

Министерство образования и науки Российской Федерации  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
(СПбГУ ИТМО)

УДК 004.4'242  
№ госрегистрации  
Инв. №

УТВЕРЖДАЮ  
Руководитель работы  
докт. техн. наук., профессор  
А. А. Шалыто  
31 мая 2007 г.

«\_\_\_» \_\_\_\_\_ 2007 г.

ОТЧЕТ О ПАТЕНТНЫХ ИССЛЕДОВАНИЯХ  
№ 2007.08.31-1 от 31 августа 2007 г.

по теме

ТЕХНОЛОГИЯ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ГЕНЕРАЦИИ АВТОМАТОВ  
УПРАВЛЕНИЯ СИСТЕМАМИ СО СЛОЖНЫМ ПОВЕДЕНИЕМ

Этап 1. Выбор направления исследований и базовых компонентов

Шифр 2007-4-1.4-18-01-033.

Декан факультета «Информационных  
технологий и программирования»  
докт. техн. наук, профессор

\_\_\_\_\_

подпись

В. Г. Парфенов

Начальник отдела  
интеллектуальной собственности и  
научно-технической информации

\_\_\_\_\_

подпись

Л. Н. Казар

Санкт-Петербург  
2007

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

## **СПИСОК ИСПОЛНИТЕЛЕЙ**

Руководитель темы  
Заведующий кафедрой  
«Технологии программирования»,  
докт. техн. наук, профессор

А.А. Шалыто

Исполнитель отчета  
Магистрант

В.Н. Точилин

## СОДЕРЖАНИЕ

СПИСОК ИСПОЛНИТЕЛЕЙ .....	2
СОДЕРЖАНИЕ .....	3
ОБЩИЕ ДАННЫЕ ОБ ОБЪЕКТЕ ИССЛЕДОВАНИЙ .....	4
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ, ТЕРМИНОВ .....	5
1. ИССЛЕДОВАНИЯ ПАТЕНТОВ .....	6
2. ИССЛЕДОВАНИЯ НЕПАТЕНТНЫХ ИСТОЧНИКОВ .....	12
2.1. ОБЩИЙ ОБЗОР ПУБЛИКАЦИЙ .....	12
2.2. ОБЗОР ИЗБРАННЫХ РАБОТ .....	24
2.2.1. <i>PADO</i> : a New Learning Architecture for Object Recognition .....	24
2.2.2. Genetic Programming – An Introduction. On the Automatic Evolution of Computer Programs and its Application .....	25
2.2.3. Empirical Analysis of Different Levels of Meta-Evolution .....	25
2.2.4. Meta-Evolution in Graph GP .....	25
2.2.5. Internal Reinforcement in a Connectionist Genetic Programming Approach .....	26
2.2.6. <i>SYSGP</i> – A C++ library of different GP variants .....	26
2.2.7. Evolving Automatic Target Detection Algorithms that Logically Combine Decision Spaces .....	27
2.2.8. Linear-Graph GP. A new GP Structure .....	27
2.2.9. Evolution of Graph-like Programs with Parallel Distributed Genetic Programming .....	28
2.2.10. Evolving Strategies for Global Optimization. A Finite State Machine Approach .....	28
2.2.11. Simulated evolution of distributed FSA behaviour-based arbitration .....	28
2.2.12. Evolving automats for distributed behavior arbitration .....	29
2.2.13. Comparing Finite-State Automata Representation with GP-trees .....	29
2.2.14. Swarm Rule-Base Development using Genetic Programming Techniques .....	30
2.2.15. Genetic Simulation for Finite State Machine Identification .....	31
2.3. РЕЗУЛЬТАТЫ НЕПАТЕНТНЫХ ИССЛЕДОВАНИЙ .....	31
3. ОТЛИЧИЯ ПРОВОДИМОГО ИССЛЕДОВАНИЯ .....	32
ЗАКЛЮЧЕНИЕ .....	33
ЛИТЕРАТУРА И ИСТОЧНИКИ .....	34
ПРИЛОЖЕНИЕ А .....	37
ПРИЛОЖЕНИЕ Б .....	38
ПРИЛОЖЕНИЕ В .....	39

## **ОБЩИЕ ДАННЫЕ ОБ ОБЪЕКТЕ ИССЛЕДОВАНИЙ**

Патентный поиск проводился с целью определения патентоспособности планируемых результатов научно-исследовательской работы по лоту «Разработки в области языков программирования и моделирования программного обеспечения, технологий и инструментальных средств проектирования программ» – тема «Технология генетического программирования для генерации автоматов управления системами со сложным поведением» (шифр «2007-4-1.4-18-01-033»), выполняемой в рамках Федеральной целевой научно-технической программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 годы» по государственному контракту № 02.514.11.4044, заключенному между Федеральным агентством по науке и инновациям и Государственным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на основании решения Конкурсной комиссии Роснауки (протокол от 28.04.2007 г. № 14), а также для получения сведений об охранных и иных документах, которые могут препятствовать применению результатов данной НИР в Российской Федерации, и условиях использования таких документов.

В соответствии с задачей исследования, разрабатываемая технология генерации автоматов должна удовлетворять следующим требованиям:

- возможность создания автоматов управления системами со сложным поведением ;
- построение детерминированных конечных автоматов по заданной функции пригодности (при этом явное задание эталонного поведения автомата не предполагается);
- использование генетических алгоритмов.

Патентный поиск проводился в соответствии с ГОСТ Р. 15.011-96 «Система разработки и постановки продукции на производство. Патентные исследования» [1]. Проверка патентоспособности проводимой научно-исследовательской работы осуществлялась на основе поиска патентных и других открытых документов, описывающих решения, максимально полно удовлетворяющие задаче исследования. Поиск патентной информации проводился в патентных базах данных Федеральной службы по интеллектуальной собственности, патентам и товарным знакам Российской Федерации (Роспатент, [www.fips.ru](http://www.fips.ru)), Бюро по патентам и товарным знакам США (USPTO, [www.uspto.gov](http://www.uspto.gov)) и Европейского патентного бюро (EPO, [ep.espacenet.com](http://ep.espacenet.com)).

Патентный поиск проводился с 1 июня 2007 г. по 31 августа 2007 г.

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

### **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ, ТЕРМИНОВ**

*НИИР* — научно-исследовательская работа.

*FSM* — Конечная машина состояний (Finite State Machine).

*FSA* — Конечный автомат (Finite State Automaton).

*PADO* — Параллельный поиск и согласование алгоритмов (Parallel Algorithm Discovery and Orchestration).

*PDGP* — Параллельное распределенное генетическое программирование (Parallel Distributed Genetic Programming).

## 1. ИССЛЕДОВАНИЯ ПАТЕНТОВ

Тема исследования ограничивается применением генетического программирования для генерации автоматов. Однако, для повышения полноты результатов поиска, рассматриваемая тематическая область была расширена на следующие направления:

- автоматическое построение программ;
- имитация естественной эволюции;
- генетические алгоритмы;
- адаптивные системы;
- применение автоматных моделей вычислений;
- построение управляющих систем.

В случаях, когда в рамках некоторого направления была возможность выделить блоки меньшего объема, блоки, заведомо не соответствующие направлению исследования, из рассмотрения исключались. Таким образом, были выбраны элементы патентных классификаций для ограничения области поиска.

В рамках патентного поиска производился анализ патентов, принадлежащих следующим категориям международной патентной классификации (рассматривались патенты, принадлежащие самому низкому указанному уровню):

<i>G</i>	Физика (Physics)
<i>G05</i>	Управление; регулирование (Controlling; regulating)
<i>G05B</i>	Регулирующие и управляющие системы общего назначения; функциональные элементы таких систем; устройства для контроля или испытания таких систем или элементов (Control or regulating systems in general; functional elements of such systems; monitoring or testing arrangements for such systems or elements)
<i>G05B 13</i>	Самонастраивающиеся системы управления (Adaptive control systems, i.e. systems automatically adjusting themselves to have a performance which is optimum according to some pre-assigned criterion)
<i>G06</i>	Вычисление; счет (Computing; calculating; counting)
<i>G06F</i>	Обработка цифровых данных с помощью электрических устройств (Electrical digital data processing)
<i>G06F 15</i>	Цифровые компьютеры вообще (Digital computers in general)
<i>G06F 15/18</i>	с модификацией программ, осуществляемой самой машиной в соответствии с опытом, накопленным машиной за время полного цикла работы; самообучающиеся машины (in which a programme is changed according to experience gained by the computer itself during a complete run; Learning machines)
<i>G06N</i>	Компьютерные системы, основанные на специфических вычислительных моделях (Computer systems based on specific computational models)

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

<i>G06N 3</i>	Компьютерные системы, основанные на биологических моделях (Computer systems based on biological models)
<i>G06N 3/00</i>	Компьютерные системы, основанные на биологических моделях (Computer systems based on biological models)
<i>G06N 3/12</i>	Компьютерные системы, использующие генетические модели (Using genetic models)
<i>G06N 7</i>	Компьютерные системы, основанные на специфических математических моделях (Computer systems based on specific mathematical models)

Кроме того, рассматривались следующие категории патентной классификации США (перевод на русский язык исполнителя отчета):

706	Обработка данных: искусственный интеллект (Data processing: artificial intelligence)
706/13	Машинное обучение. Системы генетических алгоритмов и генетического программирования (Machine learning. Genetic algorithm and genetic programming system)
706/14	Адаптивные системы (Adaptive system)
716	Обработка данных: разработка и анализ электрических схем и масок полупроводников (Data processing: design and analysis of circuit or semiconductor mask)
716/18	Разработка электронных схем. Синтезатор логических схем. (Circuit design. Logical circuit synthesizer)

Также производился поиск без ограничения разделов классификаций по следующим ключевым словам: *FSM, FSA, automaton, automata, transducer, finite state, genetic, inference, evolve, evolving, evolution*.

При этом многие из относительно близких к рассматриваемой теме патентов были отнесены сразу к нескольким разделам классификаций, по которым производился поиск. Это свидетельствует о достаточно полном покрытии области исследования областью поиска. Так как патенты, полностью удовлетворяющие поставленным перед исследованием требованиям, найдены не были, в качестве более подробного результата поиска рассмотрим наиболее близкие к теме исследования патенты:

1. Патент JP 10-307806 от 17.11.1998 “*Autonomous evolution type character action generation method and device*”.

Полный текст патента доступен только на японском языке. Однако выводы о его содержании можно сделать из краткого англоязычного описания, приведенного в патентном архиве. Решаемая задача – создание персонажа компьютерной игры, способного длительное время представлять интерес для пользователя в качестве объекта взаимодействия. Персонаж описывается конечным автоматом, который эволюционирует в процессе игры, на основе взаимодействия с пользователем. Из краткого описания можно сделать вывод о том, что эволюция автомата осуществляется через применение генетического алгоритма. Можно выделить следующие отличия решаемой патентом задачи от задачи проверяемого на патентоспособность исследования:

Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

- узкая специализация – предложено специальное применение генетического построения автоматов для управления персонажем компьютерной игры, в то время как проверяемое исследование не ограничивается даже задачами управления вообще;
- отсутствие требования оптимальности результата: для персонажа компьютерной игры важно постепенное развитие вместе с развитием пользователя. Задача получения как можно более сильного персонажа не ставится;
- автономная эволюция – для оптимизации автомата используется только информация, доступная управляемому им персонажу.

2. Патент JP 10-307804 от 17.11.1998 “*Method and device for generating autonomous progress type character action*” почти идентичен предыдущему.

3. Патент US 5615124 от 25.03.1997 “*Autonomous evolution type hardware design system*”.

Предложен способ подстройки электронной схемы под решаемую задачу. Описание схемы на языке *HDL (hardware description language)* модифицируется генетическим алгоритмом для наилучшего соответствия результата работы схемы требуемому. Заявлена применимость изобретения не только для аппаратного обеспечения, но также, в частности, для искусственных существ, описываемых автоматом. Язык *HDL* действительно позволяет описывать автоматы, и поэтому описываемый в данном патенте подход применим для их построения. Однако, это низкоуровневые автоматы, которые имеют отношение к генетическому программированию с использованием графовых, и даже линейных, моделей, чем к предмету проверяемого исследования.

4. Патент WO 98/02825 от 22.01.1998 “*Computer implemented machine learning method and system*”.

Описывает реализацию системы линейного генетического программирования. Линейное генетическое программирование, заключающееся в применении генетических алгоритмов для синтеза непосредственно исполняемых процессором программ, хотя и является относительно новым направлением генетического программирования, существовало задолго до появления данного патента. Патент же описывает естественную реализацию системы линейного генетического программирования. Линейное генетическое программирование отличается от рассматриваемого в проводимом исследовании еще сильнее, чем генетическое программирование, оперирующее низкоуровневыми графовыми моделями.

5. Патент US 5943659 от 24.08.1999 “*Deterministic encoding of fuzzy finite state automata in continuous recurrent neural networks*”.

На основе представления детерминированного конечного автомата в виде рекуррентной нейронной сети предлагается кодирование в виде нейронной сети нечеткого автомата, распознающего некоторый нечеткий язык с любой заданной точностью. Широко известна взаимная эквивалентность конечного автомата и рекуррентной нейронной сети, а также способы применения генетических алгоритмов для оптимизации таких сетей. В проводимом исследовании не применяется такой подход, не рассматриваются нечеткие автоматы, а распознавание языка не является типичной задачей.

6. Патенты EP 1296281 от 26.03.2003 и US 6839698 от 04.01.2005 “*Fuzzy genetic learning automata classifier*”.

Описывает построение нечеткого автомата, наилучшим образом разделяющего на классы входные последовательности. Классификация последовательностей или распознавание языка – одна из наиболее известных областей применения конечных автоматов. Однако нечеткие автоматы и зада-



Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

чи классификации не характерны для темы данного патентного поиска. Кроме того, в патенте не рассматривается способ модификации автомата.

7. Патенты US 5327544 от 05.07.1994 “*Method and apparatus for designing gateways for computer networks*” и US 7136947 от 14.11.2006 “*System and method for automatically synthesizing interfaces between incompatible protocols*”.

Описывается автоматическое построение преобразователя, позволяющего передавать данные между узлами с различными протоколами взаимодействия. Протоколы и преобразователь представляются в виде конечных автоматов. Эти изобретения можно отнести к применению автоматов в более сложной области, чем распознавание языков – при преобразованиях (*transducers*). К этому же классу относятся, например, компиляторы, но не объект проверяемого исследования.

8. В патенте JP 2004/004521 от 08.01.2004 “*System and method for generating weighted finite state automaton indicating grammar*” построение автомата-преобразователя используется в качестве промежуточного шага для построения автомата, описывающего грамматику.

9. Патент WO 2004/021181 “*Method and apparatus for efficient implementation and evaluation of state machines and programmable finite state automata*”.

Описывается способ реализации автоматов, но не способ их оптимизации.

10, Патенты JP 10-063634 от 06.03.1998 и US 5761386 от 02.06.1998 “*Method and device for time sequential prediction/ classification*”.

Описывают способ предсказания значений в последовательности на примере дневных обменных курсов валют. После предварительной обработки последовательность используется для обучения рекуррентной нейронной сети, из которой впоследствии выводится автомат, представляющий грамматику последовательности, или модель поведения рынка. Следовательно, подобно предыдущим патентам, автомат используется для представления языка и модифицируется в форме нейронной сети.

11. Патенты EP 0941504 от 15.09.1999 “*Process for parametering a fuzzy automaton that compares a measurement system to a pattern signal*” и US 6345206 от 05.02.2002 “*Method for configuring a fuzzy automatic-control device which is used for comparing a measurement signal with a pattern signal*”.

Описывают метод автоматического конфигурирования нечеткого управляющего автомата или нечеткой управляющей системы (в патенте США). В случае управляющей системы, даже если она не названа автоматом, можно построить эквивалентный нечеткий автомат, который, однако, не является предметом проводимого исследования.

12. Патент EP 1202141 от 02.05.2002 “*Method of reducing finite controlling automata and corresponding computer-readable medium*”.

Часто несколько небольших управляющих автоматов объединяют в один большой. Большой автомат сложно преобразовывать и часто не представляется возможным изобразить графически. Для применения техник верификации большой автомат сокращают. Для этого можно применять описанный в патенте способ. Из описания автомата исключаются избыточные переходы, недостижимые состояния и переходы внутри одного состояния без действий. Таким образом, патент охватывает только упрощение заданного автомата, что может быть применено после исследуемого построения автомата методами генетического программирования, но не заменяет его.

13. Патент US 4829450 от 09.05.1989 “*Reasoning machine*”.

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

Решается задача создания компьютерной программы, способной обучаться на опыте, взаимодействовать с пользователем, учитывая не только текущие данные, но и историю. Для истории выбрано представление в виде состояния конечного автомата. Состояния представляют действия, а переходы – их взаимосвязи. Новые действия создаются на основе входных данных с интерфейса пользователя, и добавляются в граф. При этом автоматически обнаруживаются повторяющиеся действия, и, таким образом, создаются циклы в графе. Также автоматически создается иерархия автоматов. Кроме создания более удобного интерфейса предлагается использовать накопленные в форме иерархии конечных автоматов знания о действиях пользователя для автономного управления устройствами, такими как, например, роботы. При этом автоматы будут способны использовать обратную связь, адекватно реагировать на внешние события, а не просто воссоздавать заданную последовательность действий. Рассмотренный подход отличается тем, что автомат выводится из эталонного поведения пользователя, в то время как в генетическом программировании задается критерий оптимальности выводимой системы, а не эталонное поведение.

Сводная информация по основным различиям рассмотренных патентов и задач проверяемого на патентоспособность исследования приведена в табл. 1.

Из изложенного следует, что поиск не выявил наличия патентов, препятствующих проведению исследований по теме «Технология генетического программирования для генерации автоматов управления системами со сложным поведением».

Таблица 1. Отличия описываемых в патентах задач от темы исследования

Патент	Нечеткий автомат	Автомат-грамматика	Нет генетического алгоритма	Автомат выводится из нейронной сети	Другое
JP 10-307806					Специальное применение, другой критерий оптимальности
JP 10-307804					Специальное применение, другой критерий оптимальности
US 5615124					Низкоуровневые, аппаратные автоматы
WO 98/02825					Не автомат
US 5943659					
US 6839698					

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
 Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

US 5327544					
US 7136947					
JP 2004/004521					
WO 2004/021181					
JP 10-063634 US 5761386					
EP 0941504 US 6345206					
EP 1202141					
US 4829450					Воспроизведение эталонного пове- дения

## 2. ИССЛЕДОВАНИЯ НЕПАТЕНТНЫХ ИСТОЧНИКОВ

Согласно Патентному закону Российской Федерации [2], уровень техники, в сравнении с которым выявляется новизна изобретения, определяется не только зарегистрированными в России патентами, но также имеющейся во всем мире общедоступной информацией. В связи с этим был произведен анализ публикаций, содержащих упоминания эволюционного построения конечных автоматов или подобных им моделей.

### 2.1. ОБЩИЙ ОБЗОР ПУБЛИКАЦИЙ

Эволюционной оптимизации моделей вычислений в виде конечных автоматов были посвящены многие исследования в различных направлениях *эволюционных вычислений*. Классификация по направлениям, приведенная на рис. 1, сделает их рассмотрение более удобным.

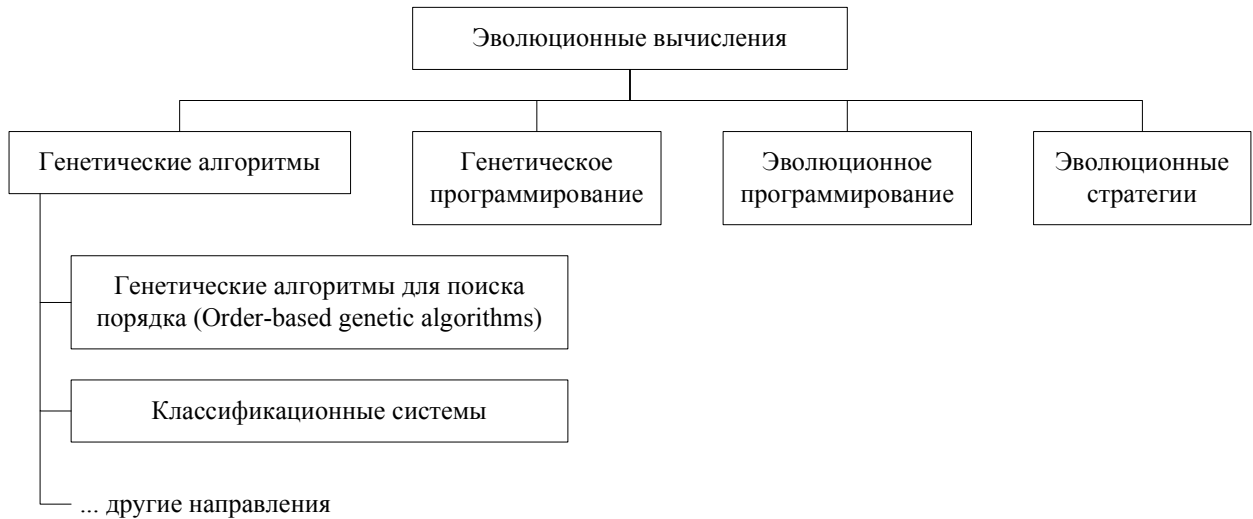


Рис. 1. Направления эволюционных вычислений

*Генетические алгоритмы* рассматривают методы функциональной оптимизации, основанные на модели естественной эволюции. В рамках данного направления существуют ответвления, наиболее известные из которых рассмотрены далее.

*Генетические алгоритмы для поиска порядка* осуществляют эволюционный поиск в пространстве перестановок.

*Классификационные системы* работают преимущественно с пространством порождающих правил, а также конечных автоматов-*распознавателей*, описывающих грамматику некоторого языка. Задача такого автомата состоит в определении принадлежности заданной строки языку. Распознаватель не производит выходных воздействий, а результат определяется состоянием автомата после обработки входной последовательности. В данном направлении можно выделить работы [4, 6, 9, 10, 16, 17, 19, 38].

Более сложная форма конечного автомата – *преобразователь (transducer)* – отображает множество входных строк на множество выходных строк, возможно над другим алфавитом. Примером преобразователя может служить компилятор, а примером распознавателя – синтаксический анализатор, определяющий соответствие кода программы грамматике языка программирования. Эволюционному построению преобразователей посвящена работа [39].

В распознающих и преобразующих автоматах условия переходов описываются в форме сравнения входного символа с заданным. Таким образом, подмножество входных воздействий, удовлетворяющее условию конкретного перехода, состоит из одного заданного входного символа.

*Генетическое программирование* было впервые упомянуто в книге [36]. Первоначально предлагалось представлять программу в виде дерева. В работе [33] рассматривалось представление в виде машинного кода, а в статье [53] впервые оптимизировалась модель в форме графа, который можно интерпретировать как диаграмму переходов конечного автомата. Модель оказалась достаточно удачной, и впоследствии использовалась практически без изменений в ряде работ [8, 31, 32, 52]. В статьях [11, 12, 30] рассматривается совместное использование различных моделей вычислений, включая графы или их модификации. В работах [13, 42] рост автоматов происходит подобно биологическим клеткам, а в работе [48] применяются ациклические графы для формирования эффективно распараллеливаемых программ. В статьях [27, 46, 47] рассматривается автоматическое построение компонент логических контроллеров в виде автоматов, а в работе [45] оценивается эффективность автоматных моделей применительно к различным задачам.

В работе [35] отмечается необходимость расширения генетического программирования для работы со сложными структурами данных и описываются достижения в данном направлении. Эта тема развивается в работах [28, 37, 53], где рассмотрено использование различных заданных структур данных. Отметим, что подход, используемый в настоящей работе, решает эту задачу полностью, позволяя использовать в составе автоматизированного объекта любой объект управления, включая произвольные стандартные структуры данных и их комбинации.

*В эволюционном программировании* сначала изучались методы создания искусственного интеллекта в форме эволюционирующей популяции автоматов, а впоследствии основной его задачей стала оптимизация числовых параметров. Характерной особенностью данного направления можно назвать отсутствие скрещивания, оптимизацию только за счет мутации. В работах этого направления [7, 18, 21, 22, 23, 24, 25, 26, 41, 51] автоматы обучаются предсказывать следующий символ входной последовательности, классифицировать последовательности и копировать поведение системы, заданной частичным набором входных и соответствующих им выходных воздействий. Задачи управления рассматриваются как обучение автомата повторению поведения системы и его анализу с целью выявления входных воздействий, приводящих к требуемому результату.

В работах [5, 14, 43, 49] рассматривается применение автоматов в играх. Авторы работы [20] рассматривают адаптивные методы эволюционного программирования, в работах [15, 29] воссоздание неизвестных систем в виде автоматов применяется для изучения поведения агентов с целью взаимодействия с ними и эффективного тестирования последовательных электрических схем.

*Эволюционные стратегии* позволяют эффективно оптимизировать параметры системы. В рамках этого направления часто рассматриваются аппаратные системы, для которых характерна экспериментальная проверка пригодности (в процессе реальной работы или ее программной эмуляции), что делает количество таких проверок решающим для производительности.

Практически во всех рассмотренных работах в автоматах на переходах обрабатывается только одна входная переменная. Исключения составляют работы [49] (четыре параллельных троичных входа) и [45] (в качестве условия перехода допускается сравнение значений двух регистров). Теоретически любое количество параллельных входов сводится к одному, в качестве воздействий которого выступают комбинации сигналов исходных параллельных входов. Однако размер алфавита полученного таким образом входа растет экспоненциально с увеличением количества исходных параллельных входов. В обоих упомянутых исследованиях параллельные входы не приводят к недопустимо большому алфавиту, но для реальных систем эта проблема крайне актуальна.

В рассмотренных работах автомат на каждом шаге может выполнить не более одного действия из заданного множества. При этом любая комбинация действий, которые может потребоваться

выполнить одновременно, также должна считаться элементарным действием. Для этого требуется априорная информация обо всех возможных комбинациях действий, либо задание вместе с элементарными действиями всех их наборов, что приводит к экспоненциальному росту количества действий. Отметим, что в работе [45] допускаются действия с аргументами, а также параллельно выполняемые автоматы, отвечающие за различные действия, что значительно ослабляет проблему.

Из всех перечисленных работ наилучшие результаты в части автоматического построения части программы на высоком уровне абстракции получены в статьях [46, 47] применительно к созданию управляющей программы робота. Насколько известно авторам, исследования в области эволюционного построения систем со сложным поведением, отличных от роботов, ранее не проводились.

В табл. 2 приведены краткие описания всех упомянутых, а также некоторых других публикаций.

Таблица 2. Краткое описание публикаций

Год	Автор	Название	Описание
1962	Fogel L. J.	Autonomous Automata	Первая публикация по эволюционному программированию и первая же работа по эволюции автоматной модели. <i>Еще нет полноценного генетического алгоритма (не применяется рекомбинация, только мутация). Автоматная модель воссоздается по известному поведению в задачах предсказания следующего символа последовательности.</i>
1964	Fogel L. J.	On the organization of intellect	Докторская диссертация, ставшая основой последующих исследований и первой монографии по эволюционному программированию. Характеристики, свойственные первой публикации, сохраняются в этой и нескольких последующих работах. <i>Неполноценный генетический алгоритм. Моделирование известного поведения.</i>
1964	Fogel L. J., Owens A. J., Walsh M. J.	On the Evolution of Artificial Intelligence	Отличия от проводимого исследования аналогичны.
1965	Fogel L. J., Owens A. J., Walsh M. J.	Artificial Intelligence through a Simulation of Evolution	Статья в сборнике трудов конференции. Содержание аналогично ранее изданным работам.

1966	Fogel L. J., Owens A. J., Walsh M. J.	Artificial Intelligence through Simulated Evolution	Первая книга по эволюционному программированию, основанная на докторской диссертации, упомянутой тремя строками выше. В ней, кроме стандартных мутаций одиночных особей, рассматривалась возможность применения более сложных мутаций, оперирующих сразу с тремя или даже большим количеством автоматов. Однако способ реализации таких мутаций предложен не был. Операция выбора оставляла половину особей с лучшими показателями. Сохранение менее приспособленных автоматов рассматривалось в качестве возможности, но пока также не было реализовано.
1967	Gold E. M.	Language Identification in the Limit	В это же время начинается использование автоматически создаваемых конечных автоматов в другом направлении – распознавании языков. Впоследствии это направление получило наибольшее внимание исследователей. Автомат в данном случае определяет принадлежность заданной строки языку. <i>Распознаватель не производит выходных воздействий.</i> Результат определяется состоянием автомата после обработки входной последовательности.
1968	Fogel L. J.	Extending Communication and Control through Simulated Evolution	Работа относится к эволюционному программированию. Как и в других работах этого направления, в ней рассматривались задачи предсказания следующего символа входной последовательности и воссоздания автоматной модели системы по ее поведению. Задача автоматического управления интерпретировалась как распознавание соответствующего объекту управления автомата и поиск последовательности входных воздействий, которые приводят к требуемому поведению.
1969	Lutter B. E., Huntsinger R. C.	Engineering applications of finite automata	Конечные автоматы, кроме предсказания последовательностей, применяются для распознавания шаблонов в данных.
1969	Burgin G. H.	On playing two-person zero-sum games against nonminimax players	Начало применения автоматически генерируемых автоматов в играх.
1976	Dearholt D. W.	Some Experiments on Generalization Using Evolving Automata	В статье решаются те же задачи, что и в предыдущих работах.

1976	Atmar J. W.	Speculation on the Evolution of intelligence and its possible realization in machine form	Докторская диссертация, рассматривающая аналогичные проблемы. Интеллект представляется как свойство процесса эволюции. Сложное поведение живых организмов считается проявлением естественной эволюции, а искусственный интеллект – неизбежным свойством любой системы, содержащей эмуляцию эволюции. <i>В качестве элемента искусственного интеллекта рассматривается популяция автоматов вместе с эволюционным алгоритмом, а не оптимизированный для решения некоторой проблемы автомат.</i>
1980	Takeuchi A.	Evolutionary Automata - Comparison of Automaton Behavior and Restle's Learning Model	С целью лучшего понимания поведения людей ставится задача поиска механизма, обладающего такой же способностью к обучению, как живые существа. Как и в предыдущей работе, рассматривается не обучение решению заданной неизменной задачи, а часто встречающаяся в природе ситуация с изменяющимися условиями, которые не должны нарушать работу системы. Предложенная модель объединяет эволюционный процесс с представленной автоматами популяцией. Введен термин <i>эволюционные автоматы (evolutionary automata)</i> . Произведено сравнение поведения предложенной модели с психологической моделью Рестла. В результате сделан вывод о пригодности эволюционных автоматов для моделирования человеческого обучения.
1989	Miller J. H.	The Coevolution of Automata in the Repeated Prisoner's Dilemma	Продолжается изучение применимости эволюционирующих автоматов в играх. Рассматривается классическая задача теории игр – дилемма заключенного. Отметим, что это игра с ненулевой суммой, что отличает данную статью от ранее рассмотренной. Значительное внимание уделяется сравнению ситуаций с различным уровнем информированности игроков. Существенным моментом является рассмотрение параллельной эволюции стратегий двух игроков. Задачи, подобные этой, в которых невозможна независимая оценка одной особи (автомата), представляют дополнительные сложности, которые успешно устраняются в данной работе.



1992	Koza J.	Genetic Programming: On the programming of Computers by Means of Natural Selection	С этой работы началось генетическое программирование. Рассматривалось эволюционное построение программ, представленных древовидной структурой, в листьях которой находятся исходные данные, в узлах – операторы, а после выполнения оператора к корню дерева получается результат. Так можно описывать программу на языке <i>LISP</i> , или другом функциональном языке, правда, без рекурсии. <i>Автоматные модели вычислений в данной работе еще не применялись.</i>
1994	Clelland C. H., Newlands D. A.	Pfsa modelling of behavioural sequences by evolutionary programming	Данную статью можно отнести не только к эволюционному программированию, как заявлено в названии, но также и к распознаванию языков. <i>Вероятностные автоматы</i> используются для обнаружения регулярностей во входных последовательностях. У автоматов <i>отсутствуют выходы</i> , а результат работы каждого автомата заключается в выборе его конечного состояния.
1994	Das S., Mozer M. C.	A Unified Gradient-Descent/Clustering Architecture for Finite State Machine Induction	Рассматривается представление конечного автомата в виде рекуррентной нейронной сети. Решаемая проблема состоит в том, что, несмотря на наличие предшествовавших успешных экспериментов по обучению нейронных сетей копированию поведения автоматов, непрерывный характер изменения внутреннего состояния такой сети плохо согласуется с дискретным поведением автомата. Представлена архитектура с адаптивной кластеризацией пространства состояний сети, позволяющая дискретным состояниям автомата эволюционировать в таком представлении. Такой подход является одним из перспективных способов автоматической генерации автоматов, однако в статье он агрегируется с применением градиентного спуска для обучения нейронной сети. Градиентный спуск очень эффективен <i>для решения задач копирования эталонного поведения</i> , или, что эквивалентно, <i>распознавания языков</i> , но не расширяем на традиционные области применения генетических алгоритмов, когда вместо эталонного поведения задана оценочная функция.

1994	Kinnear K. E.	Advances in Genetic Programming. Volume 1	В этой книге было впервые предложено линейное представление эволюционирующей программы. Представление в виде графа было предложено Теллером позднее. Он в данной книге написал главу 9 под названием <i>The Evolution of Mental Models</i> . В этой главе отмечается необходимость учета внутреннего состояния обучаемой системы, что противопоставляется простому сопоставлению выходного сигнала входному. Однако внутренние состояния пока не оформляются в виде состояний автомата, а представляются содержанием индексируемой памяти.
1994	Fogel L. J., Angeline P. J., Fogel D. B.	A Preliminary Investigation on Extending Evolutionary Programming to Include Self-adaptation on Finite State Machines	В различных областях эволюционных вычислений началось исследование схем, при которых оператор мутации, модифицирующий модель в процессе эволюции, также эволюционирует, параллельно с моделью. В этой работе такой подход был впервые применен к конечным автоматам, используемым в эволюционном программировании. Предложены нормально распределенные модификации интенсивности мутации, что впоследствии послужит основой многих исследований. Однако рассматриваемая модель эволюции предполагает ограничение на интенсивность мутации. В используемом численном представлении интенсивности 0 соответствует отсутствию модификаций, а 1 – максимально допустимой мутации. В то же время, нормальное распределение определено на бесконечном интервале, что приводит к невозможности прямой реализации этого подхода.
1995	Lankhorst M. M.	A Genetic Algorithm for the Induction of Nondeterministic Pushdown Automata	Предложен генетический алгоритм, позволяющий выводить автомат с магазинной памятью из примеров принадлежащих и не принадлежащих языку автомата строк. Эта работа относится к распознавателям языков. Достоинством данной работы является использование дополнительной памяти (в данном случае стека).

1996	Teller A., Veloso M.	PADO: A New Learning Architecture for Object Recognition	Самая важная работа по теме конечных автоматов в генетическом программировании. Впервые для представления программы используется граф, допускающий циклы. Правда, этот <i>граф отражает поток данных в программе, а не управляющий автомат</i> . В таком виде автоматы представлены в генетическом программировании практически до настоящего времени. Как и в предыдущей работе, используется дополнительная память – здесь она представлена индексированной памятью и стекком. Это обеспечивает более широкие возможности по сравнению со стекком классического автомата-классификатора. Кроме того, вводятся высокоуровневые действия, которые может вызывать автомат, но их набор фиксирован и <i>позволяет решать только узкий класс задач</i> распознавания объектов по изображениям (классификация). Предложенная архитектура названа <i>PADO</i> .
1996	Angeline P., Kinnear K.	Advances in Genetic Programming. Volume 2	В этой книге автоматные модели рассматриваются в главе 3 « <i>Evolving Programmers: The Co-evolution of Intelligent Recombination Operators</i> ». Начало главы посвящено краткому изложению описанной в предыдущей работе моделью <i>PADO</i> , а основное внимание уделено «умным» генетическим операторам. Представление автоматов полностью аналогично предложенному в <i>PADO</i> .
1996	Brave S.	Evolving Deterministic Finite Automata Using Cellular Encoding	Предложен метод клеточного кодирования детерминированного конечного автомата в генетическом программировании. Этот метод имеет мало общего с популярными, но имеющими слабое отношение к рассматриваемой теме клеточным автоматам. Здесь применяется классический конечный <i>автомат-классификатор</i> , а клеточное кодирование определяет его инкрементальную эволюцию. В начале автомат состоит из единственного состояния, к которому в процессе эволюции добавляются состояния и переходы. Как и в случае других автоматов-классификаторов, <i>область применения используемой в данной работе модели резко ограничена</i> , распознаванием языков.
1996	Handley S.	A new class of function sets for solving sequence problems	Продолжение попыток применения дополнительной памяти в генетическом программировании. В данной работе выделяются статистические вычислительные зоны в последовательностях данных биологического происхождения. <i>Конечные автоматы не применяются</i> .

1996	Carmel D., Markovitch S.	Learning models of intelligent agents	Рассматриваются мультиагентные системы. Задача каждого агента состоит в поиске эффективной стратегии взаимодействия с другими агентами. Проблема состоит в зависимости оптимальной стратегии от поведения других агентов. С целью решения этой проблемы предложено описание стратегий конечными автоматами. На первом этапе взаимодействия агент строит модели поведения других участников в форме конечных автоматов. После этого находится оптимальная стратегия взаимодействия с полученными автоматами. Решение типично для интерпретации задач управления эволюционным программированием. Однако в данной работе предложен алгоритм <i>вывода автомата по примерам поведения, не использующий генетические алгоритмы.</i>
1997	Poli R.	Evolution of Graph-like Programs with Parallel Distributed Genetic Programming	Программы-графы подвергаются эволюции с целью получения высокой степени параллелизма. Характерными особенностями работы являются <i>ацикличность используемых графов и представление графом потока данных</i> в программе, а не управляющего автомата.
1997	Koza J.	Future Work and Practical Applications of Genetic Programming	Статья была написана и опубликована в электронном виде в 1996 году, а в 1997 была включена в качестве главы в книгу <i>Handbook of Evolutionary Computation</i> , для которой изначально и предназначалась. Глава рассматривает перспективы развития генетического программирования, возможности улучшения технологий и возможные области практического применения.

1997	Hsiao M.	Sequential Circuit Test Generation Using Genetic Techniques	В этой диссертации эволюционирующие автоматы используются для тестирования последовательных электрических цепей. Большая часть времени при автоматической генерации тестов для таких цепей уходит на поиск тестовых последовательностей, необходимых для обнаружения редко проявляющихся ошибок. Предложено несколько подходов к решению проблемы. Один из них объединяет генетические алгоритмы и детерминированную генерацию тестов. Трудно обнаруживаемые ошибки обнаруживаются генетическими алгоритмами и передаются детерминированному алгоритму для генерации тестовых последовательностей. Второй, основанный на эмуляции, подход представляет больший интерес. При этом тестовые последовательности генерируют генетически построенные конечные автоматы.
1998	Banzhaf W., Nordin P., Keller R., Francone F.	Genetic Programming. An Introduction : On the Automatic Evolution of Computer Programs and Its Applications	В книге выполнен обзор работ по генетическому программированию. Рассматриваются все основные формы представления программ (деревья, линейные структуры и графы), но не предложено какой-либо новой модели автомата. Таким образом, автоматы, как и в большинстве предыдущих публикаций, остаются схемами потоков данных.
1998	Belz A., Eskikaya B.	A genetic algorithm for finite state automata induction with an application to phonotactics	Используются автоматы-классификаторы в применении к весьма узкой задаче.
1998	Brameier M., Kantschik W., Dittrich P., Banzhaf W.	<i>SYSGP</i> – A C++ library of different GP variants	Библиотека, обеспечивающая работу с основными формами представления программ. Автоматы представлены архитектурой <i>PADO</i> .
1998	Langdon W.	Genetic Programming and Data Structures	Продолжение исследования проблемы дополнительной памяти в генетическом программировании. Использование индексируемой памяти автор считает устаревшим. Вместо этого рассматривается использование нескольких других стандартных структур данных, таких как предшествовавший индексируемой памяти стек, очередь, список и кольцо.

1999	Spector L., Langdon W., O'Reilly U., Angeline P.	Advances in Genetic Programming. Volume 3	Как и в предыдущем томе этой книги, одна глава, написанная Теллером, рассматривает использование автоматов в генетическом программировании. Это глава 14 – <i>The Internal Reinforcement of Evolving Algorithms</i> . Как и в предыдущем томе, основное содержание главы имеет слабое отношение к теме исследований по контракту. Предложен способ учета генетической истории особи-программы при ее дальнейшей генетической модификации. В основе предложенного подхода лежит представление программы в виде графа. Правда, <i>граф отражает поток данных при вычислении, а не управляющий автомат</i> .
1999	Kantschik W., Dittrich P., Brameier M., Banzhaf W.	Empirical Analysis of Different Levels of Meta-Evolution	Несмотря на то, что это не отражено в названии, эволюционной модификации подвергаются графы. Графами же представляются и модифицирующие их операторы, которые также параллельно модифицируются. В основе представление графов остается <i>PADO</i> .
1999	Kantschik W., Dittrich P., Brameier M., Banzhaf W.	Meta-Evolution in Graph GP	Используется графовое представление автоматов. Приведено сравнение эффективности мета-эволюции с другими подходами.
1999	Ngom L., Baron C., Geffroy J.	Genetic Simulation for Finite State Machine Identification	Предложен новый подход к воссозданию автоматной модели <i>заданной поведением системы</i> .
2000	Belz A.	Computational Learning of Finite-State Models for Natural Language Processing	Рассматривается обработка естественных языков – применяются <i>автоматы-классификаторы</i> .
2000	Benson K.	Evolving Automatic Target Detection Algorithms that Logically Combine Decision Spaces	Используется модель <i>PADO</i> с дополнительными специфичными для задачи модулями.
2000	Spears W., Gordon D.	Evolving Finite-State Machine Strategies for Protecting Resources	В отличие от большинства публикаций, в этой работе автомат анализирует несколько параллельных входов. Однако количество входов фиксировано и равно всего четырем. Причем каждый вход может принимать одно из трех значений, а некоторые комбинации входных значений недопустимы. Таким образом, этот набор входов успешно представляется одним входом с не более чем 80 значениями.

2000	Teller A., Veloso M.	Internal reinforcement in a connectionist genetic programming approach	Модель <i>PADO</i> в аспекте автоматов остается без изменений. Рассматривается способ повышения эффективности генетического процесса.
2001	Frey C., Leugering G.	Evolving Strategies for Global Optimization. A Finite State Machine Approach	Оптимизируется программа контроллера. При этом, в отличие от ранее рассмотренных публикаций, применяются управляющие автоматы Мили.
2002	Ashlock D., Wittrock A., Wen T-J.	Training finite state machines to improve PCR primer design	Автоматы применяются для конструирования строительных блоков хромосом.
2002	Kantschik W., Banzhaf W.	Linear-Graph GP. A new GP Structure	Предложено объединение линейной модели генетического программирования с графом. При этом <i>граф</i> , как и в большинстве рассмотренных работ, <i>отражает поток данных</i> .
2003	Lucas S.	Evolving Finite State Transducers: Some Initial Explorations	Расширение хорошо изученной области автоматов-классификаторов до автоматов-преобразователей. Однако, как и в классификаторах, для обучения преобразователя <i>предоставляется эталонное поведение</i> (выходная строка для каждой входной). Кроме того, если представлять преобразователь в качестве программы в генетическом программировании, она будет иметь <i>специфичную область применения</i> . При этом, как и указано в статье, вероятно большая эффективность по сравнению с универсальным подходом.
2003	Miller J., Thomson P.	A Developmental method for growing Graphs and Circuits	Электрические схемы представляются автоматом, который растет подобно совокупности биологических клеток, где клетка соответствует каждому состоянию автомата.
2003	Petrovic P.	Simulated evolution of distributed FSA behaviour-based arbitration	Предложено использование эволюционирующих автоматов для управления действиями роботов. Автоматы достаточно высокоуровневые. Каждой подсистеме робота сопоставляется отдельный автомат.
2004	Ashlock D., Emrich S., Bryden K. et al.	A comparison of evolved finite state classifiers and interpolated markov models for improving PCR primer design	В работе рассматриваются автоматы-классификаторы в применении к геной инженерии.
2005	Petrovic P.	Evolving automaton for distributed behavior arbitration	Продолжение исследования применения автоматов в робототехнике. Рассматривается инкрементальная эволюция, позволяющая значительно сократить вычислительную сложность.

2006	Ashlock D.	Evolutionary Computation for Modeling and Optimization	Несмотря на очень широкую область охвата (все эволюционные вычисления), в книге рассматривается применение графов в генетическом программировании в ранее рассмотренном представлении.
2006	Petrovic P.	Comparing Finite-State Automata Representation with GP-trees	Предложено новое представление автоматов. При выборе перехода осуществляется сравнение значений двух регистров или значения регистра с константой.
2007	Lucas S., Reynolds T.	Learning Finite-State Transducers: Evolution Versus Heuristic State Merging	Предложен новый метод <i>воссоздания автомата по языку</i> . Эволюционирует только матрица переходов, а символы в состояниях выбираются детерминировано. Такой подход позволяет существенно сократить пространство поиска и превосходит по эффективности использования вызовов оценочной функции популярный алгоритм <i>EDSM</i> .

## 2.2. ОБЗОР ИЗБРАННЫХ РАБОТ

Далее будут подробно рассмотрены работы, наиболее близко соответствующие направлению проводимого по контракту исследования, и указаны их отличия.

### 2.2.1. PADO: a New Learning Architecture for Object Recognition

В последнее время большинство систем искусственного интеллекта работают с простыми задачами и искусственными областями применимости, так как основаны на точном восприятии мира задачи. Распознавание объектов – крайне важная составляющая задачи восприятия мира. Машинное обучение призвано переместить распознавание объектов в реальный мир. Так как на сегодняшний день машинное обучение не достигло успеха в распознавании объектов, предлагается исследовать другое направление: архитектуры обучения. В данной статье описывается метод обучения, основанный на архитектуре, названной *PADO (Parallel Algorithm Discovery and Orchestration)* [53].

Поставлена конкретная задача распознавания образов, похожая на задачу, стоящую перед автоматом-грамматикой. Однако в данном случае автомат представляет не грамматику, а компьютерную программу. В каждом состоянии автомата вызывается более низкоуровневая процедура, которая может быть встроенной или представляться другим автоматом. В качестве дополнительной памяти (кроме памяти в состояниях) используется обычная для компьютеров индексируемая память и стек. Описывается адаптированный генетический алгоритм и детали функционирования получаемой программы. Вводятся эволюционирующие вместе с программой “умные” генетические операторы. Это первая, и очень удачная, попытка применения автоматных моделей в генетическом программировании. Она напоминает исследование, проводимое по контракту, но имеет отличия, перечисленные ниже.

- Специальное применение – решается задача распознавания объектов, а не создания программ в общем виде.
- Низкоуровневые автоматы – программа описывается единообразно до достаточно низкого уровня, включающего, например, операции с индексируемой памятью. Такой подход типичен для генетического программирования и позволяет достичь высокой универсальности. В исследовании, проводимом по контракту, автоматами описывается высокоуровневая управляющая компонента программы, а низкоуровневые операции ин-



капсулируются в объекте управления, что сокращает пространство поиска и делает результирующие автоматы понятными человеку.

- Одна переменная, анализируемая при переходе. Более конкретно – выбор состояния, в которое будет осуществлен переход, осуществляется на основе анализа одной переменной. Таким образом, получается всего два варианта перехода из каждого состояния. Одновременный учет нескольких входов автомата неосуществим. В случае, когда допустим последовательный анализ переменных, решение задач с большим числом параллельных входов приведет к существенному росту автомата, пространства поиска и требований к ресурсам.

### 2.2.2. Genetic Programming – An Introduction. On the Automatic Evolution of Computer Programs and its Application

В книге [8] генетическое программирование рассматривается в разных вариациях, в том числе и с использованием автоматных моделей. Описывается та же модель *PADO*, основанная на графах система генетического программирования. При этом название *PADO* никоим образом не указывает на использование графов для эволюции. Графы способны компактно представлять очень сложные структуры программ. Структуры в виде графов – это всего лишь узлы, соединенные ребрами. Можно считать ребро указателем перехода управления от одного узла к другому.

*PADO* не просто допускает циклы и рекурсию – он их поощряет. Это существенно – другие системы генетического программирования очень осторожно экспериментировали с циклами и рекурсией из-за огромных трудностей, которые они порождают. В этой работе операторы генетического программирования на графах не рассматриваются..

### 2.2.3. Empirical Analysis of Different Levels of Meta-Evolution

В статье [31] развивается идея эволюционирующих генетических операторов. Анализируются различные уровни мета-эволюции с использованием систем генетического программирования на основе графов. Система позволяет представлять как особей в пространстве поиска, так и сами генетические операторы в виде программ-графов, различающихся только набором операторов (подобно “умным” операторам в *PADO*). Семь вариантов мета-эволюции проверяются на трех реальных задачах классификации. Наиболее сложный вариант состоит из трех мета-уровней, где программа в виде графа на первом уровне осуществляет генетический поиск в базовом пространстве особей, программа-граф на втором уровне модифицирует программы первого уровня, а программы третьего мета уровня изменяют программы второго уровня и себя. На примерах показана возможность успешного применения мета-уровней. Можно выделить следующие отличия от исследования, проводимого по контракту.

- Суть работы – в эволюции только представленных графами генетических операторов.
- Основа представления программ в виде графов, в то время как в исследовании, проводимом по контракту, разрабатывается собственное, отличное от *PADO*, более высокоуровневое представление.
- Специальное применение – задачи классификации.

### 2.2.4. Meta-Evolution in Graph GP

По сути, статья [32] повторяет предыдущую, но имеет несколько смещенные акценты.

В данной работе представлена эволюция операторов генетического программирования методами генетического программирования. Более конкретно, мета-эволюция операторов рекомбинации в генетическом программировании на основе графов применяется и сравнивается с другими методами изменения и рекомбинации в генетическом программировании на основе графов. Демонстрируется,

что непосредственное применение операторов рекомбинации к ним самим не дает желаемого результата. После ввода дополнительного уровня операторов рекомбинации (мета-уровня), которые модифицируют множество операторов рекомбинации, даже само-рекомбинация дополнительных операторов становится применимой. На задаче распознавания речи показано, что общая эффективность данной системы выше, чем в других вариантах генетического программирования на основе графов.

Основные отличия от выполняемого по контракту исследования аналогичны предыдущим случаям. Например, рассматривается только задача классификации.

### 2.2.5. Internal Reinforcement in a Connectionist Genetic Programming Approach

В работе [52] исследуется оперирующий графами генетический алгоритм. Для оптимизации эволюции каждому состоянию или переходу сопоставляется некоторая оценка, основанная на истории эволюции. Таким образом, преимущество генетических алгоритмов, состоящее в способности выделять и комбинировать полезные составляющие хромосом, усиливается.

Генетическое программирование – успешная техника машинного обучения, предоставляющая мощные параметризованные примитивы конструкций программ, используя эволюцию в качестве механизма поиска. Однако, в отличие от некоторых техник машинного обучения, таких как, например, искусственные нейронные сети, генетическое программирование не имеет заданной процедуры изменения частей найденной структуры на основе прошлой успешности этой структуры. Генетическому программированию недостает понятной, локально оптимальной процедуры преобразования, эквивалента градиентного спуска обратного распространения ошибки в искусственных нейронных сетях. В статье предлагается новый механизм определения эффективности программы в процессе ее эволюции. Этот алгоритм, названный “внутренним усилением” (*internal reinforcement*), применен для представления автоматных эволюционирующих параметризованных программ, называемого “нейронным программированием” (*neural programming*). Представлен алгоритм генерации оценок составляющих программы в процессе обучения на основе нейронного программирования и внутреннего усиления. Статья включает краткий обзор генетического программирования и эмпирические эксперименты, демонстрирующие ускорение обучения при использовании предложенного подхода. Отмечается, что данная статья не описывает *PADO*, но экспериментальная проверка производилась в рамках его контекста. Это и определяет отличия от исследования, проводимого по контракту.

- Специальное применение – задачи классификации. Точнее, были рассмотрены задачи классификации изображений объектов (как в *PADO*) и акустических сигналов.
- Низкоуровневые автоматы. Представление автомата изменилось, но сохранило типичный для генетического программирования уровень операций.
- При переходах анализируемая только одна переменная.

### 2.2.6. SYSGP – A C++ library of different GP variants

Отчет [12] описывает библиотеку для реализации системы генетического программирования, позволяющей использовать различные модели представления программы, включая графы:

В работе отмечается, что в последние годы появились различные варианты генетического программирования. Все они основываются на идее автоматической эволюции компьютерных программ. Сегодня используется три основных формы представления генетических программ: деревья, графы и линейные структуры. В данном отчете представлена система под названием *SYSGP*, поддерживающая все указанные представления. Она позволяет исследователям экспериментировать с различными представлениями с минимальной избыточностью реализации. Более того, система представляет возможность комбинации модулей с различными представлениями внутри одной генетической программы. Библиотека *SYSGP* реализована как C++ библиотека с использованием шаблонов, оперирующих генетическими типами данных. Для представления автоматных блоков вновь используется *PADO*.

Используя комбинацию автомата на верхнем уровне с древовидными и линейными структурами на нижних уровнях, пользователь может добиться повышения уровня абстракции автоматной составляющей, но этот вопрос авторами библиотеки не рассматривается.

### 2.2.7. Evolving Automatic Target Detection Algorithms that Logically Combine Decision Spaces

В работе [11] представлен новый подход к активно решаемой в рассматриваемой предметной области задаче классификации. Классифицирующие (дискриминантные) функции строятся в виде комбинаций свойств классифицируемых объектов с простыми математическими функциями, такими как  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $max$ ,  $min$ . Такие классифицирующие функции способны формировать нелинейные прерывные гиперповерхности. Несколько классифицирующих функций могут объединяться логическими операторами перед классификацией. Разработан алгоритм, способный определить, когда требуется комбинация классифицирующих функций, а когда достаточно одной функции. Используемые для классификации алгоритмы не пишутся вручную. Они получаются в результате эволюции разновидности конечного автомата. Используемая модель названа *EMMA (Environment manipulating mutable automaton)* и состоит из конечного автомата на верхнем уровне и древовидных структур генетического программирования, соответствующих каждому состоянию. Кроме того, каждому состоянию автомата сопоставлены две логические функции, используемые для классификации. Таким образом, данная работа имеет не характерное для ранее рассмотренных публикаций сходство с исследованием, проводимым по контракту – относительно высокоуровневый автомат. Правда, этот автомат все еще сохраняет значительное сходство с классическими моделями генетического программирования. Например, состояния автомата представляют не управляющие состояния программы, а ее вычислительные блоки. При переходе из одного состояния в другое выходное значение начального состояния становится входным значением нового состояния. Автомат описывает распространение данных, аналогично дереву в генетическом программировании. В исследовании, проводимом по контракту, используются управляющие автоматы, а потоки данных инкапсулированы в объекте управления. Остаются и другие различия, например, специальное применение (только задачи классификации)..

### 2.2.8. Linear-Graph GP. A new GP Structure

До публикации [30] основными формами представления генетических программ были деревья, линейные структуры и графы. В данной работе представлен новый тип структуры генетического программирования, полученный путем комбинирования автомата на верхнем уровне и линейных подпрограмм на нижнем, и названный *linear-graph*. Эта структура стала дальнейшим развитием разработанной ранее авторами данной работы структуры под названием *linear-tree*. В статье описывается предлагаемая структура, а также генетические операторы для ее модификации. Эффективность оценена в сравнении с линейными и древовидными программами. *Linear-graph* – это граф без циклов, каждая вершина которого представлена линейной программой. При входе в состояние запускается соответствующая подпрограмма. После выполнения подпрограммы в случае единственной исходящей дуги осуществляется переход, а при наличии нескольких дуг необходимая дуга выбирается в результате вычисления функции выбора перехода. Таким образом, эффективный анализ многих переменных при выборе перехода может быть проведен только вручную. Можно выделить следующие нетипичные для других работ отличия.

- Не допускаются циклы в графе.
- Выбор перехода выполняется на основе сравнения двух переменных или переменной с константой. Точнее, в статье не указано ограничения количества анализируемых переменных, но в примерах используется сравнение не более двух переменных и проблема анализа значительного числа переменных не рассматривается.

### 2.2.9. Evolution of Graph-like Programs with Parallel Distributed Genetic Programming

Работа [48] похожа на предыдущую ациклическостью применяемых графов. Граф также представляет структуру и поток выполнения программы, но уже распараллеливаемой. Высокий уровень параллелизма получаемой программы – основная цель этого исследования.

*Параллельное распределенное генетическое программирование (Parallel Distributed Genetic Programming, PDGP)* – новая форма генетического программирования, пригодная для разработки программ с высокой степенью параллелизма. Программы в *PDGP* представляются как графы, узлы которых представляют функции и терминальные вершины, а связи – поток управления и результаты. В статье приведены используемые в *PDGP* формы представления программ, операторы и интерпретаторы. Описаны эксперименты, в которых *PDGP* сравнивается со стандартными техниками генетического программирования. В качестве основных отличий можно выделить следующие.

- Низкоуровневые графы, как в стандартном генетическом программировании.
- Отсутствие циклов.
- Специальное применение – генерация программ с высоким уровнем параллелизма. Результат будет эффективным при наличии соответствующего вычислителя.
- Отсутствие выбора перехода. Результат выполнения соответствующей состоянию функции передается на все исходящие связи.

### 2.2.10. Evolving Strategies for Global Optimization. A Finite State Machine Approach

В статье [27] методы генетического программирования применяются для поиска правил перехода двухступенчатой дискретной динамической системы. Рассматривается формирование выходных значений контроллера на основе входных значений. В рамках решения задачи оптимального управления производится оптимизация программы контроллера, состоящей из дерева, описывающего фильтр входных сигналов и конечного автомата Мили. Применение такой формы конечного автомата является существенным отличием от ранее рассмотренных исследований в области генетического программирования. Автомат Мили в данном случае управляющий, как и в исследовании, проводимом по контракту, а не вычисляющий, как в классическом генетическом программировании. В качестве отличий можно выделить следующее:

- Специальное применение в контроллерах. Объект управления представлен управляющим устройством, в то время как исследование, проводимое по контракту, не ограничивается генерацией автоматов для управления устройствами, а уделяет значительное внимание системам, в которых объект управления автомата не задан решаемой задачей, как устройство, которым требуется управлять, а разрабатывается вместе с автоматом для решения внешней для автоматизированного объекта задачи.
- Заданная задачей функция приспособленности (оптимизируемый функционал). В нашем исследовании задание такой функции рассматривается как одна из основных задач, стоящих перед программистом.

### 2.2.11. Simulated evolution of distributed FSA behaviour-based arbitration

Область *эволюционной робототехники (Evolutionary robotics, ER)* занимается автоматическим созданием контроллеров роботов с использованием эволюционных алгоритмов. В большинстве случаев задача состоит в создании программного контроллера для мобильного робота. При этом контроллер обычно представляет собой программу, эволюционирующую с использованием генетического программирования. Также встречаются и другие архитектуры. Контроллер обычно принимает данные с датчиков и генерирует управляющие сигналы для исполнительных устройств, возможно учи-

тывая и модифицируя некоторое внутреннее состояние. Однако, архитектура полученного таким образом контроллера обычно равномерна, без внутренней иерархии или структуры. Более того, эволюцию обычно делают ответственной за генерацию всего поведения робота, определение управляющих команд или действий и обработку входных данных до самого низкого уровня. К сожалению, физическое взаимодействие мобильных роботов с их окружением имеет очень сложную природу, еще сильнее усложняемую погрешностями датчиков и приводов. В результате, даже очень простая задача, как, например, обнаружение узнаваемого объекта и доставка в одно из двух помеченных мест в зависимости от его цвета, достаточно сложна для такого эволюционного решения. Кроме того, надежность полученного поведения робота сомнительна. Нельзя с уверенностью сказать, что детально описанное поведение будет иметь желаемый эффект. Однако возможность автоматической генерации контроллеров была бы важным преимуществом эволюционной робототехники, если бы удалось обойти указанные трудности.

В работе [47] предлагается использовать эволюционирующий конечный автомат для выбора действий, каждое из которых описывается вручную или автоматически, но другим способом, и не изменяется в ходе эволюции. Ставится задача разработки эффективного метода автоматического проектирования контроллеров при заданных базовых элементах поведения робота как на верхнем, так и на нижнем уровнях. Предлагается создавать распределенный механизм выбора действий, сопоставляя каждому модулю поведения отдельный управляющий автомат. Для экспериментов используется аппаратная платформа *LEGO RCX*. Отличия от нашего исследования следующие.

- Выбор перехода на основе анализа одной переменной-сообщения. Такую схему можно интерпретировать как учет только событий, но не защитных условий. Проблема решения задач с большим числом входных воздействий решается через применение множества параллельно работающих автоматов, каждый из которых отвечает за определенную деятельность.
- Специальное применение – робототехника.

### 2.2.12. Evolving automata for distributed behavior arbitration

Работа [46], по сути, повторяет предыдущую, но основное внимание в ней уделяется инкрементальным эволюционным алгоритмам, когда задача делится на части, которые решаются последовательно, сокращая тем самым пространство поиска и требования к ресурсам. Отмечаются следующие преимущества автоматического построения механизма выбора действий.

Автоматические методы могут рассматривать неочевидные решения, которые были бы пропущены человеком при ручном и полуавтоматическом проектировании. Мобильные роботы могут создаваться из расчета широкой применимости, а механизмы выбора действий могут потребоваться разные. В таких случаях может потребоваться генерация механизма выбора действий на основе описания задачи. Наличие возможности автоматического построения таких механизмов может сэкономить значительные объемы работ, необходимых при ручном построении каждого механизма. Кроме того, при ручном проектировании могут возникать трудности из-за недостаточного понимания имеющего место в реальности взаимодействия робота с окружением. Такие взаимодействия бывает сложно описать аналитически. Автоматическое проектирование способно анализировать характеристики взаимодействий робота более надежно, эффективно и точно. В аспекте отличий от проверяемого исследования данная работа эквивалентна предыдущей. В частности, используется та же схема выбора переходов в автоматах.

### 2.2.13. Comparing Finite-State Automata Representation with GP-trees

В работе [45] анализируется эффективность применения автоматных программ в генетическом программировании по сравнению с деревьями. Анализируются структуры и свойства представления

программ, адекватные решаемой задаче. По словам автора, в предыдущих его работах для управления роботами автоматы использовались по следующей причине: автоматы схожи по структуре с задачей контроллера робота – робот в процессе выполнения некоторой деятельности всегда находится в некотором состоянии и отвечает на внешние воздействия немедленными действиями или переходом в другое состояние. Таким образом, деятельность робота может быть точно смоделирована диаграммой переходов. Кроме того, по мнению автора, конечные автоматы более просты в понимании, анализе и верификации, чем другие представления, такие как, например, нейронные сети. Автоматы удобнее для инкрементальной эволюции контроллеров, исследование которой продолжается в этой работе. При этом расширение функциональности состоит в добавлении состояний и переходов, при относительно небольших изменениях в ранее существовавших состояниях и переходах, в то время как архитектуру нейронных сетей в таких случаях часто приходится изменять очень значительно, за исключением случаев применения какого-нибудь модульного подхода. Однако, исследование модульных подходов к проектированию нейронных сетей еще находится на очень ранней стадии своего развития.

Используемые в этой статье конечные автоматы отличаются от ранее использованных автором классических автоматов, применяемых в генетическом программировании, и от автоматов, используемых в исследовании, проводимом по контракту. В рассматриваемой работе в качестве условия перехода используется сравнение значений двух регистров, или значения регистра с константой (напомним, что ранее автор использовал сообщения, а в исследовании, проводимом по контракту, анализируются многоместные булевы формулы). Кроме того, упоминаются взвешенные автоматы и модели Маркова. Сделан вывод о том, что большая эффективность деревьев и автоматов зависит от решаемой задачи. Деревья более удобны в случае линейных программ, а автоматы – в случае повторяющихся и возможно нерегулярных шаблонов поведения, с возможностью изменения режима функционирования. Для новой модели автомата характерны следующие особенности:

- две анализируемые при выборе перехода переменные;
- не обеспечивается полнота условий на переходах. В случае отсутствия перехода, условие которого выполняется, программа завершается;
- непротиворечивость переходов обеспечивается через их упорядочивание.

#### **2.2.14. Swarm Rule-Based Development using Genetic Programming Techniques**

Ранее рассматривалось применение генетических автоматов для управления роботами. В статье [34] исследуется случай, когда роботов или других агентов, участвующих в решении задачи, несколько, и требуется обеспечение их совместной деятельности. Программирование такого рода агентов для выполнения определенной задачи требует большого времени. Предложена система программирования эволюционирующих систем правил для стай (swarm) агентов. Алгоритмы-кандидаты представляются в используемой системе генетического программирования конечными автоматами. Описывается решение задачи сбора объектов: стая агентов должна найти распределенные на плоскости объекты и доставить их в определенное место.

При выборе перехода автомата могут учитываться показания нескольких датчиков, что является значительным усложнением по сравнению с ранее рассмотренными работами. Однако выбранное представление условий переходов обладает недостатками.

- Описание условия перехода объединением полных наборов показаний датчиков. Таким образом, проблема быстрого роста размерности пространства поиска с увеличением числа входных воздействий (датчиков) не решена.
- Полнота описания переходов обеспечивается через переходы по умолчанию без действий и изменения состояния.
- Выполняется только одно действие на переходе.

### **2.2.15. Genetic Simulation for Finite State Machine Identification**

Другим направлением, применяемым для построения неизвестной модели вычислений, является ее распознавание (identification). Методы распознавания, формальные или основанные на эмуляции работы модели, используются в логическом проектировании, тестировании и последовательном обучении. Суть этих методов заключается в выводе автоматной модели заданной последовательной системы из функционального описания ее поведения. В работе [44] представлен новый подход к распознаванию модели, основанный на применении генетических алгоритмов. Выполнен анализ использования генетических алгоритмов и предложен подход к их применению для функционального распознавания модели. Описывается экспериментальная проверка предложенного подхода.

Распознавание модели заданной системы принципиально отличается наличием эталонного поведения от эволюционного построения модели на основе оптимизации оценочной функции. Задача распознавания характерна для эволюционного программирования.

## **2.3. РЕЗУЛЬТАТЫ НЕПАТЕНТНЫХ ИССЛЕДОВАНИЙ**

Исследование публикаций по эволюционному построению конечных автоматов показало, что работы по этой тематике в мире проводятся. При этом в эволюционном программировании конечные автоматы использовались с самого начала. Анализ показал, что применение автоматов наиболее целесообразно для управления роботами и другими автономными агентами. Однако ни в одной из найденных публикаций, также как и в патентах, не было найдено описания подхода, полностью удовлетворяющего требованиям, предъявляемым к автоматам в рамках работ по контракту, основным из которых является наличие сложных условий, обеспечивающих переходы между состояниями.

### 3. ОТЛИЧИЯ ПРОВОДИМОГО ИССЛЕДОВАНИЯ

Проверяемое на патентоспособность исследование ориентировано на применение эволюционного построения автоматов для автоматизации широкого класса программных систем со сложным поведением, в то время как многие из рассмотренных патентов и публикаций применимы только в узкоспециальных областях. По этой причине можно исключить из сравнения все рассмотренные японские патенты (JP 10-0636346 JP 10-307804, JP 10-307806, JP 2004/004521), почти все рассмотренные патенты США (US 5327544, US 5615124, US 5761386, US 5943659, US 6839698, US 7136947) и публикации [4, 5, 6, 9, 10, 11, 13, 14, 16, 17, 19, 27, 29, 34, 38, 39, 42, 43, 46, 47, 48, 49, 53]

В исследовании, проводимом по контракту, задача, которую должна решать полученная в результате эволюции программа, задается функцией приспособленности. Таким образом, достаточно наличия способа оценки качества программы. Такой ситуации можно противопоставить подход, используемый в патенте США (US 4829450) и публикациях [15, 18, 20, 21, 22, 23, 24, 25, 26, 39, 40, 41, 44], в которых для обеспечения процесса эволюции требуется наличие эталона поведения программы.

В нашем исследовании детерминированные конечные автоматы эволюционируют в результате применения генетического алгоритма. В патенте WO 98/02825, докладе [28] и книгах [33, 36] не применяются автоматы, а в патентах EP 1202141, WO 2004/021181 и в докладе [15] нет генетического алгоритма.

В проводимом исследовании предполагается использование автоматов, управляющих вычислениями, инкапсулированных в объекте управления, который может представлять произвольную структуру данных. В публикациях [3, 8, 12, 30, 31, 32, 50, 52, 53] автомат представляет вычислительную структуру программы и поток данных в ней, что принципиально отличается от управляющего автомата. Данные передаются от одного состояния, представляющего оператор, другому, часто через стек. В работе [37] рассматривается применение лишь ограниченного набора структур данных. Подход, разрабатываемый в исследовании по контракту, значительно шире.

В проводимом исследовании эволюция осуществляется в процессе разработки программы. После завершения эволюции полученная программа используется без изменений. В работах [7, 51] рассматриваются адаптивные системы, включающие генетический алгоритм в конечный продукт. Кроме того, в работе [51] автоматы используются для моделирования обучения людей (а не для создания программ). Статья [35] представляет обзор достижений и описание возможных направлений дальнейшего развития, не описывая каких-либо новых подходов.

Основное преимущество проводимого исследования – решение проблемы экспоненциального роста размерности пространства поиска с увеличением количества параллельных входов программы. Эта проблема не рассматривалась ни в одной из рассмотренных публикаций и ее решение не было найдено в патентах. В работах [46, 47] для специфичного применения в робототехнике проблема снимается параллельно работающими автоматами, каждый из которых анализирует на более двух переменных за раз. При этом для каждого отдельного автомата проблема остается нерешенной. В работе [49] у автомата четыре троичных входа, но это не более 80 комбинаций, количество входов фиксировано и проблема роста не рассматривается. В работе [45] допускается сравнение значений двух регистров, а во всех остальных публикациях у автомата единственный вход, размер алфавита которого предполагается ограниченным, и проблема его роста не рассматривается.

Таким образом, выполнены анализ патентных и непатентных источников показал, что рассматриваемая в работе по контракту задача в настоящее время не решена.



## **ЗАКЛЮЧЕНИЕ**

Данное патентное исследование, проведенное в рамках первого этапа работы по государственному контракту № 02.514.11.4044, показало, что в настоящее время отсутствуют патенты и иные охраняемые документы, которые могут препятствовать применению в Российской Федерации технологии генетического программирования для генерации автоматов управления системами со сложным поведением, разрабатываемой в рамках указанной работы. Таким образом, в соответствии с пунктами 5.2 и 5.3 технического задания обеспечена патентная чистота результатов и отсутствие препятствий для их применения.

На дальнейших этапах работ по указанному контракту необходимо проведение дополнительных патентных исследований для выявления вновь появившихся патентов и непатентных источников в рассматриваемой области.

### ЛИТЕРАТУРА И ИСТОЧНИКИ

1. ГОСТ Р. 15.011-96 «Система разработки и постановки продукции на производство. Патентные исследования»
2. *Патентный закон РФ*. <http://www.legal-support.ru/information/laws/intellect/patent-law.html>
3. *Angeline P., Kinnear K.* *Advances in Genetic Programming. Volume 2.* Cambridge: MIT Press, 1996. 554 P.
4. *Ashlock D., Emrich S., Bryden K. and others* A comparison of evolved finite state classifiers and interpolated markov models for improving PCR primer design / 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'04). 2004, pp.190–197.
5. *Ashlock D.* *Evolutionary Computation for Modeling and Optimization.* NY: Springer, 2006. 572 P.
6. *Ashlock D., Wittrock A., Wen T-J.* Training finite state machines to improve PCR primer design // Congress on Evolutionary Computation (CEC'02). 2002, pp.13–18.
7. *Atmar J.* Speculation on the Evolution of intelligence and its possible realization in machine form. PhD thesis. New Mexico State University. 1976.
8. *Banzhaf W., Nordin P., Keller R., Francone F.* *Genetic Programming : An Introduction : On the Automatic Evolution of Computer Programs and Its Applications.* CA: Morgan Kaufmann Publishers, 1998. 450 P.
9. *Belz A.* Computational Learning of Finite-State Models for Natural Language Processing. PhD thesis. University of Sussex. 2000.
10. *Belz A., Eshkikaya B.* A genetic algorithm for finite state automata induction with an application to phonotactics / ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing. Saarbruecken, 1998, pp.9–17.
11. *Benson K.* Evolving Automatic Target Detection Algorithms that Logically Combine Decision Spaces / Proceedings of the 11th British Machine Vision Conference. Bristol: BMVA Press. 2000, pp. 685–694. <http://www.bmva.ac.uk/bmvc/2000/papers/p69.pdf>
12. *Brameier M., Kantschik W., Dittrich P., Banzhaf W.* SYSGP – A C++ library of different GP variants. Internal Report. Univ. of Dortmund. 1998.
13. *Brave S.* Evolving Deterministic Finite Automata Using Cellular Encoding / Genetic Programming 1996: Proceedings of the First Annual Conference. CA: MIT Press. 1996, pp. 39–44. <http://citeseer.ist.psu.edu/brave96evolving.html>
14. *Burgin G.* On playing two-person zero-sum games against nonminimax players // IEEE Trans. on Systems Science and Cybernetics. 1969. V. 5, I. 4, pp. 369–370. <http://ieeexplore.ieee.org/iel5/4082035/4082254/04082270.pdf>
15. *Carmel D., Markovich S.* Learning Models of Intelligent Agents / Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference. California: AAAI Press. 1996. V. 2, pp. 62–67. <http://citeseer.ist.psu.edu/carmel96learning.html>
16. *Clelland C., Newlands D.* Pfsa modelling of behavioural sequences by evolutionary programming // Complex'94 – Second Australian Conference on Complex Systems. IOS Press, 1994, pp.165–172.
17. *Das S., Mozer M.* A Unified Gradient-Descent/Clustering Architecture for Finite State Machine Induction / Advances in Neural Information Processing Systems. 1994.
18. *Dearholt D.* Some Experiments on Generalization Using Evolving Automata / Proceedings of the 9th International Conference on System Sciences. Honolulu. 1976, pp. 131–133.

19. *Gold E.* Language Identification in the Limit // *Information and Control*. 1967. № 10, pp.447–474.
20. *Fogel L., Angeline P., Fogel D.* A Preliminary Investigation on Extending Evolutionary Programming to Include Self-adaptation on Finite State Machines // *Informatica*. 1994. V. 18, I. 4, pp. 387–398.
21. *Fogel L., Owens A., Walsh M.* *Artificial Intelligence through Simulated Evolution*. NY: John Wiley & Sons, 1966.
22. *Fogel L., Owens A., Walsh M.* *Artificial Intelligence through a Simulation of Evolution / Biophysics and Cybernetic Systems* Biophysics and Cybernetic Systems: Proceedings of the Second Cybernetic Sciences Symposium. London: Macmillan & Co. 1965, pp. 131–155.
23. *Fogel L.* Autonomous Automata // *Industrial Research*. 1962. V. 4, I. 1, pp. 14–19.
24. *Fogel L.* Extending Communication and Control through Simulated Evolution / *Bioengineering - An Engineering View: Symp. Engineering Significance of the Biological Sciences*. San Francisco: San Francisco Press. 1968, pp. 286–304.
25. *Fogel L., Owens A., Walsh M.* On the Evolution of Artificial Intelligence / *Proceedings of the 5th National Symposium on Human Factors in Electronics*. San Diego. 1964, pp. 63–76.
26. *Fogel L.* On the organization of intellect. PhD thesis. University of California. 1964.
27. *Frey C., Leugering G.* Evolving Strategies for Global Optimization. A Finite State Machine Approach / *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. CA: Morgan Kaufmann. 2001, pp. 27–33. <http://citeseer.ist.psu.edu/frey01evolving.html>
28. *Handley S.* A New Class of Function Sets for Solving Sequence Problems / *Genetic Programming 1996: Proceedings of the First Annual Conference*. Cambridge: MIT Press. 1996, pp. 301–308. <http://cognet.mit.edu/library/books/chapter?isbn=0262611279&=chap39>
29. *Hsiao M.* Sequential Circuit Test Generation Using Genetic Techniques. PhD thesis. University of Illinois at Urbana-Champaign. 1997. <http://citeseer.ist.psu.edu/article/hsiao97sequential.html>
30. *Kantschik W., Banzhaf W.* Linear-Graph GP. A new GP Structure / *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2002*. Kinsale: Springer-Verlag. 2002, pp. 83–92. <http://citeseer.ist.psu.edu/kantschik02lineargraph.html>
31. *Kantschik W., Dittrich P., Brameier M., Banzhaf W.* Empirical Analysis of Different Levels of Meta-Evolution / *Proceedings of the Congress on Evolutionary Computation*. Washington D.C.: IEEE Press. 1999, pp. 2086–2093. <http://citeseer.ist.psu.edu/kantschik99empirical.html>
32. *Kantschik W., Dittrich P., Brameier M., Banzhaf W.* Meta-Evolution in Graph GP / *Genetic Programming: Second European Workshop (EuroGP'99)*. Berlin: Springer. 1999, pp. 15–28. <http://citeseer.ist.psu.edu/kantschik99metaevolution.html>
33. *Kinnear K.* *Advances in Genetic Programming*. Volume 1. Cambridge: The MIT Press, 1994. P. 532 P.
34. *Kovacina M., Palmer D., Vaidyanathan R.* Swarm Rule-Base Development using Genetic Programming Techniques. [http://www.nps.navy.mil/se/ravi/My%20Papers/Self\\_programming\\_swarm\\_WASP\\_03.pdf](http://www.nps.navy.mil/se/ravi/My%20Papers/Self_programming_swarm_WASP_03.pdf)
35. *Koza J.* Future work and practical applications of genetic programming / *Handbook of Evolutionary Computation*. New York: Oxford University Press. 1997. pp. H1.1:1–6. <http://www.genetic-programming.com/jkpdf/hecfuture1997.pdf>
36. *Koza J.* *Genetic Programming: On the programming of Computers by Means of Natural Selection*. Cambridge: The MIT Press, 1992. 840 P.
37. *Langdon W.* *Genetic Programming and Data Structures*. MA: Kluwer Academic Publishers, 1998. 292 P.

Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

38. *Lankhorst M.* A Genetic Algorithm for the Induction of Nondeterministic Pushdown Automata. Computing Science Report. University of Groningen Department of Computing Science. 1995.
39. *Lucas S.* Evolving Finite State Transducers: Some Initial Explorations / Genetic Programming: 6th European Conference (EuroGP'03). Berlin: Springer. 2003, pp. 241–257. <http://algoval.essex.ac.uk/rep/fst/EuroFST.pdf>
40. *Lucas S., Reynolds T.* Learning Finite-State Transducers: Evolution Versus Heuristic State Merging // IEEE Transactions on Evolutionary Computation. 2007. V. 11, I. 3, pp. 308–325. <http://algoval.essex.ac.uk/rep/fst/TecFST.pdf>
41. *Lutter B., Huntsinger R.* Engineering applications of finite automata // Simulation. 1969. V. 13, I. 1, pp. 5–11. <http://sim.sagepub.com/cgi/reprint/13/1/5>
42. *Miller J., Thomson P.* A Developmental method for growing Graphs and Circuits / Evolvable Systems: From Biology to Hardware. Berlin: Springer. 2003, pp. 93–104. <http://citeseer.ist.psu.edu/miller03developmental.html>
43. *Miller J.* The Coevolution of Automata in the Repeated Prisoner's Dilemma. Working Paper. Santa Fe Institute. 1989 // Journal of Economic Behavior & Organization. 1996. V. 29, I. 1, pp. 87–112.
44. *Ngom L., Baron C., Geffroy J.* Genetic Simulation for Finite State Machine Identification / Proceedings of the Thirty-Second Annual Simulation Symposium. Washington: IEEE Computer Society. 1999, p.118. <http://ieeexplore.ieee.org/iel5/6222/16607/00766462.pdf>
45. *Petrovic P.* Comparing Finite-State Automata Representation with GP-trees. Technical Report. Norwegian University of Science and Technology. 2006. <http://www.idi.ntnu.no/~petrovic/fsatr/fsatr.pdf>
46. *Petrovic P.* Evolving automats for distributed behavior arbitration. Technical Report. Norwegian University of Science and Technology. 2005. <http://www.idi.ntnu.no/~petrovic/shortpaper.pdf>
47. *Petrovic P.* Simulated evolution of distributed FSA behaviour-based arbitration. Poster at The Eighth Scandinavian Conference on Artificial Intelligence (SCAI'03). 2003. <http://www.idi.ntnu.no/grupper/ai/eval/incremental/scai03-poster-petrovic.pdf>
48. *Poli R.* Evolution of Graph-like Programs with Parallel Distributed Genetic Programming / Genetic Algorithms: Proceedings of the Seventh International Conference. MI: Morgan Kaufmann. 1997, pp. 346–353. <http://citeseer.ist.psu.edu/578015.html>
49. *Spears W., Gordon D.* Evolving Finite-State Machine Strategies for Protecting Resources / Proceedings of the 12th International Symposium on Foundations of Intelligent Systems. London: Springer-Verlag. 2000, pp. 166–175. <http://citeseer.ist.psu.edu/spears00evolving.html>
50. *Spector L., Langdon W., O'Reilly U., Angeline P.* Advances in Genetic Programming. Volume 3. Cambridge: MIT Press, 1999. 488 P. <http://www.cs.bham.ac.uk/~wbl/aigp3>
51. *Takeuchi A.* Evolutionary Automata – Comparison of Automaton Behavior and Restle's Learning Model // Information Sciences. 1980. V. 20, I. 2, pp. 91–99. [http://dx.doi.org/10.1016/0020-0255\(80\)90013-4](http://dx.doi.org/10.1016/0020-0255(80)90013-4)
52. *Teller A., Veloso M.* Internal reinforcement in a connectionist genetic programming approach // Artificial Intelligence. 2000. V. 120, I. 2, pp. 165–198. <http://citeseer.ist.psu.edu/teller00internal.html>
53. *Teller A., Veloso M.* PADO: A New Learning Architecture for Object Recognition / Symbolic Visual Learning. NY: Oxford University Press. 1996, pp. 81–116. <http://www.cs.cmu.edu/afs/cs/usr/astro/public/papers/PADO.ps>

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

## ПРИЛОЖЕНИЕ А

УТВЕРЖДАЮ  
Руководитель работ  
докт. техн. наук., профессор  
А. А. Шалыто  
31 мая 2007 г.

### ЗАДАНИЕ № 2007.05.31-01 на проведение патентных исследований

Наименование работы (темы): Технология генетического программирования для генерации автоматов управления системами со сложным поведением.

Шифр работы (темы): 2007-4-1.4-18-01-033.

Этап работы: определение уровня развития техники и патентоспособности.

Сроки выполнения: 01.06.2007 г. – 31.08.2007 г.

Задачи патентных исследований: определение уровня развития техники и патентоспособности разрабатываемой технологии, получение сведений об охраняемых и иных документах, которые будут препятствовать применению этих результатов на территории Российской Федерации.

### КАЛЕНДАРНЫЙ ПЛАН

Виды патентных исследований	Подразделения-исполнители (соисполнители)	Ответственные исполнители (Ф.И.О.)	Сроки выполнения патентных исследований. Начало. Окончание	Отчетные документы
Определение уровня развития техники и патентоспособности		Точилин В.Н.	01.06.2007 – 31.08.2007	Отчет о патентных исследованиях

Декан факультета «Информационные технологии и программирование»  
СПбГУ ИТМО  
докт. техн. наук, профессор

В. Г. Парфенов

Начальник отдела  
интеллектуальной собственности и  
научно-технической информации

Л. Н. Казар

Технология генетического программирования для генерации автоматов управления системами со сложным поведением.  
Отчет по патентным исследованиям за I этап «Выбор направления исследований и базовых компонентов»

## ПРИЛОЖЕНИЕ Б

### Регламент поиска № 2007.05.31-01

31 мая 2007 г.

Наименование работы (темы): Технология генетического программирования для генерации автоматов управления системами со сложным поведением

Шифр работы (темы): 2007-4-1.4-18-01-033.

Этап работы: определение уровня развития техники и патентоспособности.

Номер и дата утверждения задания: 2007.05.31-01, 31.05.2007 г.

Цель поиска информации: создание перечня документов, которые потенциально могут нарушить патентоспособность разрабатываемой технологии и препятствовать ее применению на территории Российской Федерации.

Начало поиска: 01 июня 2007 г.

Окончание поиска: 31 августа 2007 г.

Предмет поиска  (объект исследования, его составные части, товар)	Страна поиска	Источники информации, по которым будет проводиться поиск								Ретро-селективность	Наименование информационной базы (фонда)
		Патентные		НТИ*		конъюнктурные		другие			
		Наименование	Классификационные рубрики: МПК (МКИ)*, МКПО*, НКИ* и другие	Наименование	Рубрики УДК* и другие	Наименование	Код товара: ГС*, СМТК*, БТН*	Наименование	Классификационные индексы		
1	2	3	4	5	6	7	8	9	10	11	12
Автоматическое построение программ Имитация естественной эволюции Применение автоматных моделей вычислений Построение управляющих систем	Россия, США, Европа	База данных Федеральной службы по интеллектуальной собственности, патентам и товарным знакам Российской Федерации. Бюро по патентам и товарным знакам США. Европейского патентного бюро. Всемирной организации интеллектуальной собственности.	G06F 11/36  G06F 9/445V	-	-	-	-	-	-	20 лет	Роспатент, www.fips.ru USPTO, www.uspto.gov EPO, ep.espacenet.com ВОИС, www.wipo.int

Руководитель темы заведующий кафедрой  
«Технологий программирования»,  
докт. техн. наук, профессор

\_\_\_\_\_

подпись

А. А. Шалыто

Начальник отдела  
интеллектуальной собственности и  
научно-технической информации

\_\_\_\_\_

подпись

Л. Н. Казар

\*МПК (МКИ) — международная патентная классификация (международная классификация изобретений);  
НКИ — национальная классификация изобретений;  
МКПО — международная классификация промышленных образцов;  
НТИ — научно-техническая информация;  
ГС — гармонизированная система (гармонизированная товарная номенклатура);  
СМТК — стандартная международная торговая классификация ООН;  
БТН — Брюссельская таможенная номенклатура;  
УДК — универсальная десятичная классификация.

**ПРИЛОЖЕНИЕ В**  
**Отчёт о поиске № 2007.08.31-01**

УТВЕРЖДАЮ  
Руководитель работ  
докт. техн. наук., профессор  
А. А. Шалыто  
31 мая 2007 г.

Поиск проведен в соответствии с заданием на проведение патентных исследований № 2007.05.31-01 от 31 мая 2007 г. и регламентом поиска 2007.05.31-01 от 31 мая 2007 г.

Этап работы: определение патентоспособности разрабатываемой технологии генетического программирования для генерации автоматов, управляющих системами со сложным поведением, на первом этапе исследований по контракту.

Начало поиска – 01 июня 2007 г. Окончание поиска – 31 августа 2007 г.

Сведения о выполнении регламента поиска: поиск выполнен полностью. Документов, которые могут препятствовать применению разрабатываемой технологии на территории Российской Федерации, не найдено.

На дальнейших этапах работ по указанному контракту необходимо проведение дополнительных патентных исследований для выявления новых патентов и непатентных работ в рассматриваемой области.

Материалы, отобранные для последующего анализа: отсутствуют.

Декан факультета «Информационные  
технологии и программирование»  
СПбГУ ИТМО  
докт. техн. наук, профессор

В. Г. Парфенов

Начальник отдела  
интеллектуальной собственности и  
научно-технической информации

Л. Н. Казар