современных систем управления бизнес- процессами	20
Птахина А.И. Разработка метамоделирования «на лету» в системе QReal .	28
Белокуров Д.Н., Бумаков Н.В., Захаров В.А., Чижова Н.А. Разработка визуального конструктора мобильных приложений.....	36
Мальгин А.О., Баклановский М.В. Разработка и реализация X-Client для ОС Windows	41
ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА.....	45
Буянов С.В. Достаточные условия сходимости несинхронизируемых итерационных методов	47
Заболотский Е.Д. Разработка и реализация быстрых алгоритмов классификации текстов опубликованных online	48
Филин А.Ю., Соловьев И.П. Web-приложение для работы с изображениями биомедицинского характера	53
М.Л. Симуни. Функциональное программирование с использованием обобщенных регулярных выражений.....	55
А.А. Овсянников. Реализация пакета программ для улучшения качества видеоинформации средствами взаимодополняемых источников.....	59
Петухова Н.Д. Обратный метод Маслова и муравьиная тактика решения задач искусственного интеллекта	64
Гошев В.А. Схема универсальной функции для языка РЕФАЛ-5	70
М.А. Герасимов. NP-полная задача о разбиении множества на K подмножеств и субэкспоненциальные функции	74
Шульгин Д.С. Сравнительный анализ серверов приложений	78
М.А. Герасимов, А.А. Носовец. Классификация древнетибетских текстов с помощью методов спектрального анализа	84
ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РАЗРАБОТКИ ПРОГРАММ И ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ	99
Вербицкая Е.А., Григорьев С.В. Абстрактный лексический анализ	101
Пилипенко А.В. Оптимизация представления байт-кода JVM для встраиваемых систем	104

Григорьев Д.А., Григорьева А.В., Сафонов В.О. Бесшовная интеграция аспектов в облачные приложения на примере библиотеки Enterprise Library Integration Pack for Windows Azure и Aspect.NET.....	107
Сиунов А.Э., Сафонов В.О. К вопросу о разработке и реализации инструмента выделения аспектов из не аспектно-ориентированных программ.....	112
РАНДОМИЗАЦИЯ, УСРЕДНЕНИЕ И МУЛЬТИАГЕНТНЫЕ ТЕХНОЛОГИИ В DATA MINING И УПРАВЛЕНИИ	
С.А. Землянская. Распознавание языка жестов на видео потоке	115
Коноплев Ю.М., Сысоев С.С. Рассмотрение подходов к реализации симулятора квантовых вычислений на классической архитектуре.....	122
А.В. Новиков, Е.Н. Бендерская. Применение процессов синхронизации в осцилляторных сетях для решения задач кластеризации	129
Петров Н.С. Распознавание недеформируемых трехмерных объектов на изображениях по контурам	139
Кривоконь Д.С., Вахитов А.Т. Восстановление глубины точки при наблюдении с одной камеры с помощью рандомизации ее положения.....	145
Белова Е.В. Система определения заданных элементов на аэрофотоснимках.....	153
А.И. Кукушкин, Е.Н. Бендерская. Моделирование стохастических процессов в вероятностной машине Тьюринга.....	160
Улитин. А.А., Вахитов. А.Т. Сравнение подходов обучения на базе словаря и MAP к проблеме повышения разрешения на примере изображений автомобильных номеров	166
ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ И ВЭЙВЛЕТНАЯ ОБРАБОТКА ЧИСЛОВЫХ ПОТОКОВ.....	
Евдокимова Т.О. О курсе лекций «Архитектура параллельных систем».....	175
В.О. Дронь. О вейвлетном гребне на нерегулярной сетке	179
Ю.К. Демьянович. Пространства минимальных сплайнов и калибровочные соотношения.....	185
Чистяков П.П. Первичный криптоанализ и распараллеливание нового алгоритма шифрования.....	190
Рудаков С.А. Особенности реализации нового варианта вэйвлетного разложения.....	201
Романовский Л.М. О локальном укрупнении триангуляции.....	207
В.В. Петров. Распараллеливание сплайн-вейвлетных разложений.....	211
Парахин М.В. Триангуляция области, её локальное укрупнение и использование.....	219
Т.О. Мухамедов. Вейвлетные преобразования	226
Мирошниченко И.Д. Структура сплайн-вейвлетных разложений.....	230

Зубарева М.В. О сравнение сплайн-вэйвлетных разложений с некоторыми вэйвлетными разложениями И. Добеши	237
Д.М. Гусев. Реализация сжатия изображений с использованием сплайн-вэйвлетного разложения и технологий OpenCL и CUDA	243
Д. Гершун. Определение и реализация вычислений для структур JSON	249
Бухтин Е.В. Структура разреженного вэйвлетного гребня	252
Ю.К. Демьянович, О.Н. Иванцова. Оценки аппроксимации неполиномиальными сплайнами второго порядка	256
ТЕОРИЯ И ПРАКТИКА КОДИРОВАНИЯ ИНФОРМАЦИИ	259
Морозов Р.А. Декодирование полярных кодов с помощью алгоритма Думера-Шабунова	261
Алексеев М.О., Егания А.В. Обобщение надежных кодов	263
Егоров Н.Д., Соколовский Р.А. Схема сжатия цветных изображений с применением адаптивного двоичного арифметического кодирования	269
Филиппов Б.С., Егоров Н.Д. О задаче выбора параметров преобразования на основе полиномов Чебышева с целью повышения эффективности сжатия изображений	275
Новиков Д.В. Об эффективности применения алгоритма SURF в задаче идентификации лиц	281
Веселов А.И., Ястребов В.А. Оценка метода маскирования визуальных искажений на стороне декодера для уменьшения влияния потерь при передаче видеоданных по сети	287
Гранкин М.А., Бакин Е.А. Методика расчета предельных характеристик системы связи, работающей на принципах нелинейного отражения	293
РАСПАРАЛЛЕЛИВАНИЕ В OPEN MP И СПЛАЙНОВЫЕ АППРОКСИМАЦИИ	299
Винник М.П., Бурова И.Г. Распараллеливание в OpenMP среднеквадратических приближений неполиномиальными сплайнами минимального дефекта	301
Н.И. Лещенко О распараллеливании решения интегральных уравнений линейной вязкоупругости	318
Абдурахимова А.С., Бурова И.Г. Распараллеливание решения задачи Коши, методами с применением интегро-дифференциальных сплайнов	326
Бурова И.Г., Родникова О.В. О распараллеливании решения задачи Эрмита-Биркгофа интегро-дифференциальными сплайнами	331
Бурова И.Г., Полуянов С.В. О построении среднеквадратического приближения интегро-дифференциальными сплайнами пятого порядка первой высоты	337

НЕЛИНЕЙНЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ: МОДЕЛИ, УПРАВЛЕНИЕ, ЧИСЛЕННЫЙ АНАЛИЗ.....	341
Думшева Т.Д., Кандоба И.Н., Костоусова Е.К., Ложников А.Б., Починский В.И. О задаче оптимального выведения ракеты-носителя на околоземную орбиту	343
Тарасьев А.М., Усова А.А. Алгоритмы построения оптимальных траекторий по аппроксимирующим управлениям в задачах на бесконечном промежутке времени	350
Ядгар З. Кандиев, Олег В. Зацепин, Геннадий Н. Малышкин, Светлана Г. Спирина, Тамара В. Вахонина, Нина А. Воронина, Елена А. Кашаева, Лариса В. Лобанова, Геннадий В. Лукин, Алексей А. Малахов, Дмитрий Г. Модестов, Рим Ф. Мухамадиев, Виктор Г. Орлов, Сергей И. Самарин, Елена В. Серова, Кирилл Е. Хатунцев. Программа ПРИЗМА для расчетов задач переноса частиц методом Монте-Карло	358
Козьмин И.В. Задача вывода ракеты-носителя на орбиту в условиях случайных возмущений параметров	373
Гомоюнов М.И. О численном решении одной дифференциальной игры с запаздыванием	378
Грибанова Е.И. Численное моделирование задачи реконструкции гранич- ных управлений в динамических системах гиперболического типа	382
Соколов Л.В., Лыков В. А, Чижков М.Н., Шушлебин А.Н. Двумерные расчеты сжатия и горения микромишеней для ЛТС с непрямым воздействием.....	389
МЕТОДЫ ХРАНЕНИЯ, ПОИСКА И АНАЛИЗА ИНФОРМАЦИИ.....	397
Долматова О.А. Адаптивный приближенный алгоритм соединения	399
Салищев С.И., Шеин Р.Е. Modeling of Hybrid CDN-P2P for Full HD video downloading with realistic demand distribution	405
Дзендзик Д.А. Извлечение событий на основе автоматизированного построения линейных шаблонов	413
КИБЕРНЕТИКА И РОБОТОТЕХНИКА	419
Корчемкин Д.А., Салищев С.И. Построение карт смещений по неректифицированным стереоизображениям	421
ПРЕПОДАВАНИЕ КОМПЬЮТЕРНЫХ ДИСЦИПЛИН ДЛЯ МАТЕМАТИКОВ.....	427
Луцив Д.В., Кириленко Я.А., Полозов В.С. Среды визуального моделирования для преподавания	429
ИЕРАРХИЧЕСКИЕ ЯЗЫКИ, ФАБРИКИ ВЫЧИСЛЕНИЙ, ОБЛАЧНЫЕ ТЕХНОЛОГИИ.....	439
Манакова И.П. Менеджер управления мультимедиа-сетью	441

МУЛЬТИАГЕНТНЫЕ ТЕХНОЛОГИИ И ИХ ПРИЛОЖЕНИЯ	
В ИНФОРМАТИКЕ	449
Титов В.В., Шардыко И.В., Даляев И.Ю. Методика обработки сигнала однокомпонентного датчика момента для применения в силовомоментном управлении электромеханическим шарниром	451
Бакурадзе А.М. Проблемы и методы много-адресной маршрутизации в глобальных телекоммуникационных системах нового поколения.....	457
З.В. Нагоев, В.А. Денисенко. Разработка библиотеки классов и алгоритмов компьютерной системы для задачи синтеза интеллектуального поведения агента на основе рекурсивной когнитивной архитектуры	466
О.В. Нагоева. Разработка принципов поддержки экономических интересов пользователей в мультиагентной интеллектуальной среде обитания	471
Никитин К.В. Методика синтеза рекуррентных нейронных сетей для распознавания динамических образов	481
А.В. Тимофеев Мульти-агентное управление и интеллектуальный анализ информационных потоков	496
Данилов А.А., А.В. Тимофеев. Информационные и мульти-агентные технологии управления квадрокоптерами для мониторинга морской фауны.....	494
А.В. Тимофеев Интеллектуализация интегрированных систем навигации, управления движением и функциональной диагностики роботов.....	498
АВТОМАТНОЕ ПРОГРАММИРОВАНИЕ, МАШИННОЕ ОБУЧЕНИЕ И БИОИНФОРМАТИКА	
.....	509
И.П. Бужинский, В.И. Ульянов. Построение автоматов управления объектами со сложным поведением по тестам с учетом непрерывных воздействий.....	511
Чивилихин Д.С. Метод построения конечных автоматов на основе муравьиного алгоритма	517
Чивилихин Д.С. Эволюционные стратегии с адаптивным параметром на основе свойств ландшафта функции приспособленности	525
Ульянцев В.И., А.А. Шальто. Построение управляющих конечных автоматов по сценариям работы на основе решения задачи удовлетворения ограничений	532
Соколов А.А., Буздалов М.В. Разработка алгоритмов для упорядочивания структур белков	538
Кобер Д.А. Гибридный алгоритм решения задачи составления расписания выполнения заданий, связанных порядком предшествования, в многопроцессорной среде	542
Буздалова А.С., Буздалов М.В. Использование вспомогательных функций приспособленности для тестирования решений олимпиадных задач по программированию	548

Е.В. Панченко, В.И. Ульянов, Ф.Н. Царев. Применение методов решения задачи о выполнимости квантифицированной булевой функции для построения управляющих конечных автоматов по сценариям работы и темпоральным свойствам.....	556
Славнейшев Ф.В. Алгоритм выравнивания последовательностей ДНК для модели MapReduce	560
Буздалов М.В., Буздалова А.С. Оценка времени работы эволюционного алгоритма RMHC под управлением алгоритма Q-learning на задаче OneMax с мешающим критерием оптимизации	565
Якорев В.О., Буздалов М.В. Генерация тестов для олимпиадных задач по программированию с помощью многокритериальных эволюционных алгоритмов	571
Якорев В.О., Шалыто А.А. Синтез систем управления балансирующим роботом на основе методов искусственного интеллекта	574
Александров А.В., Казаков С.В., Сергушичев А.А., Царев Ф.Н. Метод сборки контигов геномных последовательностей из парных чтений с ошибками вставки и удаления на основе совместного применения графов де Брёйна и графов перекрытий.....	578
СИНТЕЗ ЭЛЕМЕНТОВ КОМПЬЮТЕРНОЙ АРХИТЕКТУРЫ.....	
Кудряшова Е.В., Кузнецова О.А., Селеджи С.М. Нелинейный анализ цифровых систем фазовой автоподстройки частоты.....	585
Арсеньев Д.Г., Кузнецов Н.В., Леонов Г.А., Оносовский В.В., Шкодырев В.П. Развитие методов и технологий в области информационного менеджмента	592
В.В. Коновалов. Информационная система для компактного мобильного устройства.....	594
Н. Кузнецов, Г. Леонов, В. Оносовский, С. Селеджи, Евгений Филиппов. Инновационные подходы к подготовке современных высококвалифицированных специалистов в области ИТ	601
МАТЕМАТИЧЕСКИЕ МЕТОДЫ И АЛГОРИТМЫ В СИСТЕМАХ ХРАНЕНИЯ ДАННЫХ ВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ.....	
Д.А. Драган Исследование и оптимизация алгоритмов устранения избыточности в цифровом представлении генома человека.....	621
Белокопытова Е.А., Фёдоров А.А., Александрова О.В. Менеджер распределения доступа к ресурсам	623
Корпоративная система организации информации и событий.....	626
Зайберт В.С., Платонов С.М. Разработка модуля вычисления синдромов и восстановления утраченных дисков в RAID-массиве с использованием арифметики поля $GF(2^8)$	627
Демьяненко И.И., Савельев Н.Г., Платонов С.М. Разработка модуля вычисления синдромов и восстановления утраченных дисков в RAID-массиве с использованием арифметики поля $GF(2^{16})$	633

Быкова Ю.С., Веселков И.Д., Платонов С.М. Разработка модуля вычисления синдромов и восстановления утраченных дисков в RAID-массиве с использованием арифметики поля $GF(2^{256})$	638
Щербаков А.В., Платонов С.М. Сравнение алгоритмов обращения элементов $GF(2^N)$	643
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ.....	
Басманов Д.Ю. Оценка применимости моделей информационной безопасности для распределённых вычислений к задаче управления робототехническими системами.....	649
Ендовский А.С., И.А. Зикратов. Методы расчета риска информационной безопасности	653
Калинин Д.А. Способы реализации Side Channel атак и методы противодействия им.....	658
Васильева Н.А. Обеспечение защиты информации дополнительных услуг операторов сотовой связи	660
Коровкин М.Г., Левина А.Б. Side-Channel атаки. Взлом COMP128.....	666
Красильников А.Е., Комаров И.И. Оценка тональности текстовой информации, анализ и сравнение эффективности алгоритмов	671
Лапшин С.В. Метод повышения точности частеречной классификации слов в морфологических анализаторах DLP-систем	674
Кустов В.А., Комаров И.И. Задача оптимизации защищенности каналов управления робототехническими системами	678
Мальшкін Г.А., Каторин Ю.Ф. Проблемы аутентификации с использованием биометрических характеристик	680
Соломатин А.Ю. Метод идентификации объектов в системах видеонаблюдения на основе математического аппарата иммунокомпьютинга.....	684
Сухопаров М.Е. Идентификация авторов коротких сообщений порталов сети Интернет с использованием методов математической лингвистики	686
Юрасов Д.С. Идентификация пользователей на основе их поведения в сети Интернет.....	688
Юрьева Р.А., Левина А.Б. Методы и средства клонирования RFID-меток	693
Пряхин В.Е., Зикратов И.А. Разделимость запросов при нагрузочной атаке веб-серверов.....	695
ВЕРОЯТНОСТНЫЕ ГРАФИЧЕСКИЕ МОДЕЛИ, НЕЧЕТКИЕ СИСТЕМЫ И МЯГКИЕ ВЫЧИСЛЕНИЯ	
Азаров А.А. Модели комплекса «информационная система — персонал — критические документы» при оценке защищенности пользователей информационных систем	701

Торопова А.В. Разработка комплекса программных средств управления депозитарием электронных научных изданий	709
Лепилина А.П. Автоматизация деятельности колл-центра: комплекс хранимых процедур	713
Хамаева Э.А. Программа для анализа защищенности от социоинженерных атак критичных документов с учетом структуры связей между пользователями информационной системы	717
Михайлов Н.Ю. Визуализация контролируемых зон при анализе степени защищенности критичных документов от социоинженерных атак с учетом структуры связей между пользователями информационной системы	722
Суворова А.В. Грануляция неточных данных в синтезе байесовской сети доверия при исследовании частично наблюдаемых реализаций пуассоновского процесса.....	729
Пашенко А.Е., Суворова А.В., Тулупьев А.Л., Тулупьева Т.В. Рандомизация неопределенности исходных данных при анализе угрозаобразующего поведения	733
Абрамов М.А., Ковалев В.С. Прототип Интернет-портала редакции электронного научного издания.....	738
Алексеев А.М. Методы построения и обработки скрытых марковских моделей, представленных в виде алгебраических байесовских сетей .	744
Фроленков К.В. Задача выбора распределения, отражающего вероятностную семантику алгебраической байесовской сети	750
Фильченков А.А. Синтез вторичной структуры алгебраической байесовской сети: компромисс между сложностью и ацикличностью	754
Соловцов Н.Б. Алгоритм расчета оценки защищенности пользователей информационной системы за счет элиминации маловероятных траекторий социо-инженерных атак	760
Вяткин А.В. Элиминация и идентификация циклов во вторичной структуре алгебраической байесовской сети на основе методов теории древовидной декомпозиции	766
Фильченков А.А., Мусина В.Ф. Равновероятностный синтез реализации случайного минимального графа смежности	770
Сироткин А.В., Тулупьев А.Л., Фильченков А.А., Фроленков К.В. Оценки сложности алгоритмов синтеза минимальных по числу ребер графов смежности	776

Научное издание

СПИСОК-2013

МАТЕРИАЛЫ
ВСЕРОССИЙСКОЙ НАУЧНОЙ КОНФЕРЕНЦИИ
ПО ПРОБЛЕМАМ ИНФОРМАТИКИ

*23–26 апр. 2013 г.,
Санкт-Петербург*

Редактор:
А. Н. Терехов
Компьютерная верстка:
М. Н. Клочков

**Издательство
DDO**

Подписано к печати 05.11.2013. Формат 60×84¹/₁₆. Бумага офсетная.
Гарнитура Таймс. Печать цифровая. Усл. печ. л. 48,2. Тираж 300 экз.
Заказ 5905

Отпечатано в Отделе оперативной полиграфии
химического факультета СПбГУ

198504, Санкт-Петербург, Старый Петергоф, Университетский пр., 26
Тел.: (812) 428-4043, 428-6919