


На правах рукописи



Чивилихин Даниил Сергеевич

Генерация конечных автоматов на основе муравьиных алгоритмов

Специальность 05.13.11 — Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург — 2015

Работа выполнена в Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики.

Научный руководитель: доктор технических наук, профессор
Шалыто Анатолий Абрамович

Официальные оппоненты: **Дубинин Виктор Николаевич**,
доктор технических наук, доцент, профессор кафедры вычислительной техники Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Пензенский государственный университет»

Тулупьев Александр Львович,
доктор физико-математических наук, доцент, заведующий лабораторией теоретических и междисциплинарных проблем информатики Федерального государственного бюджетного учреждения науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук

Ведущая организация: Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»

Защита состоится 24 декабря 2015 г. в 17 часов 00 минут на заседании диссертационного совета Д 212.227.06 при Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики по адресу: 197101, г. Санкт-Петербург, Кронверкский просп., д. 49., ауд. 431.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики по адресу: 197101, г. Санкт-Петербург, Кронверкский просп., д. 49. и на сайте frro.ifmo.ru.

Автореферат разослан 10 ноября 2015 года.

Ученый секретарь
диссертационного совета

Д 212.227.06, к.ф.-м.н., доцент



Холодова Светлана Евгеньевна

Общая характеристика работы

Актуальность и степень разработанности проблемы. В последнее время для реализации управления объектами со сложным поведением все чаще применяется *автоматное программирование*. Это парадигма программирования, в которой предлагается реализовывать программы в виде *автоматизированных объектов управления*, каждый из которых состоит из *системы управления*, представленной одним или несколькими *управляющими конечными автоматами*, и *объекта управления*. Основными достоинствами автоматного программирования являются наглядность описания поведения программ, возможность автоматической генерации кода и высокий уровень автоматизации верификации автоматных программ с помощью метода *проверки моделей (model checking)*. Идеи автоматного программирования используются, например, в средах разработки *MATLAB/Stateflow*, *IBM Rational Rhapsody*, а также в стандартах разработки приложений для промышленной автоматизации, например *IEC 61131* и *IEC 61499*.

В некоторых случаях конечные автоматы для прикладных задач удается построить эвристически, однако известны примеры, когда построенные вручную автоматы неоптимальны, или их вовсе не удается построить. Это указывает на актуальность повышения степени автоматизации процесса разработки автоматных программ.

В последнее время проводится все больше исследований в области *поисковой инженерии программного обеспечения (search-based software engineering)*. Это новая, стремительно развивающаяся область исследований, по которой на сентябрь 2015 года насчитывается всего лишь 1389 публикаций. При этом до 1992 года было опубликовано менее десяти работ, а в 2014 году вышло более 200 научных статей. В рамках этого подхода для автоматизированного решения вычислительно сложных задач, возникающих при разработке программного обеспечения, применяются *метаэвристические* алгоритмы *поисковой оптимизации*, позволяющие в большинстве случаев находить близкие к оптимальным решения. Примерами метаэвристик являются эволюционные и генетические алгоритмы, эволюционные стратегии и муравьиные алгоритмы. Они применяются, поскольку реализуют неполный направленный перебор решений в *пространстве поиска* – множестве всех возможных решений задачи. Обычно поиск начинается со случайного решения, которое постепенно улучшается путем внесения в него небольших случайных изменений. Для оценки степени соответствия решения условиям задачи применяется так называемая *функция приспособленности (ФП)*.

Большинство существующих методов генерации конечных автоматов основано либо на использовании примеров поведения (сценариев или тестов), либо на моделировании, которое позволяет тестировать автоматы. Применение этих подходов не дает никаких гарантий относительно поведения сгенерированных автоматов на данных, не использованных для их генерации. Поэтому в последнее время получили развитие методы, в которых наряду со сценария-

ми или тестами используется проверка *темпоральных свойств*, позволяющих задать более общие закономерности поведения. Эти свойства задаются с помощью формул на языках темпоральной логики, например *Linear Temporal Logic (LTL)*. Такие методы гарантируют, что поведение автомата на любых входных данных будет соответствовать заданным примерам поведения и темпоральным формулам, но не более того. При этом известно, что задача генерации конечного автомата, удовлетворяющего *LTL*-формуле, является *2EXPTIME*-полной, а число состояний сгенерированного автомата в худшем случае дважды экспоненциально зависит от длины формулы. Эти оценки также можно распространить на случай сценариев, так как их можно записать в виде *LTL*-формул.

Известные методы генерации конечных автоматов с учетом темпоральных формул чаще всего основаны на генетических алгоритмах и так называемой *AE*-парадигме. Методы первой группы на основе генетических алгоритмов позволяют генерировать автоматы по примерам поведения с учетом заданных *LTL*-формул. Недостатком этих методов является то, что для нахождения решения может потребоваться очень много времени.

Вторая группа методов основана на *AE*-парадигме, в которой программа с входным сигналом x и выходным сигналом y , удовлетворяющая темпоральной формуле $\varphi(x, y)$, конструируется как побочный продукт доказательства теоремы $(\forall x)(\exists y)\varphi(x, y)$. Название парадигмы происходит от английских терминов для кванторов \forall (*for All*) и \exists (*Exists*). Методы, основанные на этой парадигме, являются *точными* в том смысле, что позволяют найти решение, если оно существует, или доказать, что решения нет. В силу высокой вычислительной сложности эти методы зачастую работают дольше генетических алгоритмов. Еще одним их недостатком является большое число состояний генерируемых автоматов. Поэтому исследования, направленные на развитие методов генерации конечных автоматов, лишенных указанных недостатков, являются **актуальными**.

Дополнительной сложностью при применении метаэвристических алгоритмов для генерации конечных автоматов является «плохой» *ландшафт функции приспособленности*. В простейшем случае двухпараметрического пространства поиска, ландшафт ФП – поверхность, задающая график ФП. Например, диссертантом были изучены свойства ландшафта ФП в задаче о построении автомата, управляющего агентом в некоторой игре на тороидальном поле. Эта задача является одной из классических модельных задач для изучения свойств метаэвристических алгоритмов. Было обнаружено, что решения, незначительно отличающиеся от оптимального, обладают большим разбросом значений ФП. Фактически, это означает, что метаэвристический алгоритм может найти оптимальное решение лишь случайно. Это наблюдение указывает на то, что метаэвристический алгоритм генерации конечных автоматов должен производить активный поиск в окрестности субоптимальных решений. Одним из вариантов реализации такого подхода является использование долговременной общей памяти по аналогии с *муравьиными алгоритмами*.

Муравьиные алгоритмы – это семейство метаэвристик для решения задач оптимизации на графах, инспирированных наблюдениями за поведением муравьев в процессе поиска пищи. Отличительной особенностью муравьиных алгоритмов является использование отдельными муравьями-агентами *общей памяти* об исследованной за время работы алгоритма области пространства поиска. Отметим, что ранее муравьиные алгоритмы для генерации автоматов не использовались. Настоящая работа направлена на *разработку более эффективных по сравнению с существующими методов генерации конечных автоматов с учетом темпоральных формул* на основе муравьиных алгоритмов.

В соответствии с паспортом специальности 05.13.11 – «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей», настоящая диссертация относится к области исследований «1. Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования».

Целью работы является развитие существующих подходов к генерации конечных автоматов по сценариям работы и темпоральным формулам за счет применения муравьиных алгоритмов.

Задачи диссертационной работы состоят в следующем.

1. Разработать метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма.
2. Разработать методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов.
3. Разработать инструментальное средство, реализующее предложенные методы.
4. Внедрить результаты работы при генерации автоматной логики для базисных функциональных блоков стандарта *IEC 61499* и в учебный процесс.

Научная новизна. В работе получены следующие новые научные результаты, которые выносятся на защиту.

1. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам, основанный на муравьином алгоритме, отличающемся от существующих тем, что в нем используется предложенный автором граф мутаций. Показано, что предложенный метод работает быстрее известных методов на основе генетических алгоритмов и *АЕ*-парадигмы.
2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам, совмещающий параллельный муравьиный алгоритм, точный метод генерации конечных автоматов по сценариям работы на основе решения задачи выполнимости булевой формулы и процедуру прореживания сценариев. Показано, что метод работает быстрее параллельного генетического алгоритма и параллельного муравьиного алгоритма.

Методология и методы исследования. Методологическая основа диссертации – итеративная генерация автоматов с последующим отбором и ранжированием по степени соответствия входным данным. В работе используются методы теории автоматов, эволюционных вычислений, темпоральной логики, дискретной математики и математической статистики.

Достоверность научных положений и выводов, полученных в диссертации, подтверждается корректным обоснованием постановок задач, точной формулировкой критериев, результатами экспериментов и их статистическим анализом.

Теоретическое значение работы состоит в том, что показана применимость муравьиных алгоритмов для решения задач, в которых пространство поиска не имеет естественного графового представления.

Практическое значение работы состоит в том, что разработанные методы генерации конечных автоматов реализованы в виде инструментального программного средства, позволяющего генерировать автоматы существенно быстрее известных методов.

Внедрение результатов работы.

Результаты диссертации были внедрены в Технологическом университете Лулео (Швеция) при генерации автоматной логики для базисных функциональных блоков стандарта *IEC 61499* и использованы в учебном процессе в Университете ИТМО на кафедре «Компьютерные технологии» в рамках курсов «Теория автоматов и программирование» и «Генетическое программирование».

Апробация результатов работы. Основные результаты работы докладывались на 14 конференциях: III-я и VI-я Всероссийская конференция по проблемам информатики СПИСОК (2012, 2013, Матмех СПбГУ), 14th, 15th and 16th Genetic and Evolutionary Computation Conference (2012, Филадельфия, 2013, Амстердам, 2014, Ванкувер), 8th International Conference on Swarm Intelligence (2012, Брюссель); VII-я Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (2013, Коломна), 7th IFAC Conference on Manufacturing Modelling, Management, and Control (2013, Санкт-Петербург), 12th and 13th International Conference on Machine Learning and Applications (2013, Майами, 2014, Детройт), 1st BRICS countries Congress on Computational Intelligence (2013, Ресифи), 6th International Student Workshop on Bioinspired Optimization Methods and their Applications (2014, Любляна), XII-е Всероссийское совещание по проблемам управления (2014, Москва), 13th IEEE International Conference on Industrial Informatics (2015, Кембридж) и 13th IEEE International Symposium on Parallel and Distributed Processing with Applications (2015, Хельсинки).

Личный вклад автора. Решение задач диссертации, разработанные методы, алгоритмы и их реализация принадлежат лично автору.

Публикации. Основные результаты по теме диссертации изложены в 18 публикациях, три из которых изданы в российских журналах, рекомендован-

ных ВАК, 11 – в изданиях, индексируемых в международных базах цитирования *Web of Science* и *Scopus*. Доля диссертанта в работах, выполненных в соавторстве, указана в списке публикаций.

Свидетельства о регистрации программы для ЭВМ. Автором по теме диссертации получено два свидетельства о регистрации программы для ЭВМ.

Участие в научно-исследовательских работах. Результаты диссертации были получены при выполнении научно-исследовательских работ по следующим темам: «Разработка методов автоматизированного построения надежного программного обеспечения на основе автоматного подхода по обучающим примерам и темпоральным свойствам» (Грант РФФИ № 14-07-31337 мол_a) и «Разработка муравьиных алгоритмов для построения управляющих конечных автоматов» (Грант РФФИ № 14-01-00551 А). Автор является победителем конкурса грантов Санкт-Петербурга для студентов, аспирантов, молодых ученых, молодых кандидатов наук 2013 г., тема проекта — «Разработка методов генерации управляющих конечных автоматов на основе муравьиных алгоритмов», а также стипендиатом Президента РФ для аспирантов, обучающихся по приоритетным направлениям модернизации и технологического развития экономики России на 2015/16 учебный год.

Объем и структура работы. Диссертация состоит из введения, пяти глав, заключения и двух приложений. Объем диссертации составляет 144 страницы, с 33 рисунками, 6 таблицами и 9 листингами. Список литературы содержит 147 наименований.

Содержание работы

В первой главе приводится обзор работ по автоматному программированию, метаэвристическим алгоритмам оптимизации и методам генерации конечных автоматов. На основе результатов обзора формулируются задачи, решаемые в диссертации.

Автоматное программирование предполагает особый способ разработки программ, в которых есть *сущности со сложным поведением*. Говорят, что сущность обладает *простым поведением*, если она всегда одинаково реагирует на одинаковые входные воздействия. Если же сущность обладает сложным поведением, то ее реакции на одинаковые входные воздействия могут различаться и зависеть не только от самих входных воздействий, но и от предыстории (предыдущего состояния).

Одним из достоинств автоматного программирования является высокий уровень автоматизации верификации автоматных программ с помощью метода *проверки моделей*. Верификация на основе этого метода позволяет проверить, удовлетворяет ли модель программы заданным *темпоральным свойствам* – спецификации. Темпоральные свойства описывают временные связи между состояниями программы и задаются в виде формул на *языках темпоральной логики*, например, на языке *LTL*, который используется в настоящей работе.

При верификации автоматных программ основной сложностью является верификация их ядра – автоматов, генерация которых рассматривается в диссертации. Построение формальной модели автомата и проверяемых свойств может быть выполнено автоматически. Особенностью автоматов является то, что преобразование контрпримера из терминов модели в термины автомата также может быть выполнено автоматически. Это привело к развитию методов, позволяющих по примерам поведения и темпоральным формулам автоматически сгенерировать удовлетворяющий им автомат. Ввиду высокой теоретической сложности задач генерации автоматов при этом зачастую используются *метаэвристические алгоритмы* комбинаторной оптимизации, позволяющие в большинстве случаев находить близкие к оптимальным решения.

Лишь немногие методы генерации конечных автоматов позволяют синтезировать автоматы, удовлетворяющие заданным темпоральным формулам. Два метода основаны на генетических алгоритмах. При каждом вычислении значения ФП с помощью верификатора оценивается степень соответствия решения темпоральным формулам. Недостатком этих методов является то, что для нахождения решения может потребоваться очень много времени.

Также известны точные методы генерации конечных автоматов по темпоральным формулам, позволяющие найти решение, если оно существует, либо доказать, что решения нет. Эти методы основаны на так называемой *AE-парадигме*. В силу высокой вычислительной сложности этих методов, они зачастую работают дольше генетических алгоритмов. Еще одним их недостатком является большое число состояний генерируемых автоматов.

Приведем постановку задачи генерации конечных автоматов по сценариям работы и темпоральным формулам, которая решается в диссертации. Сценарий работы представляет собой последовательность троек, включающих входное событие, охранное условие (булеву формулу от входных переменных) и последовательность выходных воздействий. Темпоральные формулы задаются на языке *LTL*. Задача состоит в генерации автомата с заданным числом состояний, удовлетворяющего заданным сценариям работы и *LTL*-формулам.

Во **второй главе** описывается предлагаемый метод *MuACO* (*Mutation-based Ant Colony Optimization*) генерации конечных автоматов по сценариям работы и темпоральным формулам, основанный на новом муравьином алгоритме, использующем предложенный автором *граф мутаций*. В классических муравьиных алгоритмах применяется граф конструирования, вершины которого ассоциированы с компонентами решений, а полное решение задачи соответствует пути в графе. В методе *MuACO* используется граф мутаций, вершинами которого являются конечные автоматы, а ребрам соответствуют *мутации* автоматов – их небольшие изменения. На каждом ребре графа мутаций хранятся значения *эвристической информации* и *феромона*.

В качестве первой вершины графа мутаций используется решение, которое генерируется случайным образом или подается алгоритму на вход. Алгоритм работает итеративно, на каждой итерации выполняется построение

новых автоматов с помощью колонии муравьев и обновление значений феромона.

На этапе построения новых решений все муравьи начинают поиск решений из вершины графа, ассоциированной с лучшим решением-кандидатом. На каждом шаге муравей находится в вершине u и выполняет переход в другую вершину v , которая выбирается одним из двух способов. В первом из них муравей генерирует несколько новых решений путем мутации своего текущего решения и переходит в лучшее из новых решений. Во втором – следующая вершина определяется с помощью метода рулетки на основе значений феромона и эвристической информации на ребрах, инцидентных текущей вершине. Муравей перемещается по графу до тех пор, пока число шагов, на которых не было увеличено текущее значение ФП, не достигнет заданного ограничения. Затем происходит обновление значений феромона на всех ребрах графа мутаций: уменьшение значений феромона вследствие испарения и увеличение значений на тех ребрах, которые были посещены муравьями.

После этого излагается комбинированный метод $SAT+MuACO$, в котором начальное решение для $MuACO$ генерируется *только по сценариям работы* с помощью точного метода $efsmSAT$ на основе решения задачи выполнимости (SAT).

Предлагаемые методы $MuACO$ и $SAT+MuACO$ экспериментально сравниваются с генетическим алгоритмом. Для выбора значений параметров методов применяется свободно распространяемое программное средство для автоматической настройки параметров алгоритмов *irace*. Для проведения сравнения используются примеры задачи на основе случайно сгенерированных автоматов. На рисунке 1 приведены ящичные диаграммы распределений времени работы методов для различных значений числа состояний автомата. Метод $MuACO$ работает существенно быстрее генетического алгоритма, а метод $SAT+MuACO$ – быстрее, чем $MuACO$. Статистическая значимость результатов подтверждается тестом Уилкоксона. Также проводится сравнение с точными методами генерации автоматов по LTL -формулам на основе AE -парадигмы, реализованными в программных средствах *lily*, *unbeast* и *G4LTL_ST*. На примере генерации автоматов для управления дверьми лифта показывается, что предложенный в диссертации метод $MuACO$ работает быстрее этих методов.

В **третьей главе** описываются предлагаемые методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов. В предлагаемом методе $pMuACO$ (*parallel MuACO*) в каждом из m доступных потоков запускается алгоритм $MuACO$. Алгоритмы в отдельных потоках взаимодействуют с помощью генетической операции кроссовера, а также посредством архива лучших решений – при перезапуске i -го алгоритма в качестве начального решения используется лучшее решение j -го алгоритма.

Метод $psMuACO$ (*parallel SAT MuACO*) отличается от $pMuACO$ тем, что в качестве начального приближения во всех потоках используется автомат,

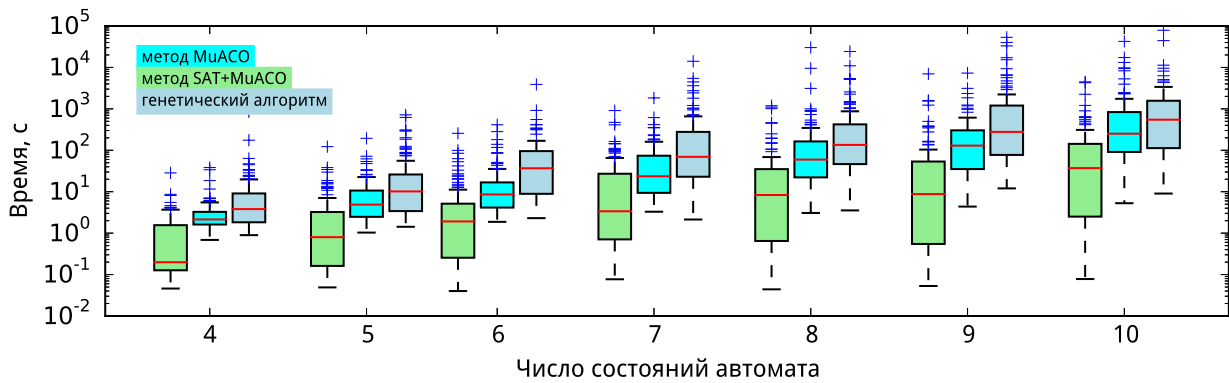


Рисунок 1 – Ящичные диаграммы распределений запусков методов *MuACO*, *SAT+MuACO* и генетического алгоритма по времени работы

сгенерированный *только по сценариям работы* с помощью точного метода *efsmSAT*. Предлагаемый метод *pstMuACO* (*parallel SAT thin-out MuACO*) совмещает точный метод генерации конечных автоматов по сценариям работы *efsmSAT*, процедуру прореживания сценариев и метод *pMuACO*. Метод включает следующие этапы.

Процедура прореживания сценариев. По набору сценариев T строится n_{start} различных автоматов. Для этого сначала строится n_{start} сокращенных наборов сценариев $T'_0, \dots, T'_{n_{\text{start}}-1}$. Каждый такой набор получается из T путем прореживания – удаления каждого сценария с вероятностью p_{thin} .

Построение начальных решений по сценариям с помощью метода efsmSAT. Каждый i -й автомат строится по сокращенному набору сценариев T'_i с помощью метода *efsmSAT*.

Генерация автомата с помощью pMuACO. Если $n_{\text{start}} = m$, автомат, построенный по набору T'_i , используется в качестве начального решения для i -го потока алгоритма. Если же $n_{\text{start}} < m$, то начальное решение в каждом потоке выбирается случайным образом из множества сгенерированных начальных решений.

Схема предлагаемого метода *pstMuACO* для случая, когда $m = n_{\text{start}}$, приведена на рисунке 2.

Приводится описание экспериментов по сравнению генетического алгоритма и методов *pMuACO*, *psMuACO* и *pstMuACO*. Ящичные диаграммы распределений времени работы этих методов для автоматов из 10, 15 и 20 состояний приведены на рисунке 3. Метод *pstMuACO* существенно быстрее, чем *psMuACO*, который, в свою очередь, быстрее метода *pMuACO*. Для автоматов из 10 и 15 состояний метод *pMuACO* быстрее генетического алгоритма, при этом 16 запусков генетического алгоритма не завершились за шесть часов. Поэтому для 20 состояний генетический алгоритм не запускался. Различия во времени работы методов являются статистически значимыми, что подтверждается результатами проведения теста Уилкоксона.

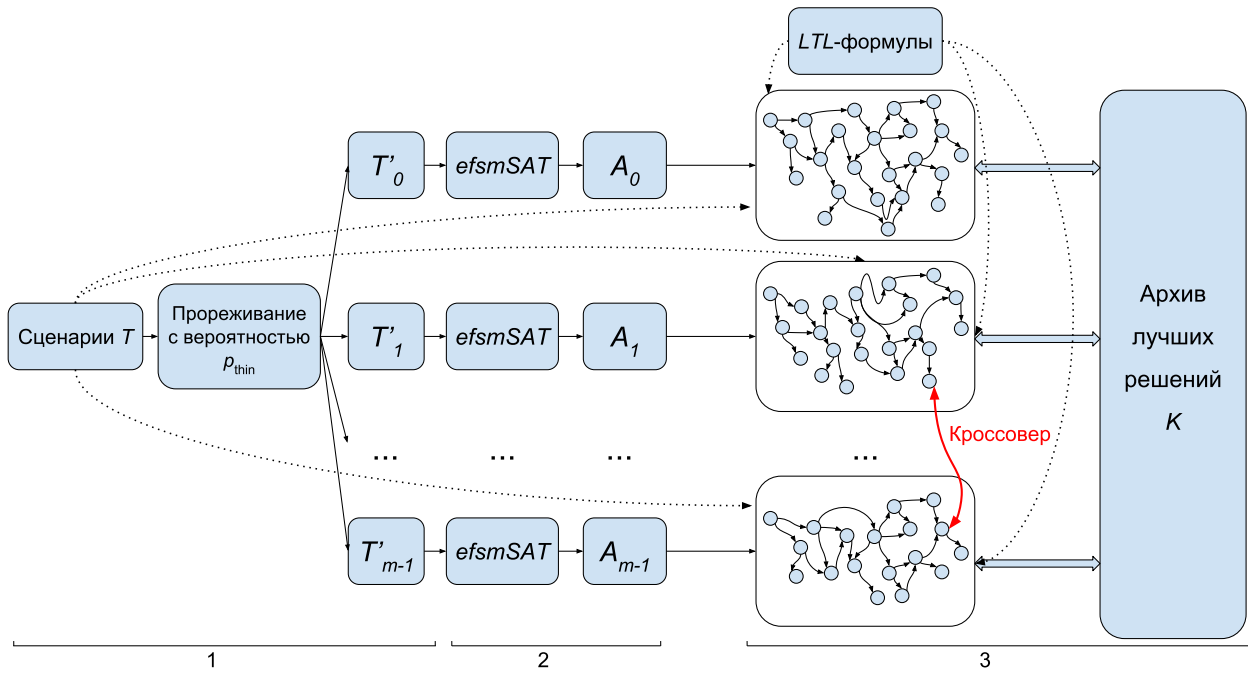


Рисунок 2 – Схема предлагаемого метода $pstMuACO$ ($m = n_{start}$)

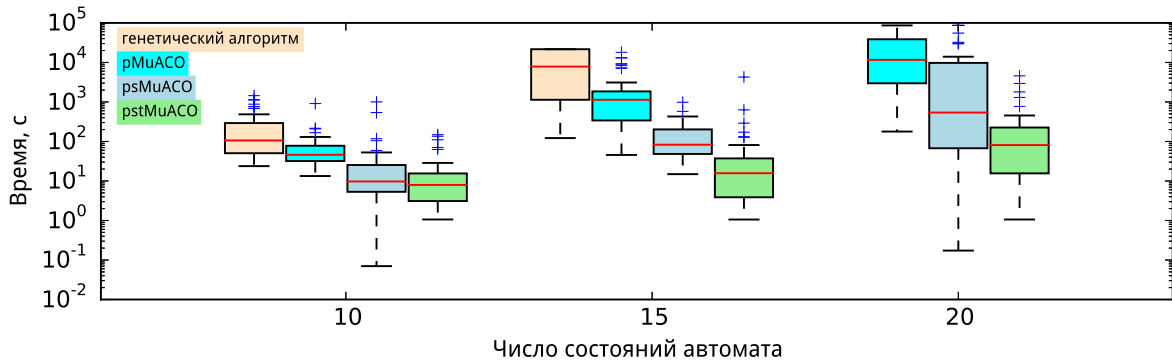


Рисунок 3 – Ящичные диаграммы распределений времени работы методов $pMuACO$, $psMuACO$, $pstMuACO$ и генетического алгоритма

Четвертая глава содержит описание разработанного инструментального средства *tuaco.jar*, реализующего предложенные в диссертации методы, исходный код которого размещен в открытом доступе в сети Интернет по адресу https://bitbucket.org/chivilikhin_daniil/aco-for-automata. Средство позволяет генерировать автоматы по сценариям работы и *LTL*-формулам, заданным в текстовом виде, и выдает на выход автомат в формате *GraphViz*. Также разработанное средство может быть использовано в качестве библиотеки для решения других задач генерации автоматов, что демонстрируется на примере задачи, решаемой в пятой главе.

В **пятой главе** описывается внедрение предложенных в диссертации методов при генерации автоматной логики для базисных функциональных блоков стандарта *IEC 61499*, который определяет открытую архитектуру разработки событийных систем распределенного и централизованного управления

и автоматизации. Элементарным компонентом стандарта *IEC 61499* является *функциональный блок (ФБ)*, характеризующийся *интерфейсом*, который определяет используемые входные/выходные события и входные/выходные переменные. *Базисный ФБ* задается с помощью событийной модели, называемой *диаграммой управления выполнением (ДУВ)*, и представляющей собой конечный автомат Мура специального вида. Каждое состояние ДУВ ассоциировано с выходным воздействием – выполнением алгоритма и генерацией выходного события. *Составной ФБ* задается сетью других ФБ, базисных или составных.

Результаты диссертации внедрены для решения задачи восстановления неизвестной ДУВ базисного ФБ, использующего только логические входные и выходные переменные, при известном интерфейсе. Эта задача возникает в случаях, когда исходный код ФБ утерян или нет инженеров, способных быстро в нем разобраться. Общая схема разработанного подхода восстановления ДУВ представлена на рисунке 4.

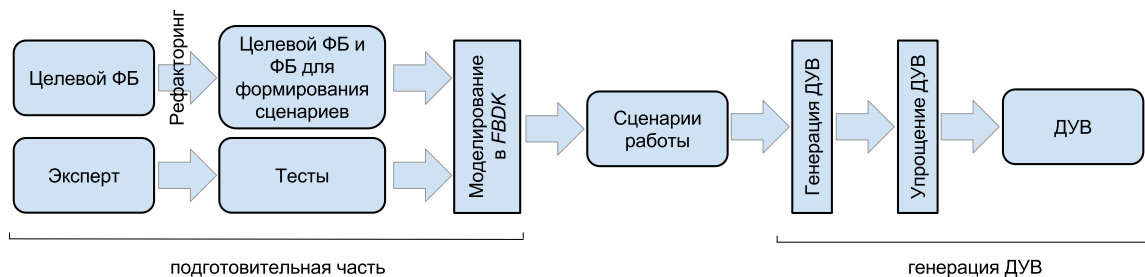


Рисунок 4 – Схема разработанного подхода восстановления ДУВ

Сначала для осуществления возможности формирования сценариев работы автоматически выполняется рефакторинг сети, содержащей целевой ФБ. После этого путем исполнения приложения в открытой среде *FBDK* разработки приложений *IEC 61499* формируется набор сценариев работы. Полученные сценарии подаются на вход методу $rMuACO_{ecc}$, который генерирует удовлетворяющую им ДУВ. Метод $rMuACO_{ecc}$ – это предложенный в диссертации метод $rMuACO$, параметризованный специальными операторами мутации и способом представления ДУВ. Диаграмма управления выполнением представляется в виде множества состояний, где в каждом состоянии для каждого входного события хранится множество групп переходов. Для каждой группы переходов задан логический массив, называемый *маской значимости* и таблица переходов. Каждый j -й элемент таблицы хранит номер состояния, в которое нужно перейти, когда двоичное число, составленное из значений входных переменных, равно 2^j . Если переход отсутствует, то в ячейке таблицы хранится значение *null*. Реализовано четыре оператора мутации: изменение состояния, в которое ведет переход, добавление/удаление переходов, изменение маски значимости группы переходов и изменение множества значимых переменных группы переходов. Для специального случая, когда каждый логический алгоритм используется ровно в одном состоянии ДУВ, разработан полиномиальный алгоритм генерации ДУВ.

Было успешно проведено восстановление ДУВ блока, осуществляющего *централизованное* управление элементами системы *Pick-and-Place* манипулятора (рисунок 5). Были получены ДУВ, достаточно близкие к исходной и обладающие эквивалентным с ней поведением на заданном наборе сценариев. Одна из сгенерированных ДУВ приведена на рисунке 6. Отметим, что предложенный подход также может быть применен для восстановления каждого отдельного элемента системы *распределенного* управления.

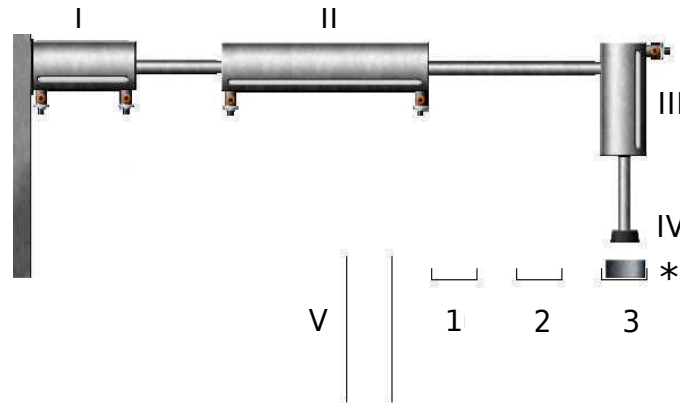


Рисунок 5 – *Pick-and-Place* манипулятор

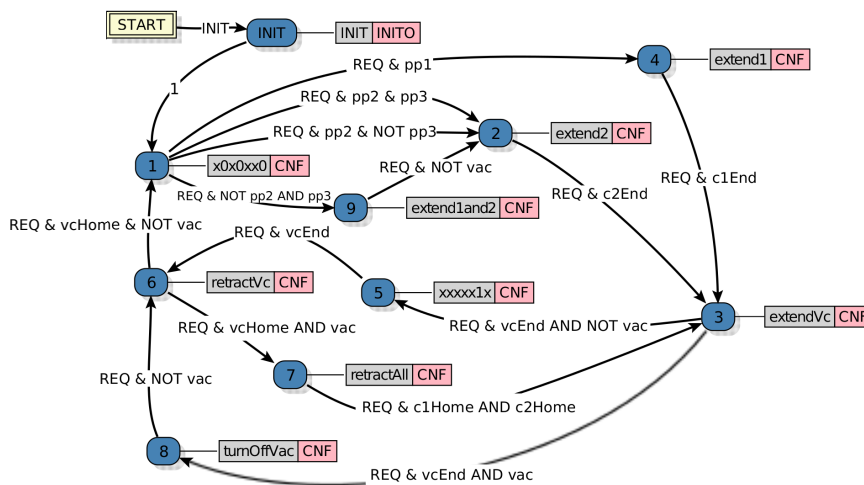


Рисунок 6 – Одна из сгенерированных ДУВ

Заключение

В диссертации получены следующие результаты.

1. Предложен метод *MuACO* генерации конечных автоматов по сценариям работы и темпоральным формулам, основанный на муравьином алгоритме с предложенным автором графом мутаций. Показано, что метод работает быстрее известных методов на основе генетических алгоритмов и *AE*-парадигмы.

2. Предложены методы *pMuACO*, *psMuACO* и *pstMuACO* генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов. Последний из них, *pstMuACO*, совмещает параллельный муравьиный алгоритм *pMuACO*, точный метод *efsmSAT* генерации конечных автоматов по сценариям работы на основе сведения к задаче выполнимости булевой формулы и процедуру прореживания сценариев. Показано, что метод *pstMuACO* работает быстрее параллельного генетического алгоритма, а также методов *pMuACO* и *psMuACO*.
3. Разработано программное средство *tuaco.jar*, реализующее предложенные методы, исходный код которого размещен в открытом доступе в сети Интернет. Средство позволяет генерировать автоматы по сценариям работы и темпоральным формулам, а также может быть использовано для решения других задач генерации автоматов.
4. Результаты диссертации были внедрены в Технологическом университете Лулео (Швеция) при генерации автоматной логики для базисных функциональных блоков стандарта *IEC 61499* [11, 12], имеется акт внедрения. Также результаты были использованы в учебном процессе на кафедре «Компьютерные технологии» Университета ИТМО в рамках курсов «Теория автоматов и программирование» и «Генетическое программирование», имеется акт использования. Наконец, некоторые результаты диссертации были использованы при генерации конечных автоматов для управления моделью беспилотного самолета [3].

Полученные результаты могут быть использованы для решения задач генерации конечных автоматов. В качестве направлений дальнейшей разработки темы диссертации можно выделить доработку предложенных методов для генерации ДУВ базисных ФБ стандарта *IEC 61499* с учетом темпоральных формул, а также целочисленных и вещественных входных/выходных переменных.

Статьи в журналах из перечня ВАК

1. Построение автоматных программ по спецификации с помощью муравьиного алгоритма на основе графа мутаций / Д. С. Чивилихин, В. И. Ульянцев, В. В. Вяткин, А. А. Шалыто // Научно-технический вестник информационных технологий, механики и оптики. — 2014. — 6(94). — С. 98—105. — 0,5 п. л. / 0,2 п. л.
2. Чивилихин Д. С., Ульянцев В. И. Метод построения управляющих автоматов на основе муравьиных алгоритмов // Научно-технический вестник информационных технологий, механики и оптики. — 2012. — 6(82). — С. 72—76. — 0,3125 п. л. / 0,16 п. л.

3. Генерация управляющих конечных автоматов по обучающим примерам на основе муравьиного алгоритма / И. П. Бужинский, В. И. Ульянов, Д. С. Чивилихин, А. А. Шалыто // Известия РАН. Теория и системы управления. — 2014. — № 2. — С. 111–121. — 0,6875 п. л. / 0,17 п. л.

Публикации в рецензируемых изданиях, индексируемых Web of Science или Scopus

4. *Chivilikhin D., Ulyantsev V., Shalyto A.* Combining Exact and Metaheuristic Techniques for Learning Extended Finite-State Machines from Test Scenarios and Temporal Properties // Proceedings of the 13th International Conference on Machine Learning and Applications. — IEEE Computer Society, 2014. — P. 350–355. — 0,375 п. л. / 0,15 п. л.
5. *Chivilikhin D., Ulyantsev V.* Learning Finite-State Machines: Conserving Fitness Function Evaluations by Marking Used Transitions // Proceedings of the 12th International Conference on Machine Learning and Applications. Vol. 2. — IEEE Computer Society, 2013. — P. 90–95. — 0,375 п. л. / 0,1875 п. л.
6. *Chivilikhin D., Ulyantsev V.* MuACOsm: a new mutation-based ant colony optimization algorithm for learning finite-state machines // Proceedings of the 15th Genetic and Evolutionary Computation Conference. — ACM, 2013. — P. 511–518. — 0,5 п. л. / 0,25 п. л.
7. *Chivilikhin D., Ulyantsev V., Shalyto A.* Solving five instances of the artificial ant problem with ant colony optimization // Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control. — IFAC / Elsevier, 2013. — P. 1043–1048. — 0,375 п. л. / 0,15 п. л.
8. *Chivilikhin D., Ulyantsev V.* Learning Finite-State Machines with Classical and Mutation-Based Ant Colony Optimization: Experimental Evaluation // Proceedings of the 1st BRICS countries Congress on Computational Intelligence. — IEEE Computer Society, 2013. — P. 528–533. — 0,375 п. л. / 0,1875 п. л.
9. *Chivilikhin D., Ulyantsev V., Tsarev F.* Test-based Extended Finite-State Machines Induction with Evolutionary Algorithms and Ant Colony Optimization // Proceedings of the 14th Genetic and Evolutionary Computation Conference companion. — ACM, 2012. — P. 603–606. — 0,25 п. л. / 0,1 п. л.
10. *Chivilikhin D., Ulyantsev V.* Learning Finite-State Machines with Ant Colony Optimization // Swarm Intelligence. — Springer Berlin / Heidelberg, 2012. — P. 268–275. — (Lecture Notes in Computer Science ; 7461). — 0,25 п. л. / 0,125 п. л.
11. Reconstruction of Function Block Logic using Metaheuristic Algorithm: Initial Explorations / D. Chivilikhin, A. Shalyto, S. Patil, V. Vyatkin // Proceedings of the 13th IEEE International Conference on Industrial Informatics. — IEEE Computer Society, 2015. — P. 1239–1242. — 0,25 п. л. / 0,1 п. л.

12. *Chivilikhin D., Shalyto A., Vyatkin V.* Inferring Automata Logic From Manual Control Scenarios: Implementation in Function Blocks // Proceedings of the 13th IEEE International Symposium on Parallel and Distributed Processing with Applications. — IEEE Computer Society, 2015. — P. 307–312. — 0,375 п. л. / 0,1875 п. л.
13. *Chivilikhin D., Ulyantsev V.* Inferring Automata-Based Programs from Specification With Mutation-Based Ant Colony Optimization // Proceedings of the 16th Genetic and Evolutionary Computation Conference companion. — ACM, 2014. — P. 67–68. — 0,125 п. л. / 0,062 п. л.
14. *Chivilikhin D., Ulyantsev V., Shalyto A.* Extended Finite-State Machine Inference With Parallel Ant Colony Based Algorithms // Proceedings of the 6th International Student Workshop on Bioinspired Optimization Methods and Their Applications. — Jozef Stefan Institute, 2014. — P. 117–126. — 0,281 п. л. / 0,093 п. л.

Другие публикации

15. *Чивилихин Д. С., Ульянцев В. И., Шалыто А. А.* Муравьиный алгоритм для построения автоматных программ по спецификации // XII Всероссийское совещание по проблемам управления ВСПУ-2014. — М. : Институт проблем управления им. В.А. Трапезникова РАН, 2014. — С. 4531–4542. — 0,75 п. л. / 0,25 п. л.
16. *Чивилихин Д. С., Ульянцев В. И.* Применение муравьиных алгоритмов для построения конечных автоматов // Всероссийская научная конференция по проблемам информатики СПИСОК. — СПб. : ВВМ. СПбГУ, 2012. — С. 409–410. — 0,062 п. л. / 0,031 п. л.
17. *Чивилихин Д. С., Ульянцев В. И.* Метод построения конечных автоматов на основе муравьиного алгоритма // Всероссийская научная конференция по проблемам информатики СПИСОК. — СПб. : ВВМ. СПбГУ, 2013. — С. 517–524. — 0,25 п. л. / 0,125 п. л.
18. *Чивилихин Д. С., Ульянцев В. И., Шалыто А. А.* Метод построения конечных автоматов на основе муравьиного алгоритма // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник тезисов докладов VII-й Международной научно-технической конференции. — М. : Физматлит, 2013. — С. 931–942. — 0,375 п. л. / 0,125 п. л.