Software & Systems
2023, vol. 36, no. 4

Print ISSN 0236-235X
Online ISSN 2311-2735

Программные продукты и системы

DOI 10.15827/0236-235X

С Новым 2024 годом!

№ 4
том 36
2023

Научно-исследовательский институт
«Центрпрограммсистем»

Программные продукты и системы

НАУЧНЫЙ ЖУРНАЛ

2023, том 36, № 4
(год издания тридцать шестой)

Главный редактор
Г.И. САВИН, академик РАН

SOFTWARE & SYSTEMS

Research Journal

2023, vol. 36, no. 4

Editor-in-Chief
G.I. SAVIN, Academician of the Russian Academy of Sciences

Research Institute CENTERPROGRAMSYSTEM

© ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ
Научный журнал

2023. Т. 36. № 4
DOI: 10.15827/0236-235X.142

Издатель НИИ «Центрпрограммсистем»
(г. Тверь, Россия)

Учредитель В.П. Куприянов

Журнал зарегистрирован в Роскомнадзоре 3 марта 2020 г.

Регистрационное свидетельство ПИ № ФС 77-77843

Подписной индекс в каталоге Урал-Пресс **70799**

ISSN 0236-235X (печатн.)
ISSN 2311-2735 (онлайн)

Главный редактор

Г.И. САВИН, академик РАН

Научные редакторы номера:

А.П. ЕРЕМЕЕВ, д.т.н., профессор

С.В. УЛЬЯНОВ, д.ф.-м.н., профессор

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Семенов Н.А. – заместитель главного редактора, д.т.н., профессор Тверского государственного технического университета (г. Тверь, Россия)

Сотников А.Н. – заместитель главного редактора, д.ф.-м.н., профессор, заместитель директора Межведомственного суперкомпьютерного центра РАН (г. Москва, Россия)

Афанасьев А.П. – д.ф.-м.н., профессор Московского физико-технического института (технического университета), заведующий Центром распределенных вычислений Института проблем передачи информации РАН (г. Москва, Россия)

Баламетов А.Б. – д.т.н., профессор Азербайджанского научно-исследовательского и проектно-изыскательского института энергетики (г. Баку, Азербайджан)

Борисов В.В. – д.т.н., профессор филиала Национального исследовательского университета «МЭИ» в г. Смоленске (г. Смоленск, Россия)

Голенков В.В. – д.т.н., профессор Белорусского государственного университета информатики и радиоэлектроники (г. Минск, Беларусь)

Елизаров А.М. – д.ф.-м.н., профессор Института математики и механики им. Н.И. Лобачевского Казанского федерального университета (г. Казань, Россия)

Еремеев А.П. – д.т.н., профессор Национального исследовательского университета «МЭИ» (г. Москва, Россия)

Кузнецов О.П. – д.т.н., профессор Института проблем управления РАН (г. Москва, Россия)

Курейчик В.М. – д.т.н., профессор Инженерно-технологической академии Южного федерального университета (г. Таганрог, Россия)

Мамросенко К.А. – к.т.н., доцент Московского авиационного института (национального исследовательского университета), руководитель Центра визуализации и спутниковых информационных технологий НИИСИ РАН (г. Москва, Россия)

Палюх Б.В. – д.т.н., профессор Тверского государственного технического университета (г. Тверь, Россия)

Сулейманов Д.Ш. – академик АН Республики Татарстан, д.т.н., профессор Казанского государственного технического университета (г. Казань, Республика Татарстан, Россия)

Татарникова Т.М. – д.т.н., доцент, профессор Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) (г. Санкт-Петербург, Россия)

Ульянов С.В. – д.ф.-м.н., профессор, ведущий научный сотрудник Объединенного института ядерных исследований (г. Дубна, Россия)

Хорошевский В.Ф. – д.т.н., профессор Московского физико-технического института (технического университета) (г. Москва, Россия)

Шабанов Б.М. – д.т.н., чл.-корр. РАН, директор Межведомственного суперкомпьютерного центра РАН (г. Москва, Россия)

Язенин А.В. – д.ф.-м.н., профессор Тверского государственного университета (г. Тверь, Россия)

АССОЦИИРОВАННЫЕ ЧЛЕНЫ РЕДАКЦИИ

Национальный исследовательский университет «МЭИ», г. Москва, Россия
Технологический институт Южного федерального университета, г. Таганрог, Россия
Тверской государственный технический университет, г. Тверь, Россия

АДРЕС ИЗДАТЕЛЯ И РЕДАКЦИИ
Россия, 170024,
г. Тверь, просп. Николая Корыткова, д. 3а
Телефон (482-2) 39-91-49
Факс (482-2) 39-91-00
E-mail: red@cps.tver.ru
Сайт: www.swsys.ru

Дата выхода в свет 16.12.2023 г.
Отпечатано ИПП «Фактор и К»
Россия, 170100, г. Тверь, ул. Крылова, д. 26
Выпускается один раз в квартал
Год издания тридцать шестой
Формат 60×84 1/8. Объем 200 стр.
Заказ № 16. Тираж 1000 экз. Цена 550,00 руб.

© SOFTWARE & SYSTEMS
Research journal

2023, vol. 36, no. 4
DOI: 10.15827/0236-235X.142

Publisher Research Institute
CENTERPROGRAMSYSTEM (Tver, Russian Federation)

Founder V.P. Kupriyanov

The journal is registered with the Federal Service for Supervision of Communications, Information Technology and Mass Communications (Roskomnadzor) March 3rd, 2020

Registration certificate ПИ № ФС 77-77843
ISSN 0236-235X (print)
ISSN 2311-2735 (online)

Editor-in-chief
G.I. SAVIN, Academician of RAS

Science editors of the issue:
A.P. Eremeev, Dr.Sc. (Engineering), Professor
S.V. Ulyanov, Dr.Sc. (Physics and Mathematics), Professor

EDITORIAL BOARD

Semenov N.A. – Deputy Editor-in-Chief, Dr.Sc. (Engineering), Professor of the Tver State Technical University (Tver, Russian Federation)

Sotnikov A.N. – Deputy Editor-in-Chief, Dr.Sc. (Physics and Mathematics), Professor, Deputy Director of the Joint Supercomputer Center of the Russian Academy of Sciences (Moscow, Russian Federation)

Afanasiev A.P. – Dr.Sc. (Physics and Mathematics), Professor of the Moscow Institute of Physics and Technology, Head of Centre for Distributed Computing of Institute for Information Transmission Problems (Moscow, Russian Federation)

Balametov A.B. – Dr.Sc. (Engineering), Professor of the Azerbaijan Scientific-Research & Design-Prospecting Power Engineering Institute (Baku, Azerbaijan)

Borisov V.V. – Dr.Sc. (Engineering), Professor of the MPEI Branch in Smolensk (Smolensk, Russian Federation)

Golenkov V.V. – Dr.Sc. (Engineering), Professor of the Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Elizarov A.M. – Dr.Sc. (Physics and Mathematics), Professor of the N.I. Lobachevsky Institute of Mathematics and Mechanics of the Kazan Federal University (Kazan, Russian Federation)

Eremeev A.P. – Dr.Sc. (Engineering), Professor of the National Research University ″Moscow Power Engineering Institute″ (Moscow, Russian Federation)

Kuznetsov O.P. – Dr.Sc. (Engineering), Professor of the Institute of Control Sciences of the Russian Academy of Sciences (Moscow, Russian Federation)

Kureichik V.M. – Dr.Sc. (Engineering), Professor of the Academy of Engineering and Technology of the Southern Federal University (Taganrog, Russian Federation)

Mamrosenko K.A. – Ph.D. (Engineering), Associate Professor of the Moscow Aviation Institute (National Research University), Head of the Center of Visualization and Satellite Information Technologies SRISA RAS (Moscow, Russian Federation)

Palyukh B.V. – Dr.Sc. (Engineering), Professor of the Tver State Technical University (Tver, Russian Federation)

Suleimanov D.Sh. – Academician of TAS, Dr.Sc. (Engineering), Professor of the Kazan State Technical University (Kazan, Republic of Tatarstan, Russian Federation)

Tatarnikova T.M. – Dr.Sc. (Engineering), Associate Professor, Professor of the St. Petersburg Electrotechnical University ″LETI″ (St. Petersburg, Russian Federation)

Ulyanov S.V. – Dr.Sc. (Physics and Mathematics), Professor of the Dubna International University for Nature, Society and Man (Dubna, Russian Federation)

Khoroshevsky V.F. – Dr.Sc. (Engineering), Professor of the Moscow Institute of Physics and Technology (Moscow, Russian Federation)

Shabanov B.M. – Dr.Sc. (Engineering), Corresponding Member of the RAS, Director of the Joint Supercomputer Center of the Russian Academy of Sciences (Moscow, Russian Federation)

Yazenin A.V. – Dr.Sc. (Physics and Mathematics), Professor of the Tver State University (Tver, Russian Federation)

ASSOCIATED EDITORIAL BOARD MEMBERS

National Research University ″Moscow Power Engineering Institute″, Moscow, Russian Federation
Technology Institute at Southern Federal University, Taganrog, Russian Federation
Tver State Technical University, Tver, Russian Federation

EDITORIAL BOARD AND PUBLISHER OFFICE ADDRESS
Nikolay Korytkov Ave, 3a, Tver, 170024, Russian Federation
Phone: (482-2) 39-91-49 **Fax:** (482-2) 39-91-00
E-mail: red@cps.tver.ru
Website: www.swsys.ru

Release date 16.12.2023
Printed in printing-office ″Faktor i K″
Krylova St. 26, Tver, 170100, Russian Federation
Published quarterly. 36th year of publication
Format 60×84 1/8. Circulation 1000 copies
Prod. order № 16. Wordage 200 pages. Price 550,00 rub.

Вниманию авторов

Журнал «Программные продукты и системы» публикует материалы научного и научно-практического характера по новым информационным технологиям, результаты академических и отраслевых исследований в области использования средств вычислительной техники. Практикуются выпуски тематических номеров по искусственному интеллекту, системам автоматизированного проектирования, по технологиям разработки программных средств и системам защиты, а также специализированные выпуски, посвященные научным исследованиям и разработкам отдельных вузов, НИИ, научных организаций.

Журнал «Программные продукты и системы» внесен в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук.

Информация об опубликованных статьях по установленной форме регулярно предоставляется в систему РИНЦ, в CrossRef и в другие базы и электронные библиотеки.

Журнал «Программные продукты и системы» включен в ядро коллекции РИНЦ, размещенное на платформе Web of Science в виде базы данных RSCI.

Автор статьи отвечает за подбор, оригинальность и точность приводимого фактического материала. При перепечатке ссылка на журнал обязательна. Статьи публикуются бесплатно.

Условия публикации

К рассмотрению принимаются оригинальные материалы, отвечающие редакционным требованиям и соответствующие тематике журнала. Группы научных специальностей:

1.2. Компьютерные науки и информатика

1.2.1. Искусственный интеллект и машинное обучение (физико-математические науки).

1.2.2. Математическое моделирование, численные методы и комплексы программ (физико-математические науки, технические науки)

2.3. Информационные технологии и телекоммуникации

2.3.1. Системный анализ, управление и обработка информации, статистика (технические науки, физико-математические науки).

2.3.2. Вычислительные системы и их элементы (технические науки).

2.3.3. Автоматизация и управление технологическими процессами и производствами (технические науки).

2.3.5. Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей (технические науки, физико-математические науки).

2.3.6. Методы и системы защиты информации (технические науки, физико-математические науки).

2.3.7. Компьютерное моделирование и автоматизация (технические науки, физико-математические науки).

2.3.8. Информатика и информационные процессы (технические науки).

Работа представляется в электронном виде в формате Word. Объем статьи вместе с иллюстрациями – не менее 10 000 знаков. Диаграммы, схемы, графики должны быть доступными для редактирования (Word, Visio, Excel). Заголовок должен быть информативным; сокращения, а также терминологию узкой тематики желательно в нем не использовать. Количество авторов на одну статью – не более 4, количество статей одного автора в номере, включая соавторство, – не более 2. Список литературы, наличие которого обязательно, должен включать не менее 10 пунктов.

Необходимы также содержательная структурированная аннотация (не менее 250 слов), ключевые слова (7–10) и индекс УДК. Название статьи, аннотация и ключевые слова должны быть переведены на английский язык (машинный перевод недопустим), а фамилии авторов, названия и юридические адреса организаций (если нет официального перевода) – транслитерированы по стандарту BGN/PCGN.

Вместе со статьей следует прислать экспертное заключение о возможности открытого опубликования материала и авторскую справку. Обязательно соблюдение автором договора (публичной оферты).

Порядок рецензирования

Все статьи, поступающие в редакцию (соответствующие тематике и оформленные согласно требованиям к публикации), подлежат двойному слепому рецензированию в течение месяца с момента поступления, рецензия отправляется авторам.

В редакции сформирован устоявшийся коллектив рецензентов, среди которых члены редколлегии журнала, эксперты из числа крупных специалистов в области информатики и вычислительной техники ведущих вузов страны, а также ученые и специалисты НИИСИ РАН, МСЦ РАН (г. Москва) и НИИ «Центрпрограммсистем» (г. Тверь).

Редакция журнала «Программные продукты и системы» в своей работе руководствуется сводом правил Кодекса этики научных публикаций, разработанным и утвержденным Комитетом по этике научных публикаций (Committee on Publication Ethics – COPE).

УДК 519.714.5 doi: 10.15827/0236-235X.142.509-522 2023. Т. 36. № 4. С. 509–522

Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации

П.Н. Бибило
Н.А. Кириенко
В.И. Романов

Ссылка для цитирования
Бибило П.Н., Кириенко Н.А., Романов В.И. Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации // Программные продукты и системы. 2023. Т. 36. № 4. С. 509–522. doi: 10.15827/0236-235X.142.509-522
Информация о статье
Поступила в редакцию: 31.05.2023 После доработки: 04.07.2023 Принята к публикации: 10.07.2023

Аннотация. В статье приводятся результаты экспериментальных исследований эффективности программ минимизации многоуровневых алгебраических представлений систем булевых функций, выполняемых при синтезе комбинационных схем. Результирующие минимизированные логические описания представлены в виде формул разложений Шеннона или формул, задающих булевы сети. Исследуются три подхода: совместная минимизация многоуровневых представлений систем булевых функций, раздельная минимизация и выделение из исходной системы связанных подсистем. При этом каждая из этих подсистем минимизируется отдельно, а функции, составляющие их, совместно. После получения минимизированных описаний схем, заданных в виде совокупности взаимосвязанных формул разложения Шеннона либо двухоперандных логических уравнений, соответствующих булевым сетям, осуществляется синтез логических схем в одной и той же библиотеке проектирования заказных цифровых сверхбольших интегральных схем, выполненных по КМОП СБИС (комплементарной метал-оксид-полупроводник технологии). Полученные логические схемы сравниваются по площади кристалла и по быстродействию (временной задержке). Были проведены эксперименты на 39 промышленных примерах схем. Результаты показали конкурентоспособность и целесообразность использования на практике всех трех рассмотренных подходов. Улучшение параметров схем (площадь, временная задержка) при выделении из исходной системы связанных подсистем достигается за счет того, что каждая выделенная подсистема минимизируется на основе разложений Шеннона по своей (для каждой подсистемы) перестановке переменных разложения. При этом для одной половины схем более эффективным является минимизация многоуровневых представлений на основе разложений Шеннона для исходных матричных описаний систем функций, а для другой – на основе разложений Шеннона систем функций, представленных в виде логических уравнений. Практическая значимость проведенного исследования заключается в том, что использование разработанной программы, реализующей предложенный алгоритм выделения подсистем булевых функций, позволяет во многих случаях сокращать площадь и увеличивать быстродействие функциональных блоков заказных КМОП СБИС.

Ключевые слова: система булевых функций, многоуровневое представление, дизъюнктивная нормальная форма, BDD, булева сеть, синтез логической схемы, VHDL, СБИС, КМОП-технология

Введение. Минимизация двухуровневых либо многоуровневых представлений (описаний) булевых функций и систем является первым и важнейшим этапом синтеза комбинационных схем, от которого зависят важнейшие параметры логической схемы – площадь, быстродействие и энергопотребление. Двухуровневыми (И-ИЛИ) называют представления функций в виде *дизъюнктивных нормальных форм* (ДНФ), многоуровневыми – различные формы функциональных разложений. Компактность минимизированных описаний схем достигается за счет нахождения общих частей в функциональных описаниях различного вида. Нахождение общих частей в функциональных двухуровневых описаниях систем функций легко проиллюстрировать при решении задачи минимизации булевых функций в классе ДНФ. При схемной реализации минимизированных систем ДНФ на программируемых логических матрицах преимущество получили методы и алгоритмы совместной минимизации, при которой одна и та же элементарная конъюнкция может входить в минимизированные ДНФ нескольких функций системы. Однако при синтезе схем из библиотечных логических элементов раздельная минимизация булевых функций в классе ДНФ не потеряла своего значения, так как промышленные синтезаторы логических схем часто извлекают из функциональных исходных описаний систем описания отдельных функций и осуществляют их раздельную схемную реализацию.

При минимизации многоуровневых скобочных представлений на основе методов факторизации выделяются общие части как элементарных конъюнкций и дизъюнкций, так и алгебраических подвыражений, что является одним из видов совместной минимизации многоуровневых описаний схем и оказалось полез-

ным для уменьшения площади схем заказных СБИС. В качестве структур данных при решении задач факторизации использованы булевы сети, представленные ориентированными бесконтурными графами DAG (Directed-Acyclic-Graph). При декомпозиции систем функций были разработаны различные методы совместной декомпозиции, при которой одна и та же промежуточная функция может использоваться совместно – участвовать в разложениях (суперпозициях) нескольких функций исходной системы.

Несколько последних десятилетий в качестве основных методов технологически независимой оптимизации многоуровневых представлений систем булевых функций применяются методы минимизации графовых BDD (Binary Decision Diagram)-представлений [1] и их модификаций, при этом внимание чаще уделяется совместной минимизации [1, 2] и декомпозиции [3, 4] BDD-представлений систем функций. Следует заметить, что структуры данных в виде BDD-представлений широко используются также при верификации логических схем, построении тестов и решении сложных задач в различных областях информатики. Размерность задач проектирования СБИС возрастает в связи с совершенствованием технологических процессов производства СБИС и уменьшением технологических норм [1]. Решение задач логической оптимизации начинается совместно с задачами топологического и физического проектирования СБИС с широким использованием программ проверки выполнимости *конъюнктивной нормальной формы* (КНФ) булевых функций, называемых SAT-solvers (Boolean Satisfiability problem – задача выполнимости булевой формулы, представленной в виде КНФ) [5–7].

После BDD для задания систем булевых функций появились новые структуры данных, которые используются при решении задач технологически независимой логической оптимизации [5–7]. Это AIG (AND-Inverter-Graph), XMG (XOR-Majority-Graph) [8], MIG (Majority-Inverter-Graph) [9–11], XAIG (XOR-AND-Inverter-Graph) [12]. Структуры AIG положены в основу системы ABC (http://www.eecs.berkeley.edu/~alanmi/abc) – известной академической системы оптимизации представлений систем булевых функций и синтеза логических схем. Характерной особенностью решаемых с помощью этих структур оптимизационных задач является то, что они используются для совместной реализации систем булевых функций. Раздельная минимизация многоуровневых представлений и выделение подсистем для их совместных реализаций не изучались.

Одинаковость частей в таблицах истинности булевых функций привела к идее использования связанности (общности) областей определений булевых функций при синтезе многовыходных комбинационных схем. Выделение связанных подсистем функций по таблицам истинности и ДНФ функций рассматривалось в [13]. Такой подход ограничен размерностью задач: число аргументов системы функций не превышает трех десятков. Для BDD-представлений задача выделения связанных по областям определения подсистем функций решалась в [13], где были предложены различные меры связанности функций, в том числе и основанная на общности заданного числа формул разложений Шеннона для всех функций выделяемой подсистемы. Такую связанность можно назвать сильной.

В данной работе предложены новая мера связанности подсистем функций и алгоритм последовательного формирования связанных подсистем как для BDD и их модификаций, задаваемых формулами разложений Шеннона, так и для булевых сетей, задаваемых одно- и двухоперандными логическими формулами. Алгоритм может быть легко обобщен на другие структуры данных, используемые для многоуровневых представлений систем булевых функций. Проведены программная реализация алгоритма и его экспериментальное исследование на потоке промышленных примеров, когда число аргументов и функций системы превышает сотню, а число формул (уравнений) достигает нескольких десятков тысяч. Выполнено сравнение эффективности предложенного подхода с процедурами совместной и раздельной минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона. Экспериментально установлена целесообразность использования предложенного алгоритма выделения подсистем функций при синтезе логических схем в библиотеке проектирования заказных цифровых КМОП СБИС как для BDD-представлений, так и для булевых сетей.

Совместная и раздельная минимизация многоуровневых представлений систем булевых функций на основе разложения Шеннона

BDDI-представления системы булевых функций. Под BDDI (*Binary Decision Diagram*

with Inverse cofactors) понимается ориентированный бесконтурный граф, задающий последовательные разложения Шеннона булевой функции $f(x)$ либо системы $F = \{f^1(x), \ldots, f^m(x)\}$ булевых функций по всем переменным $x_1, \ldots, x_n$ при заданном порядке (перестановке) переменных, по которым проводятся разложения, при условии нахождения пар взаимно инверсных кофакторов. Разложением Шеннона булевой функции $f(x) = f(x_1, \ldots, x_n)$ по переменной x_i является представление $f(x)$ в виде

$$f(x) = \overline{x_i} f_0 \vee x_i f_1.$$

Каждый из кофакторов (*cofactors*) $f_0 = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$, $f_1 = f(x_1, \ldots x_{i-1}, 1, x_{i+1}, \ldots, x_n)$ может быть разложен по одной из переменных из множества $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. Процесс разложения кофакторов заканчивается, когда все n переменных будут использованы для разложения. Если не ищутся взаимно инверсные кофакторы, то в результате последовательных разложений Шеннона строятся ROBDD-представления (Reduced Ordered Binary Decision Diagram – сокращенная упорядоченная бинарная диаграмма решений). В ROBDD функциональная вершина реализует один кофактор, в то время как в BDDI взаимно инверсные кофакторы представляются парой взаимосвязанных вершин. Таким образом, под BDDI будут пониматься совместные ROBDD, в которых на каждом уровне найдены пары взаимно инверсных кофакторов. Алгоритм построения ROBDD описан в [2], он был модифицирован для минимизации BDDI-представлений.

Булевы сети – Bool-представления системы булевых функций. В статье рассматриваются булевы сети [2], функциями вершин которых могут быть бинарные логические операции И (конъюнкция), ИЛИ (дизъюнкция), а также унарная операция НЕ (инверсия). Логическая минимизация булевых сетей на основе разложения Шеннона описана в [2] и заключается в поиске такой перестановки участвующих в разложении переменных, при которой число литералов в булевой сети является наименьшим. Литерал – это булева переменная либо ее инверсия. Таким образом, после разложения Шеннона по очередной переменной минимизация булевой сети сводится к следующему: ищутся вершины булевой сети, опирающиеся на одинаковые подсети, после чего проводится редукция (сокращение) сети и находятся уравнения, соответствующие редуцированной сети.

Раздельная BDDI и Bool-минимизация заключаются в нахождении своей перестановки $< x_{i_1}, \ldots, x_{i_n} >$ переменных разложения для каждой из функций $f^1(x), \ldots, f^m(x)$, в то время как при совместной BDDI- и Bool-минимизации используется одна и та же перестановка переменных разложения для всех функций системы.

Связанность функций в многоуровневом представлении системы булевых функций

Пусть система $F = \{f^1(x), \ldots, f^m(x)\}$ булевых функций задана на общем множестве Z взаимосвязанных уравнений (формул) в булевом базисе И, ИЛИ, НЕ – это наборы формул разложений Шеннона либо наборы формул, соответствующих булевым сетям. Каждая формула из Z задает одну промежуточную (внутреннюю) булеву переменную либо функцию, которая является выходной. Обозначим через $R(f^j)$ множество всех формул, задающих промежуточные переменные, связанные с формульным заданием только одной функции $f^j(x)$, $j = 1, \ldots, m$, $R(f^j) \subseteq Z$. Заметим, что собственно формула для выходной функции f^j в $R(f^j)$ не включается. Для многоуровневого формульного представления системы $F = \{f^1(x), \ldots, f^m(x)\}$ булевых функций через $R(F) = R(f^1, \ldots, f^m) = \bigcup_{i=1}^{m} R(f^i)$ обозначим множество внутренних формул для всех функций системы.

Рассмотрим далее понятие *связанности* представлений функций, которую будем понимать как общность формул в заданиях функций в виде множеств взаимосвязанных алгебраических логических уравнений. Прежде всего следует отметить, что характеристика связанности трактуется как отношение между парой объектов – компонент системы, в роли которых выступает одна или несколько функций, задаваемых уравнениями рассматриваемой системы. Для определения количественных оценок связанности введем ряд обозначений. Пусть F^g, F^h – подсистемы системы F: $F^g \subset F$, $F^h \subset F$. Назовем *весом* связаности подсистем F^g, F^h число

$$S(F^g, F^h) = \left|R(F^g) \cap R(F^h)\right| = \left|\left(\bigcup_{f^i \in F^g} R(f^i)\right) \cap \left(\bigcup_{f^j \in F^h} R(f^j)\right)\right|, \quad (1)$$

где через $|A|$ обозначена мощность множества A.

В целях нормирования количественных оценок связанности наряду с весом (1) введем понятие меры $M(F^g, F^h)$ связанности подсистем F^g, F^h, вычисляемой по формуле

$$M(F^g, F^h) = \frac{S(F^g, F^h)}{\max(|R(F^g)|, |R(F^h)|)}. \quad (2)$$

Когда каждая из подсистем представляет единственную функцию системы F, формулу (2) можно записать в виде

$$M(f^i, f^j) = \frac{S(f^i, f^j)}{\max(|R(f^i)|, |R(f^j)|)} = \frac{R(f^i) \cap R(f^j)}{\max(|R(f^i)|, |R(f^j)|)}. \quad (3)$$

Под $M(f^i, f^j)$ понимается мера связанности многоуровневых представлений пары функций f^i, f^j.

Предлагаемый далее алгоритм выделения связанных подсистем основан на последовательном пополнении уже выделенной подсистемы $F^m = \{f^1, \ldots, f^m\}$, $m \geq 2$, новой функцией f^{m+1}. Под мерой связанности системы функций $F^m \cup f^{m+1}$ будем понимать величину

$$M(F^m \cup \{f^{m+1}\}) = \frac{|R(f^1,\ldots, f^m) \cap R(f^{m+1})|}{\max(|R(f^1,\ldots, f^m)|, |R(f^{m+1})|)}. \quad (4)$$

В дальнейшем, оценивая меру связанности с использованием формулы (3), будем называть ее мерой связанности пары функций системы F в отличие от меры компонентной связанности, вычисляемой по формуле (4). Предлагаемое в данной работе понятие компонентной связанности подсистемы функций базируется на том, что для различных пар функций из этой подсистемы имеются свои подмножества общих уравнений.

Величина меры связанности всегда находится в пределах отрезка [0,1]: от полного различия множеств формул исходного представления функций (либо подсистем функций) до их полного совпадения. Это свойство меры связанности дает возможность определять границы выделяемых для совместной минимизации подсистем, причем при проведении экспериментальных исследований эту меру удобно выражать в процентах.

Примем, что рассматриваемая мера связанности обладает уровнем q, если справедливо $q \leq M(f^i, f^j)$ для связанности *пары* функций f^i, f^j или $q \leq M(F^m \cup \{f^{m+1}\})$ для компонентной связанности при пополнении системы новой функцией. Далее, когда речь пойдет о связанности систем или подсистем функций, будет подразумеваться связанность их многоуровневых представлений с оценкой по мере связанности последнего проведенного пополнения.

Рассмотрим пример системы булевых функций, заданной в таблице 1.

Таблица 1

Таблица истинности системы полностью определенных булевых функций

Table 1

Truth table of a system of fully defined Boolean functions

$x_1\ x_2\ x_3\ x_4$	$f^1 f^2 f^3 f^4$
0 0 0 0	0 0 1 1
0 0 0 1	1 1 0 0
0 0 1 0	1 0 0 0
0 0 1 1	1 1 0 0
0 1 0 0	0 0 1 1
0 1 0 1	1 1 0 0
0 1 1 0	0 1 1 1
0 1 1 1	1 1 1 0
1 0 0 0	0 0 1 1
1 0 0 1	1 0 0 0
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 0
1 1 0 0	0 0 1 1
1 1 0 1	1 0 1 1
1 1 1 0	0 0 0 1
1 1 1 1	1 0 1 0

Для данной системы функций BDDI-представление состоит из 12 формул:

$$\begin{aligned}
&f^1 = \overline{x_3} x_4 \vee x_3 s_1;\ f^2 = \overline{x_3} s_2 \vee x_3 s_3;\\
&f^3 = \overline{x_3} s_4 \vee x_3 s_5;\ f^4 = \overline{x_3} s_4 \vee x_3 \overline{s_1};\\
&s_1 = x_4 \overline{x_2} \vee x_4;\ s_2 = \overline{x_4}\,\overline{x_1};\ s_3 = \overline{x_4} s_8 \vee x_4 \overline{s_6};\\
&s_4 = \overline{x_4} \vee x_4 s_6;\ s_5 = \overline{x_4} s_7 \vee x_4 x_2;\\
&s_6 = x_1 x_2;\ s_7 = \overline{x_1} x_2;\ s_8 = \overline{x_1} x_2 \vee x_1 \overline{x_2}.
\end{aligned} \quad (5)$$

Множество $R(f^1, f^2, f^3, f^4)$ формул, задающих функции системы $F=\{f^1, f^2, f^3, f^4\}$, состоит из восьми формул, так как первые четыре формулы из (5) не включаются в множество $R(f^1, f^2, f^3, f^4)$. Отдельные функции системы заданы следующими множествами: $R(f^1) = \{s_1\}$, $R(f^2) = \{s_2, s_3, s_6, s_8\}$, $R(f^3) = \{s_4, s_5, s_6, s_7\}$, $R(f^4) = \{s_1, s_4, s_6\}$. Здесь и далее будем задавать множества формул, перечисляя только их левые части. Подсистема $\{f^3, f^4\}$ функций имеет вес связанности $S(f^3, f^4) = 2$ (в задании функций имеются две общие формулы – s_4, s_6. По формуле (3) мера связанности $M(f^3, f^4) = 2/4 = 0{,}5$, так как $|R(f^3)| = 4$, $|R(f^4)| = 3$ и поэтому $\max(|R(f^3)|, |R(f^4)|) = \max(4,3) = 4$. Легко заметить, что пара f^2, f^3 имеет меру связанности, равную 1/4. Мера компонентной связанности (4) подсистемы $\{f^2, f^3\} \cup f^4$ равна 2/7, так как $R(f^2, f^3) = \{s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$, $R(f^4) = \{s_1, s_4, s_6\}$, $R(f^2, f^3) \cap R(f^4) = \{s_4, s_6\}$, $|\{s_4, s_6\}| = 2$, $\max(|R(f^2, f^3)|, |R(f^4)|) = 7$. Если же к подсистеме $\{f^2, f^3, f^4\}$ добавить функцию f^1, то мера ком-

понентной связанности (4) системы $\{f^2, f^3, f^4\} \cup f^1$ будет равна 1/7, так как подсистема $\{f^2, f^3, f^4\}$ имеет одно общее уравнение s_1 с функцией f^1.

Рассмотрим другое множество формул, задающих те же функции:

$$
\begin{aligned}
&f^1 = \overline{t_{12}}; \quad f^2 = t_1 \vee t_2; \quad f^3 = t_3 \vee t_4; \\
&f^4 = t_5 \vee t_6; \; t_1 = x_1 t_{17}; \; t_2 = \overline{x_1} t_{13}; \\
&t_3 = x_1 t_7; \; t_4 = \overline{x_1} t_{18}; \; t_5 = x_1 t_{10}; \\
&t_6 = \overline{x_1} t_{12}; \; t_7 = t_8 \vee t_9; \; t_8 = x_4 x_2; \\
&t_9 = \overline{x_4}\,\overline{x_3}; \; t_{10} = t_{11} \vee t_{12}; \; t_{11} = x_4 t_{16}; \\
&t_{12} = \overline{x_4} t_{17}; \; t_{13} = x_4 \vee t_{15}; \; t_{14} = x_4 t_{15}; \\
&t_{15} = x_2 x_3; \; t_{16} = x_2 \overline{x_3}; \; t_{17} = \overline{x_2} x_3; \\
&t_{18} = t_{14} \vee t_{12}.
\end{aligned}
\tag{6}
$$

Как видим, каждая формула в (6) состоит только из одного булева оператора и является менее сложной (содержащей меньшее число логических операторов) по сравнению с формулами (5), задающими разложения Шеннона. Формулы (6) описывают булеву сеть, реализующую систему функций из таблицы 1.

Множества уравнений, задающие отдельные функции системы: $R(f^1) = \{t_{12}, t_{17}\}$; $R(f^2) = \{t_1, t_2, t_{13}, t_{15}, t_{17}\}$; $R(f^3) = \{t_3, t_4, t_7, t_8, t_9, t_{12}, t_{14}, t_{15}, t_{17}, t_{18}\}$; $R(f^4) = \{t_5, t_6, t_{10}, t_{11}, t_{12}, t_{16}, t_{17}\}$.

Подсистема $\{f^3, f^4\}$ функций имеет вес связанности $S(f^3, f^4) = 2$ (в задании функций имеются две общие формулы t_{12} и t_{17}) и меру связанности $M(f^3, f^4) = 2/10 = 0{,}2$, так как $|R(f^3)| = 10$, $|R(f^4)| = 7$ и $\max(|R(f^3)|, |R(f^4)|) = \max(10,7) = 10$. Примеры показывают, что различные представления (множества формул) одних и тех же функций могут иметь различные меры связанности.

Выделение подсистем с заданной мерой связанности

Предлагаемый эвристический алгоритм является модификацией предложенного в [13] алгоритма и состоит в последовательном формировании (на каждой итерации i) по текущей (остаточной) системе функций очередной подсистемы P^i функций (подсистема P^i характеризуется мерой компонентной связанности, не меньшей q). На первой итерации ($i = 1$) текущую систему функций образуют функции исходной системы.

На каждой итерации требуется выполнить шаги 1–3.

Шаг 1. Рассмотреть $C_m^2 = m(m-1)/2$ неупорядоченных пар функций $\{f^i, f^j\}$, $i, j = 1, ..., m$, $i \neq j$, текущей системы и найти такую пару функций L, которая имеет максимальное значение меры связанности, вычисляемое по формуле (3) и не меньшее параметра q. Если таких пар функций несколько, то выбирается первая из них. Если указанной пары функций нет, то переход на шаг 4.

Шаг 2. Составить из функций найденной на первом шаге пары L формируемую подсистему P^i, исключив выбранную пару функций L из текущей системы, и добавлять в формируемую подсистему поочередно те функции f^r, которые находятся с помощью следующей эвристики: из множества функций текущей системы выбирается та функция f^r, которая обеспечивает наибольшее возможное значение меры компонентной связанности (4) для подсистемы $P^i \cup \{f^r\}$. Если таких функций несколько, то выбирается и добавляется в формируемую подсистему P^i первая из них.

Шаг 3. Если нет ни одной такой функции f^r, что подсистема $P^i \cup \{f^r\}$ имеет меру компонентной связанности, не меньшую q, то закончить формирование подсистемы P^i и объявить не входящие в нее функции текущей подсистемой. Переход на шаг 1 для формирования подсистемы на итерации $i + 1$.

Шаг 4. Закончить формирование подсистем, когда все функции текущей системы будут включены в формируемые подсистемы, либо в текущей подсистеме нельзя будет найти ни одной пары функций, характеризуемых мерой (3) связанности, не меньшей q, либо в текущей системе имеется только одна функция. Конец алгоритма.

Достоинствами данного алгоритма являются его высокое быстродействие и возможность обобщения на формулы AIG, MIG, XMG, XAIG, а недостатком – зависимость от последовательности формирования результирующих выделенных подсистем. Чтобы подсчитать меру связанности результирующей выделенной подсистемы, надо знать последнюю добавляемую в подсистему функцию и, очевидно, ту подсистему, в которую она добавляется.

Приведем пример работы алгоритма на формулах (6) при значении допустимого уровня связанности $q = 0{,}2$ (20 %).

Шаг 1. Рассмотрим все неупорядоченные пары функций из формул (6) исходной системы и найдем для каждой из пар меру связанности (3): $M(f^1, f^2) = 1/5$; $M(f^1, f^3) = 2/10$; $M(f^1, f^4) = 2/7$; $M(f^2, f^3) = 2/10$; $M(f^2, f^4) = 1/7$; $M(f^3, f^4) = 2/10$.

Исходя из полученных оценок мер связанности пар функций при уровне связанности $q = 0{,}2$, результатом выполнения шага 1 является пара $\{f^1, f^4\}$, имеющая наибольшую меру связанности $M(f^1, f^4) = 2/7$.

Шаг 2. К подмножеству $\{f^1, f^4\}$ функций будем добавлять поочередно функции f^2, f^3 и находить меры компонентной связанности: $M(\{f^1, f^4\} \cup \{f^2\}) = 1/7$, $M(\{f^1, f^4\} \cup \{f^3\}) = 2/10$. Для подсистемы $(\{f^1, f^4\}, f^2)$: $R(f^1, f^4) = \{t_5, t_6, t_{10}, t_{11}, t_{12}, t_{16}, t_{17}\}$, $R(f^2) = \{t_1, t_2, t_{13}, t_{15}, t_{17}\}$, $R(f^1, f^4) \cap R(f^2) = \{t_{17}\}$, $|\{t_{17}\}| = 1$, $\max(|R(f^1, f^4)|, |R(f^2)|) = \max(7,5) = 7$, откуда $M(\{f^1, f^4\} \cup \{f^2\}) = 1/7$, что меньше 0,2.

Шаг 3. Для подсистемы $(\{f^1, f^4\}, f^3)$ значение $M(\{f^1, f^4\} \cup \{f^3\}) = 0{,}2$, поэтому на шаге 3 выделенной подсистемой объявляется $\{f^1, f^4, f^3\}$. Добавить к этой подсистеме функцию f^2 не удается, так как полученное значение меры компонентной связанности будет меньше 0,2. Текущей подсистемой объявляется $\{f^2\}$, и выполняется переход на шаг 1.

Шаг 4. В текущей подсистеме остается одна функция, поэтому алгоритм прекращает работу. Результатом являются одна выделенная подсистема $\{f^1, f^4, f^3\}$ и остаток – подсистема $\{f^2\}$.

Если провести синтез схемы непосредственно по формулам (6) (Bool-представлению), то будет получена схема с площадью 4 336 условных единиц, состоящая из 12 элементов, задержка схемы составит 2,78 нс. Если же провести совместную минимизацию в классе булевых сетей подсистемы $\{f^1, f^4, f^3\}$ и совместную минимизацию подсистемы $\{f^2\}$ (для этой подсистемы совместная минимизация вырождается в раздельную одной функции), то в результате синтеза получим схему (см. рисунок) с площадью 4 262 условные единицы, состоящую из 12 элементов, задержка схемы составит 3,21 нс. Функции логических элементов схемы и их площади приведены в таблице 2. Таким образом, параметры схемы изменились – площадь схемы уменьшилась, задержка увеличилась.

Таблица 2

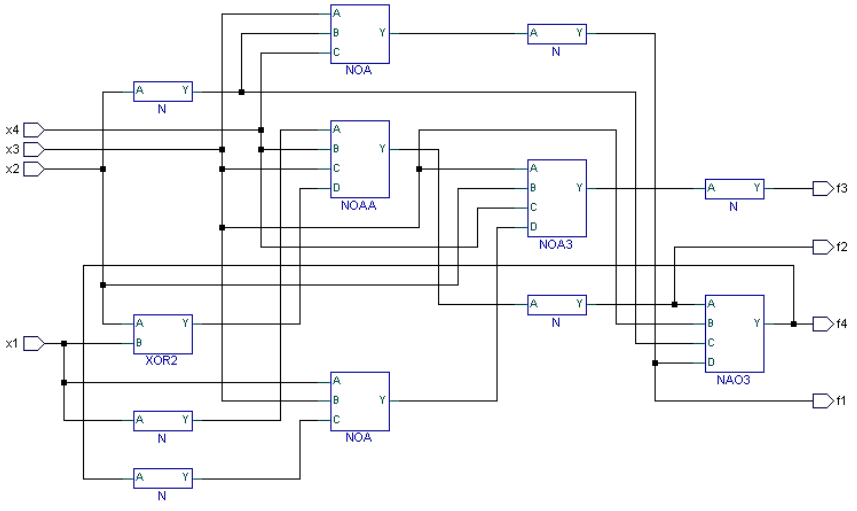
Функции и площади элементов КМОП логической схемы

Table 2

Functions and areas of CMOS logic circuit elements

Элемент	Функция	Площадь
N	$y = \overline{A}$	223
NOA	$y = \overline{(AB) \vee C}$	363
NAO3	$y = \overline{(A \vee B \vee C)D}$	491
XOR2	$y = A \oplus B$	692
NOAA	$y = \overline{AB \vee CD}$	485
NOA3	$y = \overline{(ABC) \vee D}$	530

Если же реализовать схему по уравнениям (5) (BDDI-представлению), то получим схему с площадью 4 659 условных единиц, состоящую из 14 элементов, задержка схемы составит 1,72 нс. Данный пример показывает, что различные способы минимизации многоуровневых представлений позволяют получать схемы различной площади и задержки. Проведенные про-



Логическая схема в базисе элементов КМОП

Logic circuit in CMOS element basis

граммные эксперименты проверяют конкурентоспособность трех подходов к минимизации представлений на основе двух моделей – BDDI и Bool: это совместная минимизация, раздельная минимизация и выделение отдельных подсистем и их (подсистем) совместная минимизация.

Исходные данные для экспериментов

Исходными описаниями блоков комбинационной логики являлись 39 примеров систем булевых функций на языке SF [14] в виде матричных описаний систем ДНФ – SDF описаний языка SF.

В матричной форме система ДНФ

$$\begin{aligned} f^1 &= \overline{x}_2 x_3 \vee x_4; \\ f^2 &= \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2 x_3 \vee \overline{x}_1 x_4; \\ f^3 &= \overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_2 x_3 x_4 \vee \overline{x}_3 \overline{x}_4; \\ f^4 &= x_1 \overline{x}_2 x_3 \vee x_2 \overline{x}_4 \vee \overline{x}_3 \overline{x}_4 \end{aligned} \qquad (7)$$

задается парой матриц: строки троичной матрицы T^x представляют элементарные конъюнкции, а единичные значения элементов в булевой матрице B^f отмечают вхождения соответствующих конъюнкций в ДНФ функций (табл. 3).

Таблица 3

Система ДНФ булевых функций

Table 3

DNF system of Boolean functions

T^x	B^f
$x_1\ x_2\ x_3\ x_4$	$f^1\ f^2\ f^3\ f^4$
– 0 1 –	1 0 0 0
0 1 1 –	0 1 1 0
– – – 1	1 0 0 0
1 0 1 –	0 1 0 0
1 1 0 –	0 0 1 1
– 1 – 0	0 0 0 1
– 1 1 1	0 0 1 0
0 – – 1	0 1 0 0
– – 0 0	0 0 1 1

Заметим, что данная система ДНФ задается логическими уравнениями (формат LOG языка SF) и представляет ту же систему булевых функций, которая приведена в таблице 1. Таблице истинности соответствует система совершенных ДНФ (СДНФ), если перейти к формульной записи функций.

Из 39 примеров четыре примера (Ttt2, X1, X3, X4) многоуровневых представлений систем функций были преобразованы в матричный формат SDF (добавлялось окончание имени _matr). Примеры Mult_7, Mult_8 – системы ДНФ, задающие функции семи- и восьмиразрядных умножителей. Описания функций умножителей были получены командой unmap в синтезаторе LeonardoSpectrum из схемных реализаций умножителей, после чего был осуществлен переход к ДНФ. Заметим, что так может быть осуществлено перепроектирование устройств комбинационной логики из одного базиса логических элементов в другой базис. Примеры Verg_1, Verg_2, Sist_4 взяты из практики проектирования цифровых устройств. Пример Psevdo_1 – сгенерированный псевдослучайный пример. Остальные примеры взяты из библиотеки примеров схем (http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex), представленных в формате PLA, при этом описания из формата PLA переводились в формат SDF системы FLC-2 [14]. Было установлено, что примеры GARY и IN0 задают одну и ту же систему булевых функций в виде различных систем ДНФ, то есть ДНФ с различными множествами элементарных конъюнкций.

Параметры примеров систем функций заданы в таблице 4: n – число аргументов; m – число функций системы ДНФ, заданной на k общих элементарных конъюнкциях; Area – суммарная площадь элементов схемы в условных единицах; Delay – временная задержка схемы (нс). Если $k = 2^n$, то исходной является матричная форма таблицы истинности системы функций. В таблицах 5–7 символом «*» выделены лучшие решения (в рамках отдельной таблицы) для испытываемого примера – меньшие значения параметров площади и временной задержки.

Организация экспериментов

В экспериментах использовались система FLC-2 [14] и следующие программы.

BDD_Builder – программа минимизации раздельных и совместных BDDI-представлений системы булевых функций.

BoolNetOpt2 – программа минимизации раздельных и совместных Bool-представлений системы булевых функций.

Splitter – программа выделения подсистем функций (при заданном значении q меры связанности) из совместных BDDI и из совместных Bool-представлений систем функций.

AutoSplit – программа нахождения лучшего значения q меры связанности по критерию минимальности общего числа литералов в BDDI-либо в Bool-представлениях систем функций.

Использовались также перечисленные далее стратегии обработки иерархических пред-

Таблица 4

Параметры систем функций, участвующих в экспериментах

Таблица 4

Parameters of function systems participating in experiments

№	Пример	*n*	*m*	*k*
1	Add6	12	7	1 092
2	B12	15	9	431
3	B9	16	5	123
4	Dist	8	5	256
5	Gary	15	11	442
6	Ibm	48	17	173
7	In0	15	11	138
8	In1	16	17	110
9	In2	19	10	137
10	Intb	15	7	664
11	Jbp	36	57	166
12	M181	15	9	430
13	M3	8	16	128
14	M4	8	16	256
15	Max1024	10	6	1 024
16	Max512	9	6	512
17	Mpl4	8	8	256
18	Mp2d	14	14	123
19	Mult_7	14	14	13 060
20	Mult_8	16	16	52 810
21	P82	5	14	24
22	Psevdo1	10	20	1 000
23	Radd	8	5	120
24	Root	8	5	256
25	Sist_4	17	12	370
26	Soar	83	94	529
27	Sqn	7	3	96
28	Tial	14	8	640
29	Tms	8	16	30
30	Ttt2_matr	24	21	222
31	Verg_1	17	61	2 004
32	Verg_2	18	63	2 129
33	Vtx1	27	6	110
34	X1_matr	51	35	324
35	X3_matr	135	99	915
36	X4_matr	94	71	371
37	X9dn	27	7	120
38	Z4	7	4	128
39	Z5xp1	7	10	128

ставлений систем перечисленных далее функций.

«Представить листья сети в виде LOG» – листья иерархического описания логической сети представлялись в формате LOG (логических уравнений), пример – уравнения (7).

«Представить листья сети в виде SDF» – листья иерархического описания логической сети представлялись в виде SDF (матричных описаний), пример – таблица 3.

«Представить листья сети в виде BDDI» – каждый лист SDF иерархического описания логической сети представлялся в виде уравнений, соответствующих BDDI-описанию, пример таких описаний – уравнения (5).

«Представить листья сети в виде Bool» – каждый лист SDF иерархического описания логической сети представлялся в виде уравнений, соответствующих Bool-описанию, пример таких описаний – уравнения (6).

Исходными для программы BoolNetOpt2 [2] являются логические уравнения, поэтому исходные матричные представления (формат SDF системы FLC-2) переводились в формат LOG логических уравнений. Для раздельной Bool-минимизации для такого перевода использовалась стратегия «Представить листья сети в виде LOG». После при раздельной BDDI-минимизации и после выделения подсистем программой Splitter проводилась их BDDI-минимизация с помощью стратегий «Представить листья сети в виде SDF» и «Представить листья сети в виде BDDI».

При Bool-минимизации отдельные функции системы (либо выделенные программой Splitter подсистемы) обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде LOG», «Представить листья сети в виде Bool». При выполнении стратегии «Представить листья сети в виде Bool» для каждого листа в иерархическом описании схемы (листу соответствует подсистема функций) применялась программа BoolNetOpt2 со следующим ограничением – 10 минут для совместной минимизации подсистемы. При раздельной Bool-минимизации задавалась одна минута для минимизации булевой сети, реализующей одну функцию исходной системы.

Аппарат стратегий обеспечивал для каждой функции сети (либо подсистемы) нахождение своей перестановки переменных при BDDI- либо Bool-минимизации.

Эксперименты были организованы следующим образом. Сначала выполнялась логическая минимизация, затем минимизированные описания многоуровневых представлений систем функций конвертировались в VHDL-описания [15] и подавались на вход того же синтезатора LeonardoSpectrum, который использовался в экспериментах [2]. Для каждого из примеров синтез осуществлялся с одними и теми же опциями управления синтезом и для одной и той же, что и в работе [2], целевой биб-

лиотеки синтеза – библиотеки проектирования заказных цифровых КМОП СБИС.

Эксперименты

Эксперимент 1. Совместная и раздельная минимизация в классе BDDI-представлений. При раздельной минимизации исходные матричные представления систем функций представлялись в виде логических сетей, элементами которых являются отдельные функции системы. Затем данные сети обрабатывались при помощи стратегии «Представить листья сети в виде BDDI». Для минимизации BDDI-представлений всей системы и отдельных функций использовалась программа BDD_Builder.

Эксперимент 2. Исходными для эксперимента 2 являлись совместно минимизированные BDDI-представления, для которых выполнялась программа Splitter выделения связанных подсистем функций для двух фиксированных значений меры связанности q = 10 %, q = 20 %. Полученные сети подсистем обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде BDDI». Результаты экспериментов 1 и 2 отражены в таблице 5.

Таблица 5

Результаты экспериментов 1 и 2 для BDDI-представлений

Table 5

Results of experiments 1 and 2 for BDDI-views

№	Пример	Эксперимент 1				Эксперимент 2. Выделение подсистем			
		Совместные BDDI		Раздельные BDDI		q = 10 %		q = 20 %	
		Area	Delay	Area	Delay	Area	Delay	Area	Delay
1	Add6	***16 534**	***6.82**	20 518	7.80	17 839	7.56	17 839	7.56
2	B12	17 064	3.04	15 496	2.51	15 819	2.81	***15 239**	***2.50**
3	B9	32 247	***5.32**	***31 376**	5.72	31 722	6.27	31 879	6.03
4	Dist	***62 808**	6.41	73 991	6.46	***62 808**	***6.05**	63 362	6.31
5	Gary	***91 395**	7.32	93 984	***6.61**	92 829	8.57	93 007	7.60
6	Ibm	66 776	5.62	***53 406**	5.37	59 667	***5.32**	66 776	5.62
7	In0	93 058	7.45	***90 156**	***6.23**	92 349	6.93	93 007	7.60
8	In1	***166 758**	***7.54**	283 525	7.58	263 449	8.87	263 393	9.68
9	In2	***70 291**	7.32	87 439	***6.20**	82 138	6.81	79 309	6.47
10	Intb	235 883	9.51	285 930	9.52	***233 707**	***8.75**	234 600	9.17
11	Jbp	98 308	***5.22**	***91 205**	7.17	96 339	5.58	95 044	5.69
12	M181	16 595	3.60	***15 557**	***2.60**	15 875	3.22	16 667	2.69
13	M3	***51 632**	5.13	60 437	4.59	57 519	5.33	57 892	***4.29**
14	M4	***78 181**	5.87	99 006	***4.68**	88 990	5.44	90 563	5.31
15	Max1024	***148 846**	7.66	202 660	8.07	191 974	***6.75**	181 272	7.44
16	Max512	84 682	5.91	***82 138**	***5.68**	85 904	5.74	88 337	5.89
17	Mlp4	71 095	6.09	***68 070**	***5.41**	72 389	7.17	72 546	5.51
18	Mp2d	18 805	4.87	19 173	***3.74**	18 124	3.94	***17 376**	5.17
19	Mult_7	2 492 681	14.28	2 734 390	15.20	***2 372 437**	***13.44**	2 793 560	15.26
20	Mult_8	7 069 051	***15.99**	7 291 481	16.31	***6 735 735**	16.01	7 069 051	***15.99**
21	P82	21 355	3.49	20 216	***2.92**	20 869	3.13	***20 205**	2.95
22	Psevdo1	***766 686**	9.01	813 464	***8.04**	858 622	9.28	851 302	8.79
23	Radd	11 802	***3.69**	***8 911**	4.77	9.480	3.85	9.480	3.85
24	Root	***26 075**	4.78	30 584	***4.13**	***26 075**	4.78	***26 075**	4.78
25	Sist_4	112 722	7.21	120 037	***6.82**	118 569	6.94	***112 565**	7.56
26	Soar	163 957	6.14	158 388	5.58	***147 976**	6.05	156 368	***5.20**
27	Sqn	22 878	3.99	22 683	***3.33**	22 526	3.74	***20 975**	3.55
28	Tial	***279 458**	9.08	384 172	10.96	292 721	***8.99**	330 197	9.08
29	Tms	***39 004**	4.65	44 952	5.38	43 708	5.13	46 560	***4.57**
30	Ttt2_matr	76 429	5.01	***45 683**	***4.41**	64 321	5.52	77 439	6.00
31	Verg_1	***465 032**	14.41	939 081	***12.21**	801 790	12.77	845 839	12.28
32	Verg_2	***536 037**	14.15	1 082 989	13.28	917 430	15.31	988 687	13.12
33	Vtx1	31 600	***4.23**	40 103	4.89	28 285	4.66	***27 370**	5.05
34	X1_matr	107 912	6.45	***71 720**	***4.46**	82 528	5.52	108 542	6.91
35	X3_matr	211 633	***5.73**	223 222	6.51	***199 318**	5.82	207 308	6.01
36	X4_matr	***120 031**	6.17	133 959	***5.99**	157 417	6.51	127 503	6.08
37	X9dn	34 719	6.58	41 939	5.09	***30 327**	***4.31**	34 278	5.34
38	Z4	***6 640**	4.25	7 700	***3.57**	6.640	4.25	6.640	4.25
39	Z5xp1	***22 582**	5.24	30 417	3.51	24 926	***2.97**	26 120	3.30
Лучших решений		17	8	10	18	8	8	7	5

Эксперимент 3. Совместная и раздельная минимизация в классе булевых сетей (Bool-представлений). При раздельной минимизации в классе Bool-представлений элементы логической сети (отдельные функции системы) обрабатывались с помощью стратегий «Представить листья сети в виде SDF», «Представить листья сети в виде LOG», «Представить листья сети в виде Bool». Базовой программой являлась программа минимизации булевых сетей BoolNetOpt2 [2].

Под Bool-представлениями в системе FLC-2 понимаются булевы сети, описываемые логическими уравнениями с одним логическим оператором дизъюнкции либо конъюнкции над двумя литералами булевых переменных.

Эксперимент 4. Исходными для эксперимента 4 являлись совместно минимизированные Bool-представления, для которых выполнялась программа Splitter выделения q-связанных подсистем функций для двух фиксированных значений меры связанности q = 5 %, q = 20 %. Полученная сеть подсистем обрабатывалась с помощью тех же трех стратегий, что и в эксперименте 3.

Результаты экспериментов 3 и 4 сведены в таблицу 6.

Таблица 6

Результаты экспериментов 3 и 4 для Bool-представлений

Table 6

Results of experiments 3 and 4 for Bool-views

№	Пример	Эксперимент 3				Эксперимент 4. Выделение подсистем			
		Совместные Bool		Раздельные Bool		q = 5 %		q = 20 %	
		Area	Delay	Area	Delay	Area	Delay	Area	Delay
1	Add6	***18 314**	6.18	29 574	4.24	20 389	***4.18**	20 389	***4.18**
2	B12	15 936	3.55	17 075	3.38	15 295	***2.80**	***15 105**	2.99
3	B9	***22 454**	4.69	25 607	4.06	28 793	***3.42**	30 316	4.20
4	Dist	66 134	***5.33**	80 603	5.98	***63 249**	5.52	69 834	5.79
5	Gary	97 795	7.94	91 802	7.96	92 996	***6.57**	***89 821**	7.31
6	Ibm	66 776	5.62	63 891	6.45	90 647	7.33	***61 720**	***4.83**
7	In0	***95.792**	***6.20**	96 506	6.87	***95 792**	6.95	98 727	6.75
8	In1	***162 478**	8.52	254 325	***7.86**	187 650	8.78	191 210	10.64
9	In2	93 716	6.25	***85 592**	***5.84**	87 807	6.46	95 044	6.04
10	Intb	***253 299**	9.30	338 594	9.58	254 816	***9.05**	255 458	9.08
11	Jbp	98 308	***5.22**	97 114	5.69	103 358	5.84	***94 827**	5.29
12	M181	16 204	3.02	***15 546**	***2.93**	16 070	3.39	16 316	3.02
13	M3	65 414	5.14	***59 823**	***4.66**	74 292	6.67	61 603	5.23
14	M4	***93 153**	5.24	98 308	***4.79**	94 732	4.93	106 617	4.89
15	Max1024	129 478	7.38	200 026	7.67	129 272	***7.09**	***123 112**	7.81
16	Max512	***72 300**	6.26	98 945	6.74	79 721	6.99	73 840	***5.49**
17	Mlp4	68 773	4.95	74 560	4.96	***67 987**	5.83	68 277	***4.82**
18	Mp2d	***17 203**	3.81	19 329	***3.67**	17 432	5.05	18 833	4.01
19	Mult_7	2 298 056	13.34	2 068 489	13.00	***2 062 212**	***12.61**	2 367 148	13.26
20	Mult_8	5 938 638	15.43	***5 536 956**	15.82	6 503 066	***14.96**	5 960 868	15.80
21	P82	***19 056**	3.23	19 792	***2.79**	19 748	3.23	19 167	3.26
22	Psevdo1	735 796	8.91	796 361	***8.29**	***729 507**	8.42	731 376	9.55
23	Radd	***7 399**	3.99	12 282	***3.26**	***7 399**	3.99	***7 399**	3.99
24	Root	31 773	***4.10**	38 139	4.43	36 331	5.49	***28 341**	4.41
25	Sist_4	128 351	7.38	***121 845**	***6.27**	153 941	8.06	139 271	8.59
26	Soar	151 787	6.52	151 793	***5.41**	***140 019**	6.10	152 485	5.59
27	Sqn	26 695	4.23	23 715	3.96	***22 008**	***3.50**	22 532	3.55
28	Tial	351 607	9.79	***337 099**	***9.85**	389 679	11.13	375 595	10.59
29	Tms	50 471	6.88	***46 202**	***4.88**	55 080	6.23	52 435	5.26
30	Ttt2_matr	52 653	***4.05**	***46 721**	4.93	51 777	5.94	48 752	4.52
31	Verg_1	***634 954**	19.25	1 145 446	***11.23**	999 679	15.64	1 051 674	13.74
32	Verg_2	***694 425**	18.90	1 212 986	***12.21**	987 481	15.84	1 059 960	13.80
33	Vtx1	***28 112**	***4.31**	32 353	5.55	31 376	4.51	29 892	5.31
34	X1_matr	107 912	6.45	***71 441**	***4.41**	93 727	7.15	90 848	7.02
35	X3_matr	***211 633**	***5.73**	212 910	5.82	213 162	7.41	219 785	7.15
36	X4_matr	120 031	6.17	119 194	4.84	***109 680**	6.31	116 432	***4.51**
37	X9dn	***20 741**	5.48	21 310	***4.89**	49 924	5.22	38 725	5.62
38	Z4	6 685	3.83	10 284	***3.11**	6 685	3.83	***6 411**	3.74
39	Z5xp1	26 014	3.76	26 594	***3.44**	***20 267**	4.64	23 536	5.75
Лучших решений		15	7	9	19	10	9	8	5

Эксперимент 5. Оценка выделения связанных подсистем. Для выделения совместно минимизируемых связанных подсистем с помощью программы AutoSplit ищутся лучшие значения параметра связанности q для BDDI-представлений путем итеративного выполнения программы Splitter для меры связанности 5 %, 10 %, 15 %, 20 %, ..., 95 %. Оценивается суммарное число литералов в каждой полученной логической сети, и по минимуму найденной оценки выбирается лучший вариант решения. Естественно, если для значения q меры связанности подсистемы не выделялись, то программа AutoSplit прекращала испытывать большие, чем q, значения меры связанности и переходила к выбору решения по критерию меньшего числа литералов. Полученная логическая сеть обрабатывалась с помощью тех стратегий, что и в эксперименте 2. Понятно, что выбор лучшего значения связанности целесообразно находить после выполнения синтеза схемы, но тогда трудоемкая процедура синтеза должна выполняться многократно, поэтому вычислялась погрешность δ эвристического

Таблица 7

Результаты экспериментов 5 и 6

Table 7

Results of experiments 5 and 6

№	Пример	Эксперимент 5				Эксперимент 6			
		***q*, %**	**Area**	**Delay**	**δ, %**	***q*, %**	**Area**	**Delay**	**δ, %**
1	Add6	5	17 839	7,56	0	5	20 389	4,18	0
2	B12	5	***15 819**	2,81	3,8	10	***15 674**	3,13	3,7
3	B9	5	31 879	6,03	0,5	5	28 793	3,42	6,1
4	Dist	5	62 808	6,41	0	5	***63 249**	5,52	0
5	Gary	20	93 007	7,6	0,2	45	93 113	7,47	3,6
6	Ibm	10	59 667	5,32	0	35	93 247	8,42	51
7	In0	20	93 007	7,6	0,7	20	98 727	6,75	5,2
8	In1	80	186 947	7,9	0	75	164 616	9,12	0
9	In2	20	79 309	6,47	0	5	87 807	6,46	3,4
10	Intb	10	***233 707**	8,75	0	5	254 816	9,05	0
11	Jbp	55	100 596	5,89	5,8	60	109 591	6	16
12	M181	5	15 875	3,22	0	10	15 808	2,72	0
13	M3	40	51 632	5,13	0	30	70 570	5,2	14,5
14	M4	60	78 237	5,91	0	45	96 043	5,59	1,3
15	Max1024	5	191 974	6,75	5,9	5	***129 272**	7,09	5
16	Max512	5	85 904	5,74	0	5	79 721	6,99	7,9
17	Mlp4	5	72 389	7,17	0	5	***67 987**	5,83	0
18	Mp2d	95	***18 369**	4,87	5,7	55	17 410	4,37	0
19	Mult_7	10	***2 372 437**	13,44	0	5	***2 062 212**	12,61	0
20	Mult_8	10	***6 735 735**	16,01	0	45	6 145 572	16,66	4
21	P82	10	20 869	3,13	3,2	50	20 886	3,3	8,9
22	Psevdo1	25	781 981	11,03	0	5	***729 507**	8,42	0
23	Radd	5	9 480	3,85	0	5	7 399	3,99	0
24	Root	50	24 585	5,21	0	25	33 195	4,33	17,1
25	Sist_4	25	130 432	8,38	15,8	35	183 286	9,58	31,6
26	Soar	90	171 836	5,79	16,1	75	***148 417**	5,81	5,9
27	Sqn	30	***22 526**	3,74	2,3	5	***22 008**	3,5	0
28	Tial	5	294 724	8,73	0,6	5	389 679	11,13	3,7
29	Tms	55	43 814	5,2	0,2	80	48 161	5,62	0
30	Ttt2_matr	20	77 439	6	22,5	30	49 517	4,89	1,5
31	Verg_1	90	484 924	15,05	0	95	672 239	15,73	0
32	Verg_2	85	634 385	14,16	0	90	759 990	16,69	0
33	Vtx1	5	***28 285**	4,66	3,3	5	31 376	4,51	16,3
34	X1_matr	5	120 232	8,34	45,6	10	108 503	8,02	19,4
35	X3_matr	70	211 912	5,76	6,3	15	217 001	8,47	1,8
36	X4_matr	5	126 962	6,9	0	5	***109 680**	6,31	0
37	X9dn	55	***33 430**	5,18	10,2	70	43 368	5,17	17,6
38	Z4	50	6 640	4,25	3,5	40	6 685	3,83	4,2
39	Z5xp1	5	26 416	3,39	1,1	5	***20 267**	4,64	0
Средняя погрешность					3,9	-	-	-	6,4

критерия «Общее число литералов» при выборе решения – лучшего результата выделения связанных подсистем. Например, в таблице 7 для примера B12 (BDDI-представление) указана погрешность δ = 3,8 %. Погрешность δ вычислялась следующим образом: программа AutoSplit нашла лучшее значение q = 5 % меры связанности, программа Splitter выделила подсистемы, которые были минимизированы, синтез схемы по полученному описанию подсистем позволил получить схему с площадью Area = 15 819, однако ручной перебор значений q (табл. 5) и синтез двух схем позволили получить схему площадью Area = 15 239 для значения q = 20 %. Погрешность составила (15 819–15 239)/15 239 = 0,038, что эквивалентно 3,8 %.

Эксперимент 6. Совпадает с экспериментом 5, при этом исходными были совместно минимизированные Bool-представления, а результирующие логические сети обрабатывались с помощью тех же стратегий, что и в эксперименте 3.

Итоги экспериментов 5 и 6 отражены в таблице 7.

Обсуждение результатов экспериментов

Результаты экспериментов 1 и 2 (табл. 5) показывают, что предложенный алгоритм выделения подсистем в 13 примерах схем (из 39 примеров) позволил получить схемы меньшей площади для исходных совместных BDDI-представлений, из которых выделялись подсистемы. Для 17 примеров схем лучшие результаты показала совместная минимизация, для 10 – раздельная минимизация BDDI-представлений. Получение схем с меньшей задержкой чаще было обеспечено применением раздельной BDDI-минимизации и алгоритмом выделения подсистем.

При выделении подсистем из совместных Bool-представлений (табл. 6) примерно в половине случаев лучшие решения были получены совместной и раздельной минимизацией, а в другой половине – алгоритмом выделения подсистем. Предложенный алгоритм выделения связанных подсистем на основе критерия компонентной связанности оказался эффективнее предложенного в [13] алгоритма, основанного на другой мере связанности функций выделяемых подсистем, предполагающей общие уравнения для всех функций подсистемы.

Критерий минимальности общего числа литералов в функциональных BDDI-представлениях оказался эффективным при автоматическом выборе лучшего решения для программы, реализующей предложенный алгоритм выделения подсистем. Сравнение с ручным выбором лучшего значения меры связанности показало погрешность в 3,9 % для BDDI-представлений и 6,4 % для Bool-представлений (табл. 7). Программа AutoSplit обеспечила восемь лучших решений для BDDI-представлений и десять – для Bool-представлений.

Проведенные эксперименты показали, что для одной половины примеров систем функций более эффективным является аппарат минимизации на основе матричных представлений, для другой – аппарат минимизации булевых сетей.

Заключение

Экспериментально установлено, что для некоторых систем функций при синтезе логических схем из библиотечных элементов более эффективна совместная минимизация, для других – раздельная минимизация, среднюю позицию (по виду синтезированных схем) занимает оптимизационная процедура выделения подсистем для совместной минимизации. Эффект достигается за счет того, что каждая выделенная подсистема минимизируется на основе разложений Шеннона по своей (для каждой подсистемы) перестановке переменных разложения. При этом для промышленных примеров описаний проектов многовыходных комбинационных схем, математическими моделями поведения которых являются системы булевых функций, целесообразно испытывать две модели функциональных описаний и, соответственно, два подхода: первый – минимизация на основе разложений Шеннона и аппарата матричных описаний, второй – минимизация на основе разложений Шеннона для логических уравнений и последующая структурная минимизация полученных булевых сетей.

Список литературы

1. Amaru L.G. New Data Structures and algorithms for logic synthesis and verification. Springer Publ., 2017, 156 p. doi: 10.1007/978-3-319-43174-1.

2. Бибило П.Н., Ланкевич Ю.Ю. Экспериментальное сравнение эффективности алгоритмов оптимизации BDD-представлений систем булевых функций // Программные продукты и системы. 2020. Т. 33. № 3. С. 449–463. doi: 10.15827/0236-235X.131.449-463.

3. Kubica M., Kania D. SMTBDD: New form of BDD for logic synthesis. IJET, 2016, vol. 62, no. 1, pp. 33–41. doi: 10.1515/eletel-2016-0004.

4. Kubica M., Kania D. Decomposition of multi-output functions oriented to configurability of logic blocks. Bull. Pol. Acad. Sci. Tech. Sci., 2017, vol. 65, no. 3, pp. 317–331. doi: 10.1515/bpasts-2017-0036.

5. Reis A.I., Drechsler R. Advanced Logic Synthesis. Springer Publ., 2017, 232 p. doi: 10.1007/978-3-319-67295-3.

6. Lee S.-Y., Riener H., Mishchenko A., Brayton R.K., De Micheli G. A simulation-guided paradigm for logic synthesis and verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, vol. 41, no. 8, pp. 2573–2586. doi: 10.1109/TCAD.2021.3108704.

7. Haaswijk W., Soeken M., Mishchenko A., De Micheli G. SAT-based exact synthesis: Encodings, topology families, and parallelism. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, vol. 39, no. 4, pp. 871–884. doi: 10.1109/TCAD.2019.2897703.

8. Haaswijk W., Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. A novel basis for logic rewriting. Proc. ASP-DAC, 2017, pp. 151–156. doi: 10.1109/ASPDAC.2017.7858312.

9. Soeken M., Amaru L., Gaillardon P., De Micheli G. Optimizing majority-inverter graphs with functional hashing. Proc. DATE, 2016, pp. 1030–1035.

10. Soeken M., Amaru L., Gaillardon P.-E., De Micheli G. Exact synthesis of majority-inverter graphs and its applications. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017, vol. 36, no. 11, pp. 1842–1855. doi: 10.1109/TCAD.2017.2664059.

11. Riener H., Testa E., Amaru L., Soeken M., De Micheli G. Size optimization of MIGs with an application to QCA and STMG technologies. Proc. NANOARCH, 2018, pp. 157–162.

12. Harlecek I., Fiser P., Schmidt J. Are XORs in logic synthesis really necessary? Proc. IEEE 20th Int. Symposium DDECS, 2017, pp. 134–139. doi: 10.1109/DDECS.2017.7934583.

13. Бибило П.Н., Позняк А.М. Выделение подсистем связанных функций из многоуровневого представления системы булевых функций // Информатика. 2020. Т. 17. № 1. С. 63–77. doi: 10.37661/1816-0301-2020-17-1-63-77.

14. Бибило П.Н., Романов В.И. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний // Проблемы разработки перспективных микро- и наноэлектронных систем. 2020. № 4. С. 9–16. doi: 10.31114/2078-7707-2020-4-9-16.

15. Тарасов И.Е. ПЛИС XILINX. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. М.: Горячая линия; Телеком, 2020. 538 с.

Software & Systems doi: 10.15827/0236-235X.142.509-522 2023, vol. 36, no. 4, pp. 509–522

Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization

Petr N. Bibilo
Natalia A. Kirienko
Vladimir I. Romanov

For citation
Bibilo, P.N., Kirienko, N.A., Romanov, V.I. (2023) 'Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization', *Software & Systems*, 36(4), pp. 509–522 (in Russ.). doi: 10.15827/0236-235X.142.509-522

Article info
Received: 31.05.2023 After revision: 04.07.2023 Accepted: 10.07.2023

Abstract. The paper presents experimental results obtained from studying the effectiveness of programs for minimizing multilevel algebraic representations of Boolean function systems performed in combinational circuit synthesis. The resulting minimized logical descriptions are presented as Shannon expansion formulas or formulas defining Boolean networks. Three approaches are investigated: joint minimization of multilevel representations of Boolean function systems; separate minimization; selection of connected subsystems from an original system, where each connected subsystem is minimized separately, however functions that make up each subsystem are minimized together. After obtaining minimized descriptions of circuits given as a set of interrelated Shannon expansion formulas or as two-operand logical equations corresponding to Boolean networks, the synthesis of logic circuits is performed within the same library for designing custom digital CMOS VLSI (very large integrated circuits made using complementary metal oxide-semiconductor technology). The resulting logic circuits are compared in terms of a crystal area and speed (time delay). Experiments involve 39 industrial examples of circuits. Experimental results show competitiveness and expediency of all three considered approaches in practice. The improvement of circuit parameters (area, time delay) in approach 3 is achieved due to minimizing each selected subsystem

on the basis of Shannon expansions in its own (for each subsystem) permutation of expansion variables. At the same time, minimizing multilevel representations based on Shannon expansions for function system matrix descriptions is more effective for one half of circuits; minimizing based on Shannon expansions of function systems presented as logical equations is effective for the other half. The practical significance of the study is that the developed software, which implements the proposed algorithm for extracting Boolean function subsystems, allows reducing the area and increasing the performance of functional blocks of custom CMOS VLSI in many cases.
Keywords: the system of Boolean functions, disjunctive normal form (DNF), Binary Decision Diagram (BDD), Boolean net, digital logic synthesis, VHDL, VLSI, CMOS

References

1. Amaru, L.G. (2017) *New Data Structures and Algorithms for Logic Synthesis and Verification*. Springer Publ., 156 p. doi: 10.1007/978-3-319-43174-1.

2. Bibilo, P.N., Lankevich, Yu.Yu. (2020) 'Experimental investigation of effectiveness of algorithms for minimizing BDD representations of Boolean function systems', *Software & Systems*, 33(3), pp. 449–463 (in Russ.). doi: 10.15827/0236-235X.131.449-463.

3. Kubica, M., Kania, D. (2016) 'SMTBDD: New form of BDD for logic synthesis', *IJET*, 62(1), pp. 33–41. doi: 10.1515/eletel-2016-0004.

4. Kubica, M., Kania, D. (2017) 'Decomposition of multi-output functions oriented to configurability of logic blocks', *Bull. Pol. Acad. Sci. Tech. Sci*., 65(3), pp. 317–331. doi: 10.1515/bpasts-2017-0036.

5. Reis, A.I., Drechsler, R. (2017) *Advanced Logic Synthesis*. Springer Publ., 232 p. doi: 10.1007/978-3-319-67295-3.

6. Lee, S.-Y., Riener, H., Mishchenko, A., Brayton, R.K., De Micheli, G. (2022) 'A simulation-guided paradigm for logic synthesis and verification', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8), pp. 2573–2586. doi: 10.1109/TCAD.2021.3108704.

7. Haaswijk, W., Soeken, M., Mishchenko, A., De Micheli, G. (2020) 'SAT-based exact synthesis: Encodings, topology families, and parallelism', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(4), pp. 871–884. doi: 10.1109/TCAD.2019.2897703.

8. Haaswijk, W., Soeken, M., Amaru, L., Gaillardon, P.-E., De Micheli, G. (2017) 'A novel basis for logic rewriting', *Proc. ASP-DAC*, pp. 151–156. doi: 10.1109/ASPDAC.2017.7858312.

9. Soeken, M., Amaru, L., Gaillardon, P., De Micheli, G. (2016) 'Optimizing majority-inverter graphs with functional hashing'. *Proc. DATE*, pp. 1030–1035.

10. Soeken, M., Amaru, L., Gaillardon, P.-E., De Micheli, G. (2017) 'Exact synthesis of majority-inverter graphs and its applications'. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(11), pp. 1842–1855. doi: 10.1109/TCAD.2017.2664059.

11. Riener, H., Testa, E., Amaru, L., Soeken, M., De Micheli, G. (2018) 'Size optimization of MIGs with an application to QCA and STMG technologies', *Proc. NANOARCH*, pp. 157–162.

12. Harlecek, I., Fiser, P., Schmidt, J. (2017) 'Are XORs in logic synthesis really necessary?' *Proc. IEEE 20th Int. Symposium DDECS*, pp. 134–139. doi: 10.1109/DDECS.2017.7934583.

13. Bibilo, P.N., Pazniak, A.M. (2020) 'The search for subsystems of related functions from multilevel representation of systems of Boolean functions', *Informatics*, 17(1), pp. 63–77 (in Russ.). doi: 10.37661/1816-0301-2020-17-1-63-77.

14. Bibilo, P.N., Romanov, V.I. (2020) 'The system of logical optimization of functional-structural descriptions of digital circuits based on production-frame knowledge representation model', *Problems of Advanced Micro- and Nanoelectronic Systems Development*, (4), pp. 9–16 (in Russ.). doi: 10.31114/2078-7707-2020-4-9-16.

15. Tarasov, I.E. (2020) *XILINX FPGA. Hardware Description Languages VHDL and Verilog, CAD, Design Techniques*. Moscow, 538 p. (in Russ.).

Авторы
Бибило Петр Николаевич [1], д.т.н., профессор, заведующий лабораторией, bibilo@newman.bas-net.by
Кириенко Наталья Алексеевна [1], к.т.н., доцент, старший научный сотрудник, kir@newman.bas-net.by
Романов Владимир Ильич [1], к.т.н., доцент, ведущий научный сотрудник, rom@newman.bas-net.by

[1] Объединенный институт проблем информатики Национальной академии наук Беларуси, г. Минск, 220012, Беларусь

Authors
Petr N. Bibilo [1], Dr.Sc. (Engineering), Professor, Head of Laboratory, bibilo@newman.bas-net.by
Natalia A. Kirienko [1], Cand. of Sci. (Engineering), Associate Professor, Senior Researcher, kir@newman.bas-net.by
Vladimir I. Romanov [1], Cand. of Sci. (Engineering), Associate Professor, Leading Researcher, rom@newman.bas-net.by

[1] United Institute of Informatics Problems of the National Academy of Sciences of Belarus (UIIP NASB), Minsk, 220012, Belarus

Software & Systems doi: 10.15827/0236-235X.142.523-538 2023, vol. 36, no. 4, pp. 523–538

Quantum algorithmic gate-based computing: Grover quantum search algorithm design in quantum software engineering

Sergey V. Ulyanov
Viktor S. Ulyanov

For citation
Ulyanov, S.V., Ulyanov, V.S. 'Quantum algorithmic gate-based computing: Grover quantum search algorithm design in quantum software engineering', *Software & Systems*, 2023, 36(4), pp. 523–538 (in Russ.). doi: 10.15827/0236-235X.142.523-538
Article info
Received: 23.03.2023 After revision: 08.06.2023 Accepted: 21.08.2023

Abstract. The difference between classical and quantum algorithms (QA) is following: problem solved by QA is coded in the structure of the quantum operators. Input to QA in this case is always the same. Output of QA says which problem coded. In some sense, give a function to QA to analyze and QA returns its property as an answer without quantitative computing. QA studies qualitative properties of the functions. The core of any QA is a set of unitary quantum operators or quantum gates. In practical representation, quantum gate is a unitary matrix with particular structure. The size of this matrix grows exponentially with an increase in the number of inputs, which significantly limits the QA simulation on a classical computer with von Neumann architecture. Quantum search algorithm (QSA) – models apply for the solution of computer science problems as searching in unstructured data base, quantum cryptography, engineering tasks, control system design, robotics, smart controllers, etc. Grover's algorithm is explained in details along with implementations on a local computer simulator. The presented article describes a practical approach to modeling one of the most famous QA on classical computers, the Grover algorithm.

Keywords: quantum search algorithms, quantum circuits, quantum algorithmic gate, quantum operators design
Acknowledgements. Fruitful discussions with V. Korepin and C. Bennett helped the authors to formulate the results obtained in the work more clearly

Introduction. Applied Quantum Search Algorithm Model

Grover Quantum Search Algorithm (QSA) is one of the famous quantum algorithms (QA) that outperform their classical counterparts [1–4]. In the conventional linear search algorithm, it required $\mathcal{O}(N)$ comparisons to find an element in an array of length *N*. Grover's algorithm achieves a quadratic speed up; i.e., it has a complexity of $\mathcal{O}\left(\sqrt{N}\right)$. Grover's search algorithm provides an example of the speed-up that would be offered by quantum computers (if and when they are built) and has the important application in solution of *global optimization* control problems. The problem solved by Grover's algorithm is finding a sought-after ("*marked*") element in an unsorted database (DB) of size *N*. To solve this problem, a classical computer would need $\frac{N}{2}$ database queries on average, and in the worst case it would $N-1$ queries.

Thus, using Grover's algorithm, a quantum computer can find the marked state using only $\mathcal{O}\left(\sqrt{N}\right)$ quantum data queries. In the case of *M* "*marked*" elements in an unsorted *DB* of size *N* speed-up of quantum search process increases as $\mathcal{O}\left(\sqrt{\frac{N}{M}}\right)$. It is believed that this complexity is optimal. This speed up is inherently due to the parallel computational nature of quantum operators that can affect all of the coefficients in the state expansion at once.

General design structure of quantum algorithms

A quantum algorithm calculates the qualitative properties of the function *f*.

From a mathematical standpoint, a function f is the map of one logical state into another.

The problems solved by a QA can be stated as follows:

Given a function $f:\{0,1\}^n \to \{0,1\}^m$; find a certain property of the function *f*.

Or in the symbolic form as:

Input: A function $f:\{0,1\}^n \to \{0,1\}^m$.

Problem: Find a certain property of *f*.

Figure 1 is a block diagram showing a gate approach for simulation of a QA using classical computers [5]:

In Fig. 1, an input is provided to a QA and the QA produces an output. However, the QA can be transformed to produce a quantum algorithmic gate (QAG) such that an input vector (correspond-

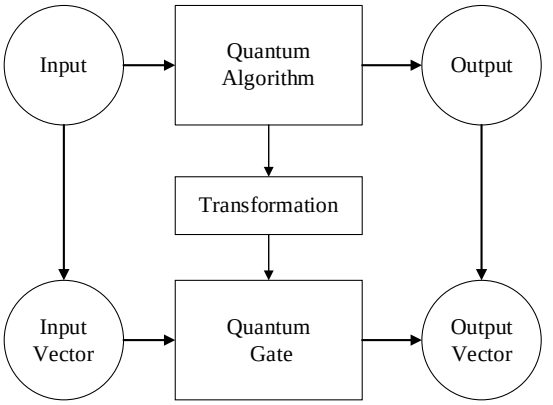


Fig. 1. The gate approach for simulation of quantum algorithms using classical computers

ing to the QA input) is provided to the QAG to produce an output vector (corresponding to the QA output) [5].

Figure 2 shows classification tree of QA's for quantum soft computing and control engineering applications. QA's are either decision-making or searching as described above.

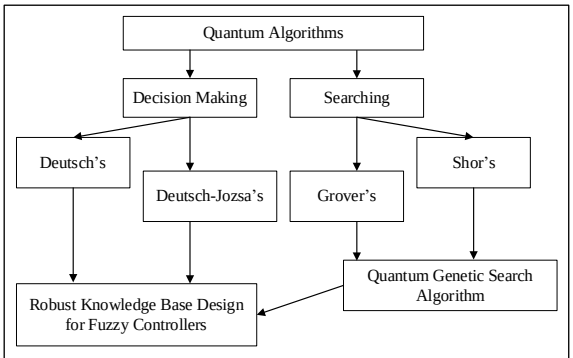


Fig. 2. Classification of quantum algorithms

As shown, as example, in Fig. 2, Quantum Genetic Search Algorithms (QGSA) follows from Grover's and Shor's algorithms, and background for Robust KB design of Fuzzy Controllers follows from Deutch's, Deutch–Josa's, Grover's and/or Shor's algorithms (see in details [4, 5]).

Let us briefly consider the design process of QAG.

Figure 3 is a block diagram showing the design process of the QAG.

In Fig. 3 an input block of the QA is a function *f* that maps binary strings into binary strings. This function *f* is represented as a map table, defined for every string its image. The function is first encoded into a unitary matrix operator U_F depending on the properties of *f*. In some sense, this operator calculates *f* when its input and output strings are encoded into canonical basis vectors of a complex

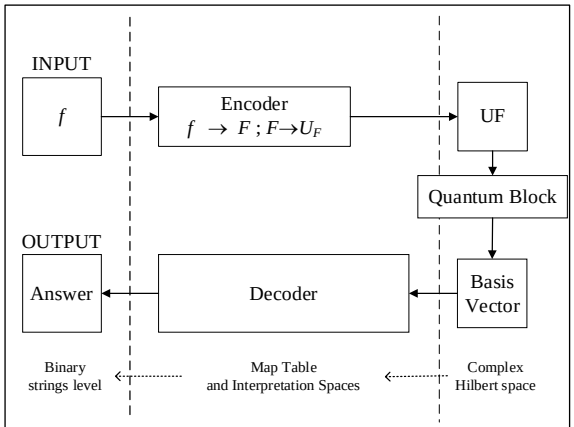


Fig. 3. Schematic diagram of QA

Hilbert space. The operator U_F maps the vector code of every string into the vector code of its image by *f*. The quantum block operates on basis vectors in a complex Hilbert space. The vectors operated on by the quantum block are provided to a decoder, which decodes the vectors to produce an answer.

Once generated, the matrix operator U_F is embedded into a quantum gate *G*. The quantum gate *G* is a unitary matrix whose structure depends on the form of matrix U_F and on the problem to be solved. The quantum gate is a unitary operator built from the dot composition of other more specific operators. The specific operators are described as tensor products of smaller matrices.

General structure of the QAG design method

Traditionally QA is written as a quantum circuit [2].

Figure 4 shows the general structure of a quantum circuit for a QAG.

As shown in Fig. 4, the general structure of the quantum circuit is based on three reversible quantum operators (superposition, entanglement, and interference) and irreversible classical operator measurement.

The quantum circuit is a high-level description of how these smaller matrices are composed using tensor and dot products in order to generate the final quantum gate as shown in Fig. 4 (see in details [4, 5]). Thus, the mathematical background of this approach is based on mappings between the quantum block operations in the complex Hilbert space [2].

The encoder and decoder operate in a map table and interpretation space, and input/output occurs on a binary string level. The Clifford and Pauli

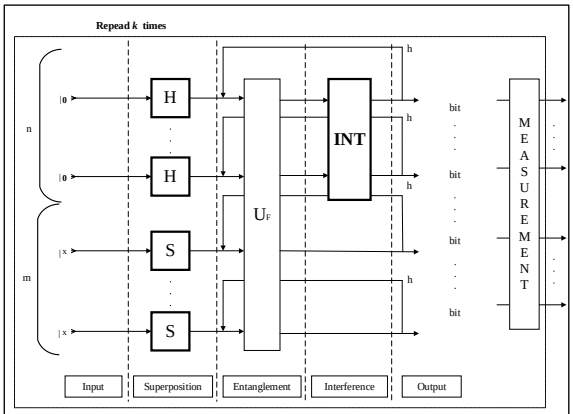


Fig. 4. The structure of a quantum circuit

groups are the background for universal QAG design for simulation of a QA's on classical computers. Therefore, the general structure of the QAG is based on three quantum operators as *superposition*, *entanglement*, and *interference*, and measurement is irreversible classical operation.

The QAG acts on an initial canonical basis vector to generate a complex linear combination (called a superposition) of basis vectors as an output. This superposition contains the full information to answer the initial problem. After the superposition has been created, measurement takes place in order to extract the answer information. In quantum mechanics, a measurement is a non-deterministic operation that produces as output only one of the basis vectors in the entering superposition. The probability of every basis vector of being the output of measurement depends on its complex coefficient (probability amplitude) in the entering complex linear combination.

Thus, the segmental action of the quantum gate and of measurement makes up a quantum block (see Fig. 3). The quantum block is repeated *k* times in order to produce a collection of *k* basis vectors. Since measurement is a non-deterministic operation, these basis vectors will not necessarily be identical, and each basis vector encodes a piece of the information needed to solve the problem. The last part of the algorithm involves interpretation of the collected basis vectors in order to get the final answer for the initial problem with some probability.

Peculiarities of general QA-structure

As mentioned above, QA estimates (without numerical computing) the qualitative properties of the function *f*. From a mathematical standpoint, a function *f* is the map of one logical state into another. The problem solved by a QA can be stated in the symbolic form as follows:

Find a certain property of function f that is a map $f:\{0,1\}^n \rightarrow \{0,1\}^m$.

Main goal of QA applications is the study and search of qualitative properties of functions as the solution of problem. Figure 5 shows the general structure of QA.

The main blocks in Fig. 5 are following: i) unified operators; ii) problem-oriented operators; iii) Benchmarks of QA simulation on classical computers; and iv) quantum control algorithms based on quantum fuzzy inference (QFI) and QGA as new types of QSA. The design process of QAG's includes the matrix design form of three quantum operators: superposition (*Sup*), entanglement (U_F) – oracle, and interference (*Int*) that are the background of QA structures.

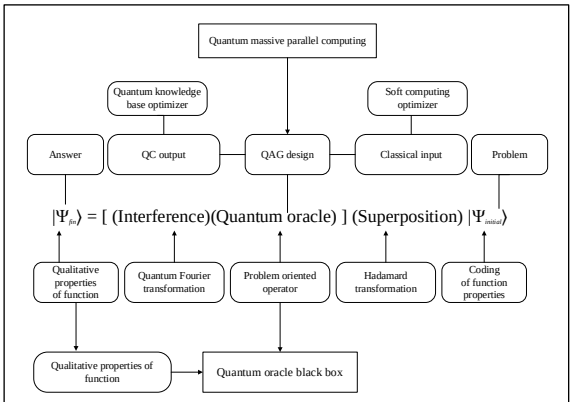


Fig. 5. General structure of QA

In general form, the structure of a QAG for QA in Fig. 5 can be [5] described as follows:

$$QAG = \left[\left(Int \otimes^n I \right) \underbrace{\{U_F\}}_{function\ property} \right]^{h+1} \left[{}^n H \otimes^m S \right], \qquad (1)$$

where *I* is the identity operator; the symbol ⊗ denotes the tensor product; *S* is equal to *I* or *H* and dependent on the problem description. The heart of the quantum block is the quantum gate, which depends on the properties of matrix U_F. One portion of the design process in Eq. (1) is the type-choice of the entanglement problem dependent operator U_F that physically describes the qualitative properties of the function *f*.

A general QA, written as a quantum circuit (as in Fig. 4), can be automatically translated into the corresponding programmable quantum gate for efficient classical simulation. This gate is represented as a quantum operator in matrix form such that, when it is applied to the vector input representation of the quantum register state, the result is the vector representation of the desired register output state.

Main QAG's and main quantum operators

Three quantum operators, superposition, entanglement (quantum oracle), and interference, are the basis for quantum computations of qualitative and quantitative measures in quantum soft computing.

As described above (see Fig. 3) the structure of a QAG based on these three quantum operations of superposition, entanglement, and interference. Thus, superposition, entanglement (quantum oracle) and interference in quantum massive parallel computing are the main operators in QA. The superposition operator of most QA's can be expressed as following: $Sp = \left(\bigotimes_{i=1}^{n} H \right) \otimes \left(\bigotimes_{i=1}^{m} S \right)$, where n and m are the numbers of inputs and of outputs respectively. Operator S may be or Hadamard operator H or identity operator I depending on the algorithm. Numbers of outputs n as well as structures of corresponding superposition and interference operators are presented in the Table 1 for different QA's on Fig. 2.

Table 1

Parameters of superposition and interference operators of main QA

Algorithm	Superposition	*m*	Interference
Deutch's	$H \otimes I$	1	$H \otimes H$
Deutsch–Jozsa's	${}^{n}H \otimes H$	1	${}^{n}H \otimes I$
Grover's	${}^{n}H \otimes H$	1	$D_n \otimes I$
Simon's	${}^{n}H \otimes {}^{n}I$	*n*	${}^{n}H \otimes {}^{n}I$
Shor's	${}^{n}H \otimes {}^{n}I$	*n*	$QFT_n \otimes {}^{n}I$

Figure 6 shows methods in QAG design. The methods as shown in Fig. 6 are based on qualitative measures of QAG design: 1) analysis of QA dynamics and structure gate design; 2) analysis of information flow; and 3) structure simulation of intelligent QA's on classical computers.

Remark. The analysis of information flow in [4, 5] is described. In this article analysis of QA dynamics and structure gate design, and structure simulation of intelligent QA's on classical computers are discussed.

As shown in Fig. 6 analysis of QA dynamics provides the background for showing the existence of a solution and that the solution is unique with the desired probability. Analysis of information flow in the QA gates provides the background for showing that the unique solution exists with the desired accuracy and that the reliability of the solution can be achieved with higher probability.

The intelligence of a QA is achieved through the principle of *minimum information distance*

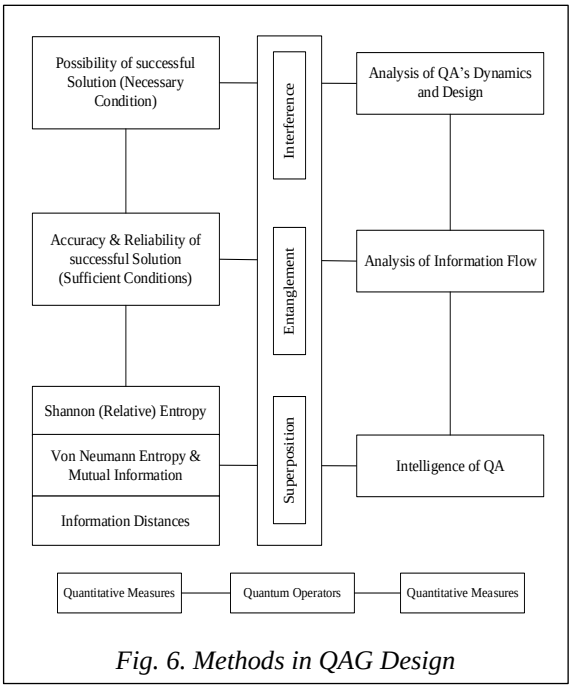


Fig. 6. Methods in QAG Design

between Shannon and von Neumann entropy and includes the solution of the QA stopping problem (see [5]). The output states of a QA as the solution of expected problems are the intelligent states with minimum entropic relations of uncertainty (coherent superposition states). The successful results of QA computing are robust to noise excitations in quantum gates, and intelligent quantum operations are fault-tolerant in quantum soft computing [5].

With the method of quantum gate design presented herein, various different structures of QA can be realized (see Fig. 4), as shown in Table 2 below.

Remark. A quantum computer is difficult to build because of decoherence effects. Decoherence introduces errors in the superposition. The decoherence problem is reduced by using tools of quantum soft computing such as a QGSA. Errors produced by decoherence are of three kinds: (i) phase errors; (ii) bit-flip errors; and (iii) both phase and bit-flip errors.

These three errors can all be modeled using unitary transformations [5].

This means that if the QGSA is implemented on a physical quantum-mechanical system, one would gain the advantages of quantum parallelism and reduce the problem of decoherence, because decoherence can be used as a natural generator of mutation and crossover operators.

Let us discuss briefly any mathematical backgrounds and its physical peculiarities for quantum computing based on QAG.

Table 2

Quantum gate parameters for QA's structure design

Name	**Algorithm**	**Gate Symbolic Form**: $\left[\underbrace{\left(Int \otimes {}^{m}I\right)}_{Interference} \cdot \underbrace{U_F}_{Entanglement}\right]^{h+1} \cdot \left(\underbrace{{}^{n}H \otimes {}^{m}S}_{Superposition}\right)$
Deutsch–Jozsa (*D.–J.*)	$m = 1$ $S = H(x = 1)$ $Int = {}^{n}H$ $k = 1$ $h = 0$	$\left({}^{n}H \otimes I\right) \cdot U_F^{D-J} \cdot \left({}^{n+1}H\right)$
Simon (*Sim*)	$m = n$, $S = I$ $(x = 0) Int = {}^{n}H$ $k = O(n)$ $h = 0$	$\left({}^{n}H \otimes {}^{n}I\right) \cdot U_F^{Sim} \cdot \left({}^{n}H \otimes {}^{n}I\right)$
Shor (*Shr*)	$m = n$ $S = I(x = 0)$ $Int = QFT_n$ $k = O(Poly(n))$ $h = 0$	$\left(QFT_n \otimes {}^{n}I\right) \cdot U_F^{Shr} \cdot \left({}^{n}H \otimes {}^{n}I\right)$
Grover (*Gr*)	$m = 1$ $S = H(x = 1)$ $Int = D_n$ $k = 1$ $h = O(2^{n/2})$	$\left(D_n \otimes I\right) \cdot U_F^{Gr} \cdot \left({}^{n+1}H\right)$

Design technology of and QAG simulation system

The searching problem can be stated in terms of a list $\mathcal{L}[0,1,\ldots,N-1]$ with a number N of unsorted elements. Denote by x_0 the marked element in $\mathcal{L}$ that are sought. The quantum mechanical solution of this searching problem goes through the preparation of a quantum register in a quantum computer to store the N items of the list. This will allow exploiting quantum parallelism. Thus, assume that the quantum registers are made of n source qubits so that $N = 2^n$.

A target qubit is used to store the output of function evaluations or calls. To implement the quantum search, construct a unitary operation that discriminates between the marked item x_0 and the rest. The following function:

$$f_{x_0}(x) = \begin{cases} 0, \ if \ x \neq x_0 \\ 1, \ if \ x = x_0 \end{cases}, \quad (2)$$

and its corresponding unitary operation $U_{f_{x_0}} |x\rangle |y\rangle = |x\rangle |y \oplus f_{x_0}(x)\rangle$. It is assumed the access to f via the following quantum oracle: $U_f|0,0\rangle = |0, f(0)\rangle$, $U_f|1,0\rangle = |0, f(1)\rangle$.

After these two queries, we can measure qubit 1 with a deterministic outcome, and answer whether $f(0) = f(1)$. However, a quantum checker can apply U_f to a linear combination of states in the computational basis. Count how many applications of this operation or oracle calls are needed to find the item. The rationale behind the Grover algorithm is: 1) to start with a quantum register in a state where the computational basis states are equally present; 2) to apply several unitary transformations to produce an outcome state in which the probability of catching the marked state $|x_0\rangle$ is large enough.

Let's imagine the stages of Gruver's algorithm:

Step 1. Initialize the quantum registers to the state:

$$|\psi_1 = input\rangle := |00\ldots0\rangle|1\rangle. \quad (3)$$

Step 2. Apply bit-wise the Hadamard one-qubit gate to the source register, so as to produce a uniform superposition of basis states in the source register, and also to the target register:

$$|\psi_2\rangle := U_H^{\otimes(n+1)} |\psi_1\rangle = \frac{1}{2^{(n+1)/2}} \sum_{x=0}^{2^n-1} |x\rangle \sum_{y=0,1} (-1)^y |y\rangle. \quad (4)$$

Step 3. Apply the operator $U_{f_{x_0}}$:

$$|\psi_3\rangle := U_{f_{x_0}} |\psi_2\rangle = \frac{1}{2^{(n+1)/2}} \sum_{x=0}^{2^n-1} (-1)^{f_{x_0}(x)} |x\rangle \sum_{y=0,1} (-1)^y |y\rangle.$$

Let U_{x_0} be the operator by

$$U_{x_0} |x\rangle := \left(1 - 2|x_0\rangle\langle x_0|\right)|x\rangle = \begin{cases} |x\rangle, & if \ x \neq x_0 \\ -|x_0\rangle, & if \ x = x_0 \end{cases}, \quad (5)$$

that is, it flips the amplitude of the marked state leaving the remaining source basis states unchanged. The state in the source register of Step 3 equals precisely the result of the action of U_{x_0}, i.e., $|\psi_3\rangle = \left(\left(1 - 2|x_0\rangle\langle x_0|\right) \otimes 1\right)|\psi_2\rangle$.

Step 4. Apply next the operation D known as inversion about the average. This operator is defined as follows

$$D := -\left(U_H^{\otimes n} \otimes I\right) U_{f_0} \left(U_H^{\otimes n} \otimes I\right),$$

and

$$|output\rangle = D|\psi_3\rangle$$

where U_{f_0} is the operator in Step 3 for $x_0 = 0$. The effect of this operator on the source is to transform

$$\sum_x \alpha_x |x\rangle \mapsto \sum_x \left(-\alpha_x + \langle\alpha|x\rangle\right), \quad (6)$$

where $\langle\alpha\rangle := 2^{-n}\sum_x \alpha_x$ is the mean of the amplitudes, so its net effect is to amplify the amplitude of $|x_0\rangle$ over the rest.

Step 5. Iterate Steps 3 and 4 a number of times *m*.

Step 6. Measure the source qubits (in the computational basis). The number *m* is determined such that the probability of finding the searched item x_0 is maximal.

According to Steps 2–4 above and (1), the QAG of Grover's QSA is $G=\left(D_n\otimes I\right)\cdot U_F\left({}^nH\otimes H\right)$ that acts on the initial state of both registers in the QSA.

Computational analysis of Grover's QSA is similar to analysis of the Deutsch–Jozsa QA. The basic component of the algorithm is the quantum operation encoded in Steps 3 and 4, which is repeatedly applied to the uniform state $|\psi_2\rangle$ in order to find the marked element. Steps 5 and 6 in Grover's algorithm are also applied in Shor's QSA. Although this procedure resembles the classical strategy, Grover's operation enhances by constructive interference of quantum amplitudes the presence of the marked state.

Computational models of QSA

We have considered in [4] *five* practical approaches to design fast algorithms for the simulation most of known QA's on classical computers:

1. *Matrix based approach*;
2. *Model representations* of quantum operators in fast QA's;
3. *Algorithmic based approach*, when matrix elements are calculated on "demand";
4. *Problem-oriented approach*, where we succeeded to run Grover's algorithm with up to 64 and more qubits with Shannon entropy calculation (up to 1024 without termination condition);
5. *Quantum algorithms with reduced number of operators* (entanglement-free QA, and so on).

Detail description of these approaches is given in [4].

Figure 7 shows the structure description of the QA Benchmark Block.

The efficient implementations of a number of operations for quantum computation include controlled phase adjustment of the amplitudes in the superposition, permutation, approximation of transformations and generalizations of the phase adjustments to block matrix transformations. These operations generalize those used as example in QSA's that can be realized on a classical computer. The application of this approach is applied herein to the efficient simulation on classical computers of the Deutsch QA, the Deutsch–Jozsa QA, the Simon QA, the Shor QA and the Grover QA.

Implementation of a QA is based on a QAG. In the language of classical computing, a quantum computer is programmed by designing a QAG. The prior art reports relatively few such gates because the basic principles underlying the quantum version of programming are in their infancy and algorithms to date have been programmed by ad-hoc techniques.

The problems solved by the QA can be stated as follows:

Input: A function $f:\{0,1\}^n\to\{0,1\}^m$.

Problem: Find a certain property of *f*.

The structure of a quantum operator U_f in QA's as shown in block of Fig. 3 is outlined, with a high-level representation, in the scheme diagram Fig. 1. In Fig. 3 the input of the QA is a function *f* that maps from binary strings into binary strings. This function is represented as a map table, defining for every string its image. The function *f* is encoded according to an *F* – truth table. The function is transformed according to a transform U_f – truth table into a unitary matrix operator U_f depending on *f*'s properties. In some sense, this operator calculates *f* when its input and output strings are encoded into canonical basis vectors of a complex Hilbert space: U_f maps the vector code of every string into the vector code of its image by *f*. A squared matrix U_f on the complex field is unitary if and only if (iff) its inverse matrix coincides with its conjugate transpose: $U_F^{-1}=U_F$. A unitary matrix is always reversible and preserves the norm of vectors.

Figure 8 shows structure of the quantum block from Fig. 3.

In the structure, the matrix operator U_F has been generated it is embedded into a quantum gate as a QAG, a unitary matrix whose structure depends on the form of matrix U_F and on the problem to be solved. In the QA, the QAG acts on an initial canonical basis vector (which can always choose the same vector) in order to generate a complex linear combination (superposition) of basis vectors as output. This superposition contains all the information to answer the initial problem.

After this superposition has been created, in measurement block takes place in order to extract this information. In quantum mechanics, measurement is a non-deterministic operation that produces as output only one of the basis vectors in the entering superposition. The probability of every basis vector of being the output of measurement depends on its complex coefficient (probability

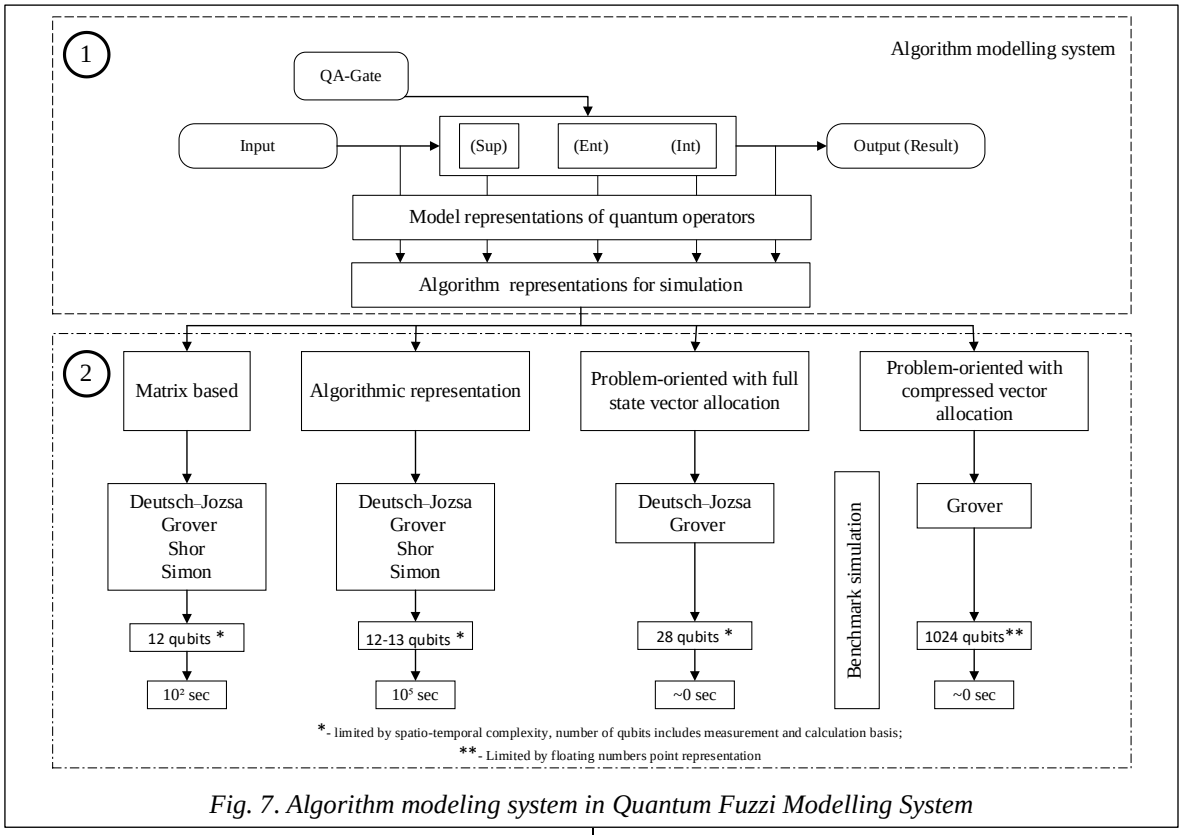


Fig. 7. Algorithm modeling system in Quantum Fuzzi Modelling System

amplitude) in the entering complex linear combination.

The segmental action of the QAG and of measurement characterizes the quantum block in Fig. 8. The quantum block is repeated k times in order to produce a collection of k basis vectors. Since measurement a nondeterministic operation, these basic vectors are not be necessarily identical and each one of them will encode a piece of the information needed to solve the problem. The collection block in Fig. 8 of the algorithm outputs the interpretation of the collected basis vectors in order to get the answer for the initial problem with a certain probability.

Encoder

The behavior of the encoder in Fig. 3 is described in the scheme diagram of Fig. 9.

Function f is encoded into matrix U_F in three steps.

In *step* 1, the map table (f – truth table) of function $f:\{0,1\}^n \to \{0,1\}^m$ is transformed into the map table (F – truth table) of the injective function $F:\{0,1\}^{n+m} \to \{0,1\}^{n+m}$ such that:

$$F(x_0,..,x_{n-1},y_0,..,y_{m-1}) =$$
$$= (x_0,..,x_{n-1},\, f(x_0,..,x_{n-1}) \oplus (y_0,..,y_{m-1})).$$

Remark. The need to deal with an injective function comes from the requirement that U_F is unitary. A unitary operator is reversible, so it cannot map two different inputs in the same output. Since U_F will be the matrix representation of F, F is injective. If one directly employed the matrix representation of function f, one could obtain a non-unitary matrix, since f could be non-injective. So, injectivity is fulfilled by increasing the number of bits and considering function F instead of function f. The function f can be calculated from F by putting $(y_0, \ldots, y_{m-1}) = (0, \ldots, 0)$ in the input string and reading the last m values of the output string.

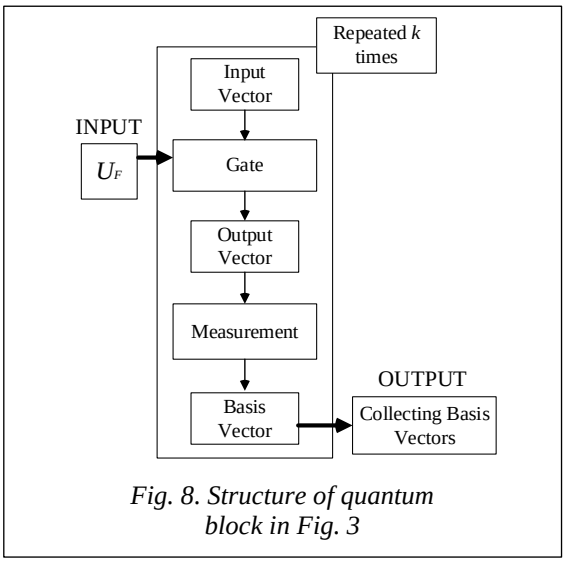


Fig. 8. Structure of quantum block in Fig. 3

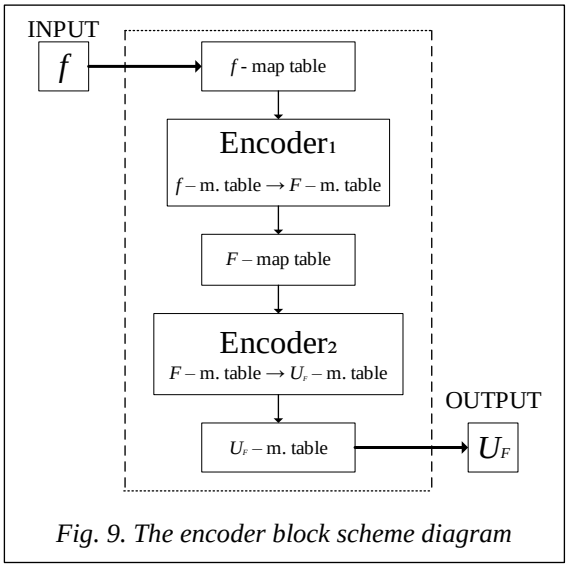


Fig. 9. The encoder block scheme diagram

Reversible circuits realize permutation operations. It is possible to realize any Boolean circuit $F:\mathbb{B}^n \to \mathbb{B}^m$ by reversible circuit. For this case, one need not calculate the function $F:\mathbb{B}^n \to \mathbb{B}^m$. One can calculate another function with expanding $F_\oplus:\mathbb{B}^{n+m} \to \mathbb{B}^{n+m}$ that is defined as following relation: $F_\oplus(x,y)=(x,y\oplus F(x))$ where the operation $\otimes$ is defined as addition on module 2.

Then the value of $F(x)$ is defined as $F_\oplus(x,0)=$ $=(x,F(x))$. For example, the *XOR* operator between two binary strings p and q of length m is a string s of length m such that the i-th digit of s is calculated as the exclusive *OR* between the i-th digits of p and q:

$$p=(p_0,...,p_{n-1}), q=(q_0,...,q_{n-1}); s=p\oplus q=$$
$$=((p_0+q_0)\bmod 2,...,(p_{n-1}+q_{n-1})\bmod 2)).$$

In *step* 2, the function from F map table is transformed into U_f map table, according to the following constraint:

$$\forall s\in\{0,1\}^{n+m}: U_F[\tau(s)]=\tau[F(s)]. \quad (7)$$

The code map

$$\tau:\{0,1\}^{n+m}\to C^{2^{n+m}}\begin{pmatrix}C^{2^{n+m}}\ \textit{is the}\ \text{target}\\ \textit{Complex Hilbert Space}\end{pmatrix}$$

is such that:

$$\tau(0)=\begin{pmatrix}1\\0\end{pmatrix}=|0\rangle,\ \tau(1)=\begin{pmatrix}0\\1\end{pmatrix}=|1\rangle;$$
$$\tau(x_0,...,x_{n+m-1})=\tau(x_0)\otimes...\otimes\tau(x_{n+m-1})=$$
$$=|x_0...x_{n+m-1}\rangle.$$

Code τ maps bit values into complex vectors of dimension 2 belonging to the canonical basis of $\mathbb{C}^2$. Besides, using tensor product, τ maps the general state of a binary string of dimension n into a vector of dimension 2^n, reducing this state to the joint state of the n bits composing the register. Every bit state is transformed into the corresponding 2-dimesional basis vector and then the string state is mapped into the corresponding 2^n-dimesional basis vector by composing all bit-vectors through tensor product. In this sense tensor product is the vector counterpart of state conjunction. Basis vectors are denoted using the *ket* notation $|i\rangle$. This notation is taken from Dirac description of quantum mechanics.

In *step* 3, the U_F map table is transformed into U_F using the following transformation rule:

$$[U_F]_{ij}=1\Leftrightarrow U_F|j\rangle=|i\rangle.$$

This rule can be understood by considering vectors $|i\rangle$ and $|j\rangle$ as column vectors. These vectors belong to the canonical basis, where U_F defines a permutation map of the identity matrix rows. In general, row $|j\rangle$ is mapped into row $|i\rangle$.

Quantum block

The heart of the quantum block is the quantum gate, which depends on the properties of matrix U_F. The quantum block uses the QAG, which depends on the properties of matrix U_F. The structure of a quantum operator U_F in QA's as shown in Fig. 3 is outlined, with a high-level representation, in the scheme diagram of Fig. 8.

The scheme in Fig. 8 gives a more detailed description of the quantum block. The matrix operator U_F of Fig. 9 is the output of the encoder block represented in Fig. 3.

Here, it becomes the input for the quantum block. This matrix operator is embedded into a more complex gate: the gate G (QAG). Unitary matrix G is applied k times to an initial canonical basis vector $|i\rangle$ of dimension 2^{n+m}. Each time, the resulting complex superposition $G\,|0\ldots01\ldots1\rangle$ of basis vectors is measured in measurement block, producing one basis vector $|x_i\rangle$ as result. The measured basis vectors $\{x_1, \ldots, x_k\}$ are collected together in block of basis vectors.

This collection is the output of the quantum block. The "intelligence" of the QA's is in the ability to build a QAG that is able to extract the information necessary to find the required property of f and to store it into the output vector collection.

In order to represent QAG's it is useful to employ some diagrams called quantum circuits, as shown in Fig. 4. Each rectangle is associated with a matrix $2^n \times 2^n$, where n is the number of lines entering and leaving the rectangle. For example,

the rectangle marked U_F is associated with the matrix U_F.

Using a high-level description of the gate and, using transformation rules shown in Fig. 10, it is possible to compile the corresponding gate-matrix.

Decoder

The decoder block of Fig. 3 interprets the basis vectors (collected in block basis vectors) of after the iterated execution in the quantum block. Decoding these vectors involves retranslating them into binary strings and interpreting them directly in decoder block if they already contain the answer or use them, for instance as coefficients vectors for some equation system, in order to get the searched solution.

Grover's Problem statement

Grover's quantum searching problem is stated as following:

Input: Given a function $f:\{0,1\}^n \to \{0,1\}$ such that

$$\exists x \in \{0,1\}^n : \begin{pmatrix} f(x) = 1 \wedge \forall y \in \{0,1\}^n : \\ x \neq y \Rightarrow f(y) = 0 \end{pmatrix}.$$

Problem: Find x.

Figure 11 shows the definition of the Grover's problem.

Figure 12 shows step design definitions in Grover's QA.

Design process of Grover's QAG

Let us consider the implementation of Grover QSA steps in QAG design.

A. *Introductory example* Consider the case: $n = 2$, $f(01) = 1$. In this case the f map table (see Fig. 9) is defined by:

x	$f(x)$
00	0
01	1
10	0
11	0

Step 1

Function f is encoded into injective function F, built according to the usual statement:

$$F:\{0,1\}^{n+1} \to \{0,1\}^{n+1}; F(x_0, x_1, y_0) =$$
$$= (x_0, x_1, f(x_0, x_1) \oplus y_0).$$

Then the F map table is:

(x_0, x_1, y_0)	$F(x_0, x_1, y_0)$
000	000
010	011
100	100
110	110
001	001
011	011
101	101
111	111

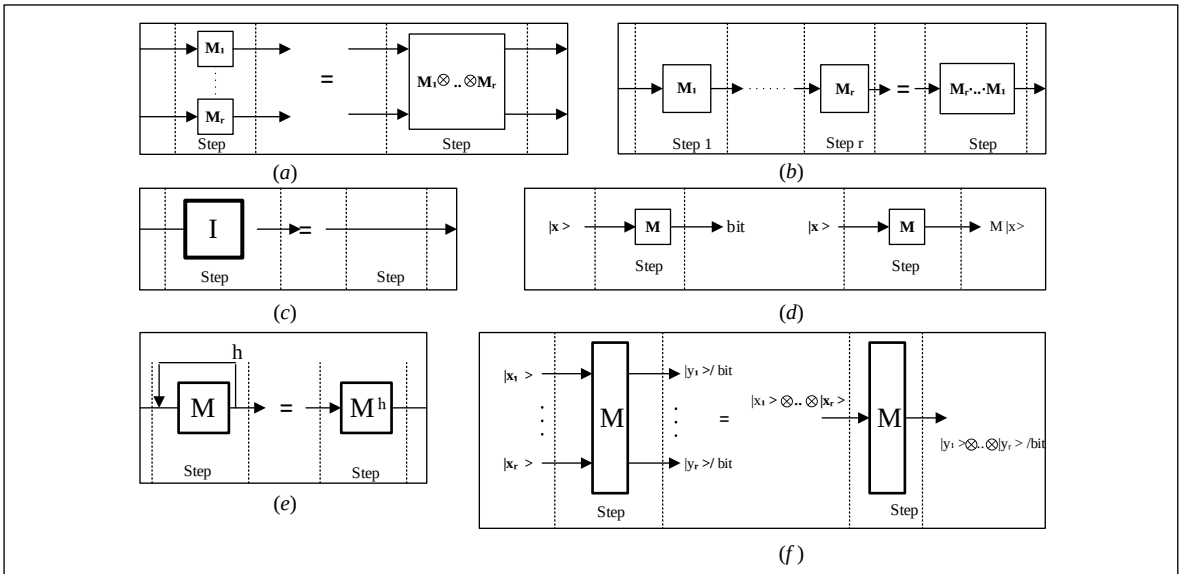


Fig. 10. Transformation rules

These rules are listed in Fig. 10 as following: (a) Rule 1 – Tensor Product Transformation; (b) Rule 2 – Dot Product Transformation; (c) Rule 3 – Identity Transformation; (d) Rule 4 – Propagation Rule; (e) Rule 5 – Iteration Rule; and (f) Rule 6 – Input/Output Tensor Rule. It will be clearer how to use these rules when we afford the first examples of quantum algorithm

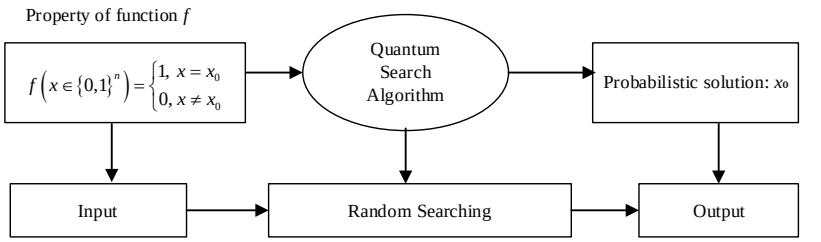


Fig. 11. Grover's QA: Problem definition

Step 2

Now encode F into the map table of U_F using the usual rule: $\forall s \in \{0,1\}^{n+1} : U_F\left[\tau(s)\right] = \tau\left[F(s)\right]$ where τ is the code map defined in above. This means:

$\|x_0\ x_1\ y_0\rangle$	$U_F\|x_0\ x_1\ y_0\rangle$
|000>	|000>
|010>	|011>
|100>	|100>
|110>	|110>
|001>	|001>
|011>	|011>
|101>	|101>
|111>	|111>

Step 3

From the map table of U_F calculate the corresponding matrix operator. This matrix is obtained using the rule: $\left[U_F\right]_{ij} = 1 \Leftrightarrow U_F \left| j \right\rangle = \left| i \right\rangle$.

U_F is thus:

U_F	|00>	|01>	|10>	|11>
|00>	I	0	0	0
|01>	0	C	0	0
|10>	0	0	I	0
|11>	0	0	0	I

The effect of this matrix is to leave unchanged the first and the second input basis vectors of the input tensor product, flipping the third one when the first vector is |0> and the second is |1>. This agrees with the constraints on U_F stated above.

B. *General case with n* = 2. Now take into consideration the more general case: $n = 2$, $f(\underline{x}) = 1$.

The corresponding matrix operator is:

U_F	|00>	|01>	|10>	|11>
|00>	M_{00}	0	0	0
|01>	0	M_{01}	0	0
|10>	0	0	M_{10}	0
|11>	0	0	0	M_{11}

with $M_{\underline{x}} = C \wedge \forall i \neq \underline{x} : M_i = I$

C. *General case* It is relatively simple now to generalize operator U_F from the case $n = 2$ to the case $n > 1$. The operator C is found on the main diagonal of the block matrix, in correspondence of the celled labeled by vector $|\underline{x}\rangle$, where $\underline{x}$ is the binary string having image one by f. Therefore:

U_F	|00>	|01>	…	|11>
|00>	M_{00}	0	…	0
|01>	0	M_{01}	…	0
…	…	…	…	…
|11>	0	0	…	M_{11}

with $M_{\underline{x}} = C \wedge \forall i \neq \underline{x} : M_i = I$

Quantum block

The matrix U_F, the output of the encoder, is embedded into the QAG.

N	Design step definition
0	**Step 0: Encoder** Step 0.1: Injective function F building Step 0.2: Preparation of map table for entanglement operator U_F
1	**Step 1: Preparation of quantum operators** Step 1.1: Preparation of superposition operator Step 1.2: Preparation of entanglement operator using information from step 0.2 Step 1.3: Preparation of interference operator Step 1.4: Quantum gat
2	**Step 2: Algorithm execution** Step 2.1: Application of superposition operator Step 2.2: Application of entanglement operator Step 2.3: Application of interference operator Step 2.4: Repeat steps 2.2 and 2.3 **h** times Step 2.5: Measurement and interpretation of the output

Fig. 12. Grover's QA: Steps of the algorithm design

Encoder: In order to make the discussion more comprehensible, it is convenient to first consider a special function with n = 2, then the general case with n = 2 is discussed, and finally to analyze the general case with n > 0

This gate is described in Fig. 13a, using a quantum circuit of Grover QSA.

Operator D_n is called a "*diffusion matrix*" of order n and it is responsible for interference in this algorithm. It plays the same role as the QFT_n in Shor's algorithm and of ${}^nH$ in Deutsch–Jozsa's and Simon's algorithms. This matrix is defined as:

D_n	|0..0>	|0..1>	…	|*i*>	…	|1..0>	|1..1>
|0..0>	$-1+1/2^{n-1}$	$1/2^{n-1}$	…	$1/2^{n-1}$	…	$1/2^{n-1}$	$1/2^{n-1}$
|0..1>	$1/2^{n-1}$	$-1+1/2^{n-1}$	…	$1/2^{n-1}$	…	$1/2^{n-1}$	$1/2^{n-1}$
…	…	…	…	…	…	…	…
|*i*>	$1/2^{n-1}$	$1/2^{n-1}$	…	$-1+1/2^{n-1}$	…	$1/2^{n-1}$	$1/2^{n-1}$
…	…	…	…	…	…	…	…
|1..0>	$1/2^{n-1}$	$1/2^{n-1}$	…	$1/2^{n-1}$	…	$-1+1/2^{n-1}$	$1/2^{n-1}$
|1..1>	$1/2^{n-1}$	$1/2^{n-1}$	…	$1/2^{n-1}$	…	$1/2^{n-1}$	$-1+1/2^{n-1}$

Using Rule 3 from Fig. 10, compile the previous circuit into the circuit presented as in the Fig. 13, b, and then into the circuit of Fig. 13c and using Rule 2 in Fig. 10 design on Fig. 13d.

Computer design process of Grover's QAG and simulation results

Consider the design process of Grover's QAG according to the steps represented in Fig. 12.

Figure 14 shows Step 0, the encoding process, for the case of order $n = 3$ and answer search 1.

Preparation of quantum entanglement (step 1.2 from Fig. 12) for the one answer search is shown in Fig. 15.

The cases for 2 and 3 answer searches if the preparation of the entanglement operator is shown by the link http://www.swsys.ru/uploaded/image/2023-4/17.jpg.

Figure 16 shows the result of interference operator design (step 1.3 of Fig. 12).

Comparison between superposition and interference operators in Grover's QAG is shown in Fig. 17.

The Grover's QAG assembly (step 1.4 of Fig. 12) is shown by the link http://www.swsys.ru/uploaded/image/2023-4/18.jpg.

The assembled entanglement and interference operators in gate representation (step 1.4 from Fig. 12) are presented by the link http://www.swsys.ru/uploaded/image/2023-4/19.jpg.

Dynamic evolution of successful results of algorithm execution for the first iteration of Grover's QAG for initial qubits state |0001⟩ and different answer search is shown in Fig. 18.

The algorithm execution results for Grover's QSA with different number of iterations for successful results with different searching answer number are presented by the link http://www.swsys.ru/uploaded/image/2023-4/20.jpg.

The results of the execution of the RSA Engraver algorithm with a different number of iterations for successful results with different searching answer number

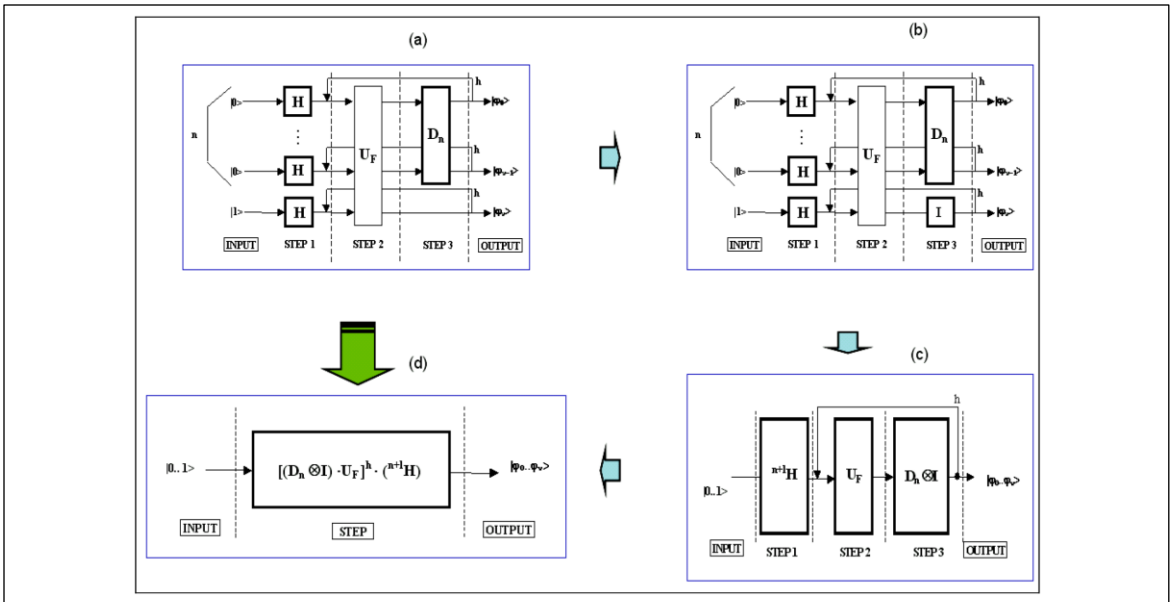


Fig. 13. Grover's quantum algorithm simulation: Circuit representation and corresponding gate design

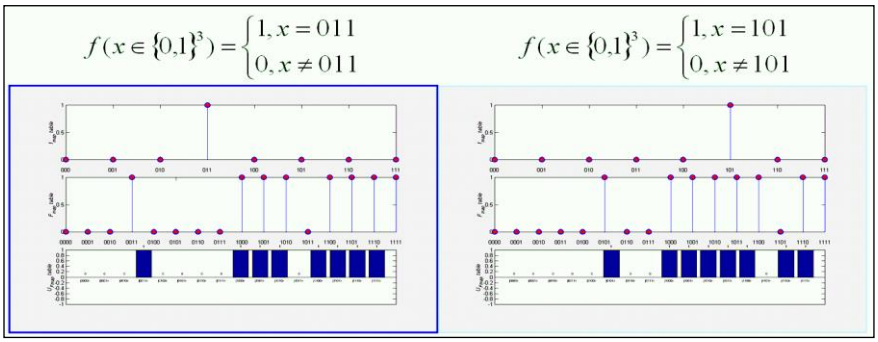


Fig. 14. Grover's QA: Step 0. Encoding: Order n =3, 1 answer search

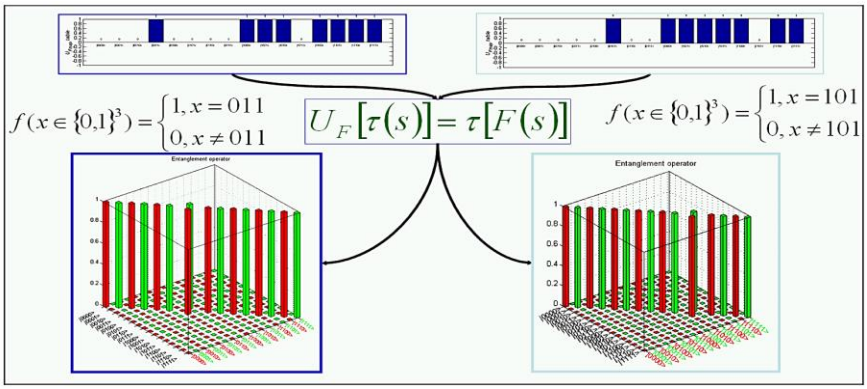


Fig. 15. Grover's QA: Step 1.2. Preparation of quantum operators: Entanglement operators for 1 answer search

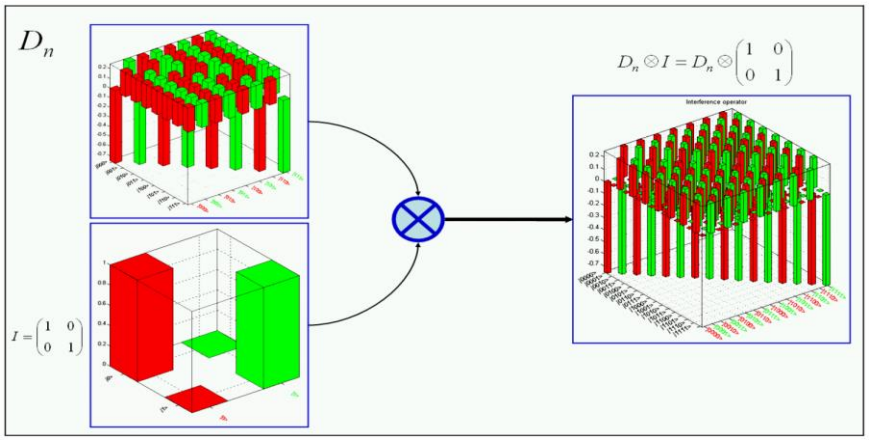


Fig. 16. Grover's QA: Step 1.3. Preparation of quantum operators: Interference operator

Figure 19 is a 3D dynamic representation of Grover's QAG probabilities evolution (step 2 of Fig. 12) for different cases of answer search.

Interpretation of measurement results in simulation of Grover's QSA-QAG

In the case of Grover's QSA this task is achieved (according to the results of this section) by preparing the ancillary qubit of the oracle of the transformation:

$$U_f : |x, a\rangle \mapsto |x, f(x) \oplus a\rangle$$

in the state $|a_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. In this case the operator $I_{|x_0\rangle}$ is computationally equivalent to

$$U_F : U_F\left[|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] =$$

$$= \left[I_{|x_0\rangle}(|x\rangle)\right] \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$= \frac{1}{\sqrt{2}}\underbrace{\left[I_{|x_0\rangle}(|x\rangle)\right]}_{\text{Computation result}} \otimes \underbrace{|0\rangle}_{\text{Measurement}} - \frac{1}{\sqrt{2}}\underbrace{\left[I_{|x_0\rangle}(|x\rangle)\right]}_{\text{Computation result}} \otimes \underbrace{|1\rangle}_{\text{Measurement}}$$

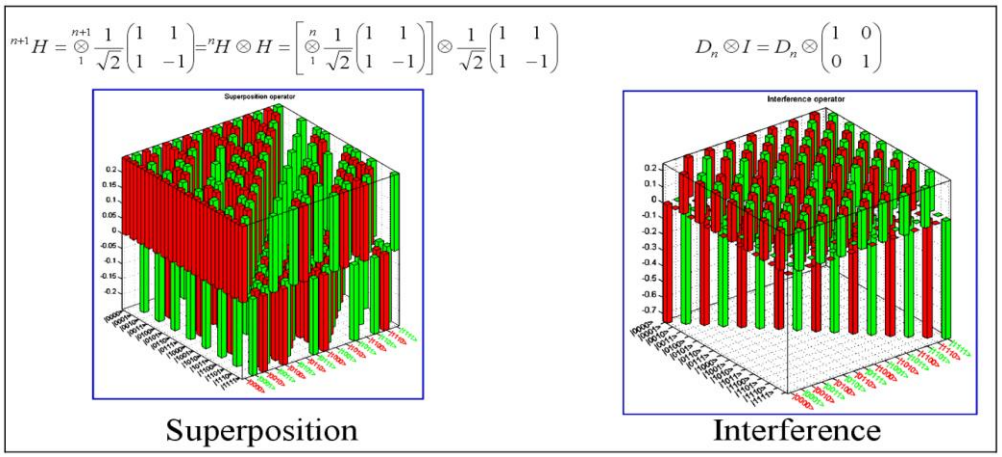


Fig. 17. Grover's QA: Superposition and interference operators

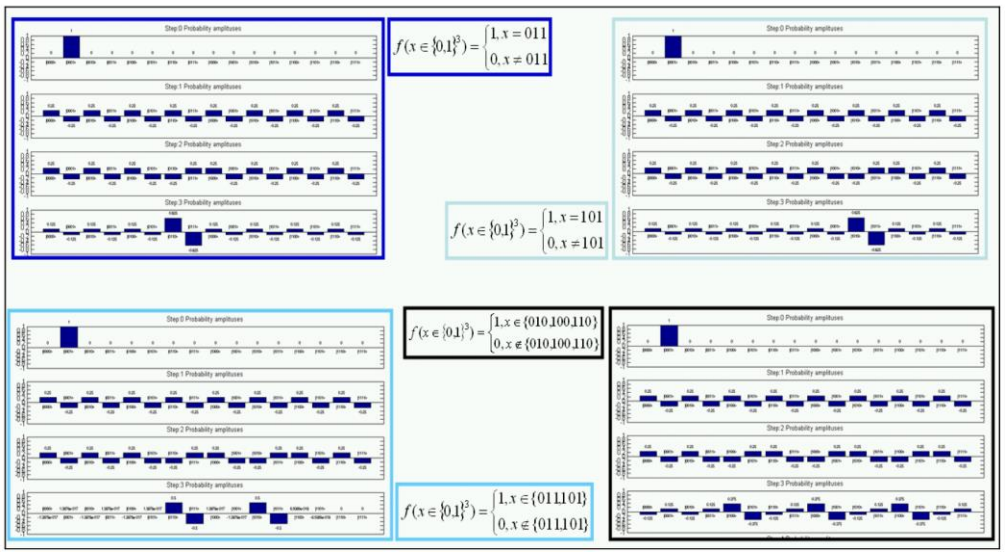


Fig. 18. Grover's QA: Algorithm execution. First Iteration

and the operator U_f is constructed from a controlled $I_{|x_0\rangle}$ and two one qubit Hadamard transformations.

Let's show the interpretation of the Grover QAG results.

If measured basis vector: $\underbrace{\left|x_0^1 \cdots x_0^n x_0^{n+1}\right\rangle}_{n+1\,qubits}$.

Consist of:

$$\underbrace{\left|x_0^1 \cdots x_0^n\right\rangle}_{n\,qubits\,of\,computational\,basis} \otimes \underbrace{\left|x_0^{n+1}\right\rangle}_{1\,qubit\,of\,measurement\,basis}.$$

Then searched argument was:

$$\left.\underbrace{x_0 = x_0^1 \cdots x_0^n}_{n\,bits}\right\} \Rightarrow \begin{matrix} Answer\,of \\ Quantum \\ Searching \end{matrix}.$$

Measured basis vector are computed from the tensor product between the computation qubit results and ancillary measurement qubit. In Grover's searching process, the ancillary qubit does not change during the quantum computing.

As described above, operator U_f is constructed from two Hadamard transformations and the Hadamard transformation H (modeling the constructive interference) applied on the state of the standard computational basis can be seen as implementing a fair coin tossing. Thus, if the matrix

$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is applied to the states of the standard basis, then $H^2|0\rangle = -|1\rangle$, $H^2|1\rangle = |0\rangle$, and therefore H^2 acts in measurement process of computational result as a *NOT*-operation, up to the phase sign. In this case, the measurement basis separated with the computational basis (according to tensor product).

The results of simulation are shown in Fig. 20a. Figure 20b shows the results of computation on a classical computer.

Figure 20b shows two possibilities:

$$\{|0110\rangle\} = |011\rangle \otimes |0\rangle$$

Result Measuremeb qubitt

and

$$\{|0111\rangle\} = |011\rangle \otimes |1\rangle \quad .$$

Result Measuremeb qubitt

A similar situation is shown in Fig. 20b.

Figure 20b demonstrate also two searching marked states:

$$\left\{ |0110\rangle = |011\rangle \otimes |0\rangle \quad or \quad \{|0101\rangle\} = |101\rangle \otimes |0\rangle \right\}$$

Result Measuremeb qubitt Result Measuremeb qubitt

and

$$\left\{ |0111\rangle = |011\rangle \otimes |1\rangle \quad or \quad \{|1011\rangle\} = |101\rangle \otimes |1\rangle \right\}$$

Result Measuremeb qubitt Result Measuremeb qubitt

A similar situation is shown for three searching marked states in Fig. 20b.

Using a random measurement strategy based on a fair coin tossing in the measurement basis $\{|0\rangle, |1\rangle\}$ one can independently receive with certainty the searched marked states from the measurement basis result. Figure 20c show accurate results of searching of corresponding marked states. Final results of interpretation for Grover's algorithm are shown in Fig. 19. The measurement results based on a fair coin tossing measurement are shown in Fig. 20c.

Figure 20c shows that for both possibilities in implementing a fair coin tossing type of measurement process the search for the answer are successful and demonstrate the possibility of the effectiveness of quantum algorithm simulator realization on classical computer.

Related problems of QA classical simulation in [6–12] discussed.

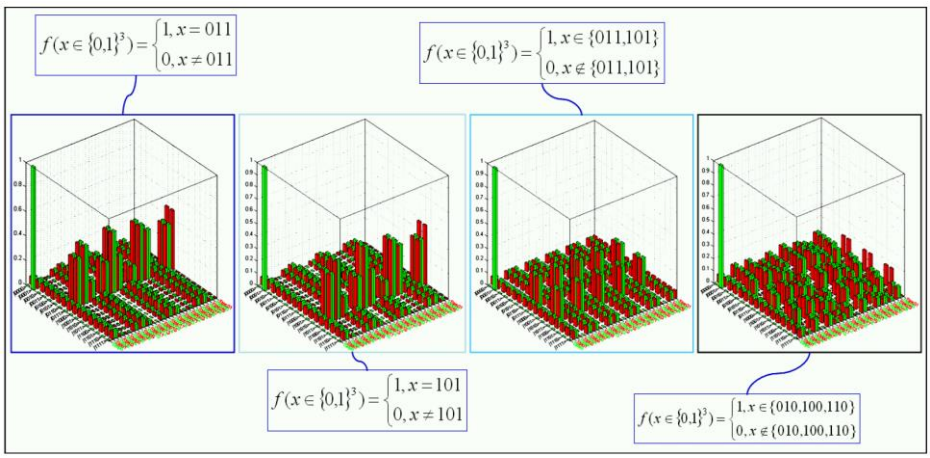


Fig. 19. Grover's QA: Step 2. Algorithm execution 3D dynamics: Probabilities

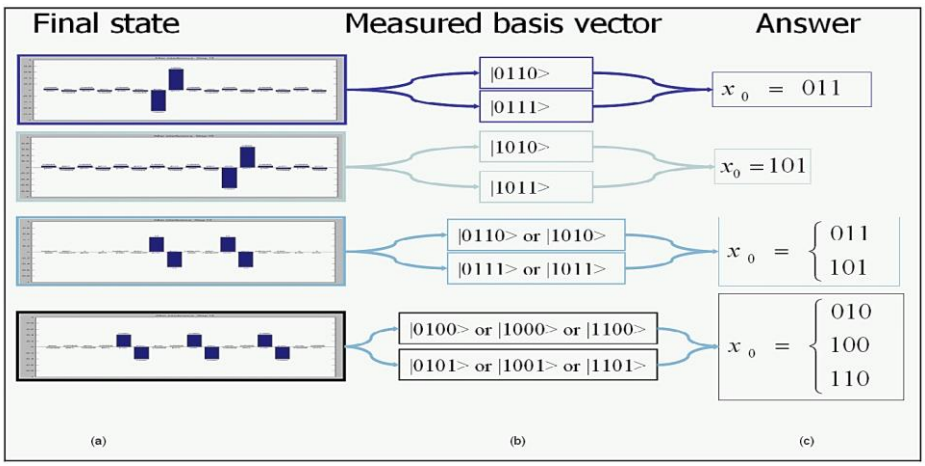


Fig. 20. Grover's QA: Results of algorithm

Conclusions

General approach to design of quantum algorithm gates is described. Gate-based quantum algorithm representation for effective simulation on computer with classical architecture demonstrated. Grover's quantum search algorithm is explained in detail along with implementations on a local computer simulator.

References

1. Grover, L.K. (2001) *A Fast Quantum Mechanical Algorithm for Database Search*, USA, Pat. US 6,317,766-B1.
2. Nielsen, M.A., Chuang, I.L. (2000) *Quantum Computation and Quantum Information*. UK: Cambridge University Press, 676 p.
3. Serrano, M.A., Perez-Castillo, R., Piattini, M. (2022) *Quantum Software Engineering*. Springer Verlag Publ., 330 p. doi: 10.1007/978-3-031-05324-5.
4. Ivancova, O.V., Korenkov, V.V., Ulyanov, S.V., Zrelov, P.V. (2022) *Quantum Software Engineering Toolkit. Pt. 1*. Moscow, 464 p.
5. Ulyanov, S.V. (2003) *System and Method for Control Using Quantum Soft Computing*, USA, Pat. US-6578018-B1.
6. Nyman, P. (2022) 'Simulation of quantum algorithms with a symbolic programming language', *ArXiv*, art. 0705.3333v2, available at: https://arxiv.org/abs/0705.3333v2 (accessed May 24, 2023).
7. Juliá-Díaz, B., Burdis, J.M., Tabakin, F. (2009) 'QDENSITY - A Mathematica quantum computer simulation', *Comput. Phys. Communications*, 180(3), pp. 474. doi: 10.1016/j.cpc.2008.10.006.
8. Johansson, N., Larsson, A.-K. (2017) 'Efficient classical simulation of the Deutsch–Jozsa and Simon's algorithms', *Quantum Inf. Process*, 16(9), art. 233. doi: 10.1007/s11128-017-1679-7.
9. Childs, A.M., Coudron, M., Gilani, A.Sh. (2022) 'Quantum algorithms and the power of forgetting', *ArXiv*, art. 2211.12447v2, available at: https://arxiv.org/abs/2211.12447v2 (accessed May 24, 2023).
10. Cumming, R., Thomas, T. (2022) 'Using a quantum computer to solve a real-world problem – what can be achieved today?', *ArXiv*, art. 2211.13080v1, available at: https://arxiv.org/abs/2211.13080v1 (accessed May 24, 2023).
11. Abhijith, J., Adedoyin, A., Ambrosiano, J., Anisimov, P. et al. (2022) 'Quantum algorithm implementations for beginners', *ACM Transactions on Quantum Computing*, 3(4), art. 18, pp. 1–92. doi: 10.1145/3517340.
12. Tezuka, H., Nakaji, K., Satoh, T., Yamamoto, N. (2022) 'Grover search revisited; application to image pattern matching', *Phys. Rev. A*, 105, art. 032440.

УДК 512.6, 517.9, 519.6 doi: 10.15827/0236-235X.142.523-538 2023. Т. 36. № 4. С. 523–538

Квантовые вычисления на основе алгоритмических вентилей: проектирование алгоритма квантового поиска Гровера в квантовой программной инженерии

С.В. Ульянов
В.С. Ульянов

Ссылка для цитирования
Ульянов С.В., Ульянов В.С. Квантовые вычисления на основе алгоритмических вентилей: проектирование алгоритма квантового поиска Гровера в квантовой программной инженерии // Программные продукты и системы. 2023. Т. 36. № 4. С. 523–538. doi: 10.15827/0236-235X.142.523-538
Информация о статье
Поступила в редакцию: 23.03.2023 После доработки: 08.06.2023 Принята к публикации: 21.08.2023

Аннотация. Отличие классического алгоритма от квантового (КА) заключается в следующем: задача, решаемая КА, закодирована в структуре квантовых операторов, применяемых к входному сигналу. Входной сигнал в структуру КА в этом случае всегда один и тот же. Выходной сигнал КА включает в себя информацию о решении закодированной проблемы. В результате КА задается функция для анализа, и КА определяет ее свойство в виде ответа

без количественных вычислений. КА изучает качественные свойства функций. Ядром любого КА является набор унитарных квантовых операторов или квантовых вентилей. На практике квантовый вентиль представляет собой унитарную матрицу с определенной структурой. Размер этой матрицы растет экспоненциально с увеличением количества входных данных, что существенно ограничивает моделирование КА на классическом компьютере с фон-неймановской архитектурой. Модели квантовых поисковых алгоритмов применяются для решения задач информатики, таких как поиск в неструктурированной базе данных, квантовая криптография, инженерные задачи, проектирование систем управления, робототехника, интеллектуальные контроллеры и т.д. Алгоритм Гровера подробно объясняется вместе с реализациями на локальном компьютерном симуляторе. В представленной статье описывается практический подход к моделированию одного из самых известных КА на классических компьютерах – алгоритма Гровера.

Ключевые слова: алгоритмы квантового поиска, квантовые схемы, квантовые алгоритмические вентили, синтез квантовых операторов

Благодарности. Плодотворные дискуссии с В. Корепиным и К. Беннеттом помогли авторам более четко сформулировать полученные в работе результаты

Список литературы

1. Grover L.K. A Fast quantum mechanical algorithm for database search. USA, 2001, Pat. US-6,317,766-B1.
2. Nielsen M.A., Chuang I.L. Quantum computation and quantum information. UK, Cambridge University Press, 2000, 676 p.
3. Serrano M.A., Perez-Castillo R., Piattini M. Quantum software engineering. Springer Verlag Publ., 2022, 330 p. doi: 10.1007/978-3-031-05324-5.
4. Иванцова О.В., Кореньков В.В., Ульянов С.В., Зрелов П.В. Инструментарий квантовой программной инженерии. Ч. 1. М.: Курс, 2022. 464 с. (in Eng.).
5. Ulyanov S.V. System and method for control using quantum soft computing. USA, 2003, Pat. US-6578018-B1.
6. Nyman P. Simulation of quantum algorithms with a symbolic programming language. ArXiv, 2022, art. 0705.3333v2. URL: https://arxiv.org/abs/0705.3333v2 (дата обращения: 24.05.2023).
7. Juliá-Díaz B., Burdis J.M., Tabakin F. QDENSITY – A mathematica quantum computer simulation. Comput. Phys. Communications, 2009, vol. 180, no. 3, pp. 474–474. doi: 10.1016/j.cpc.2008.10.006.
8. Johansson N., Larsson A.-K. Efficient classical simulation of the Deutsch–Jozsa and Simon's algorithms. Quantum Inf. Process, 2017, vol. 16, no. 9, art. 233. doi: 10.1007/s11128-017-1679-7.
9. Childs A.M., Coudron M., Gilani A.Sh. Quantum algorithms and the power of forgetting. ArXiv, 2022, art. 2211.12447v2. URL: https://arxiv.org/abs/2211.12447v2 (дата обращения: 24.05.2023).
10. Cumming R., Thomas T. Using a quantum computer to solve a real-world problem – what can be achieved today? ArXiv, 2022, art. 2211.13080v1. URL: https://arxiv.org/abs/2211.13080v1 (дата обращения: 24.05.2023).
11. Abhijith J., Adedoyin A., Ambrosiano J., Anisimov P. et al. Quantum algorithm implementations for beginners. ACM Transactions on Quantum Computing, 2022, vol. 3, no. 4, art. 18, pp. 1–92. doi: 10.1145/3517340.
12. Tezuka H., Nakaji K., Satoh T., Yamamoto N. Grover search revisited; application to image pattern matching. Phys. Rev. A, 2022, vol. 105, art. 032440.

Авторы
Ульянов Сергей Викторович [1, 2],
д.ф.-м.н., профессор,
ulyanovsv46_46@mail.ru
Ульянов Виктор Сергеевич [3],
к.т.н., доцент,
ulyanovik@mail.ru

[1] Государственный университет «Дубна» –
Институт системного анализа и управления,
г. Дубна, 141980, Россия
[2] Объединенный институт ядерных исследований –
Лаборатория информационных технологий
им. М.Г. Мещерякова, г. Дубна, 141980, Россия
[3] Московский государственный университет
геодезии и картографии (МИИГАиК),
г. Москва, 105064, Россия

Authors
Sergey V. Ulyanov [1, 2],
Dr.Sc. (Physics and Mathematics), Professor,
ulyanovsv46_46@mail.ru
Viktor S. Ulyanov [3],
Cand. of Sci. (Robotics and Mechatronics),
Associate Professor, ulyanovik@mail.ru

[1] Dubna State University –
Institute of System Analysis and Management,
Dubna, 141980, Russian Federation
[2] Joint Institute for Nuclear Research –
Meshcheryakov Laboratory of Information Technologies,
Dubna, 141980, Russian Federation
[3] Moscow State University of Geodesy
and Cartography (MIIGAiK),
Moscow, 105064, Russian Federation

УДК 004.021 doi: 10.15827/0236-235X.142.539-550 2023. Т. 36. № 4. С. 539–550

Поиск решения задачи целочисленного программирования с помощью итеративного округления координат

А.В. Иванов
Ю.Н. Матвеев

Ссылка для цитирования
Иванов А.В., Матвеев Ю.Н. Поиск решения задачи целочисленного программирования с помощью итеративного округления координат // Программные продукты и системы. 2023. Т. 36. № 4. С. 539–550. doi: 10.15827/0236-235X.142.539-550
Информация о статье
Поступила в редакцию: 24.04.2023 После доработки: 02.05.2023 Принята к публикации: 17.05.2023

Аннотация. В статье предлагается алгоритм поиска целочисленного решения, использующий идею округления координат точки оптимального нецелочисленного решения и построения луча, направленного вглубь области допустимого решения. Алгоритм основан на итеративном процессе округления координат точки в направлении построенного луча. В ходе исследования обнаружено, что движение в сторону направления луча без перебора всех возможных вариантов упрощает алгоритм и позволяет избежать ветвления. Это выделяет данный подход из других существующих на данный момент открытых методов, таких как методы отсечений и ветвей и границ. В процессе работы осуществлялись описание и экспериментальная проверка данного алгоритма и возможности его применения при разных конфигурациях области допустимых решений. Теоретическая значимость исследования заключается в разработке нового алгоритма, который не требует выполнения симплекс-метода на каждом этапе и на каждом шаге использует луч вместо плоскости, что предотвращает рост пространственной сложности задачи по сравнению с другими методами. В ходе исследования стало видно, что предложенный алгоритм имеет ограничения, однако основная идея доказала свою работоспособность, и в дальнейшем планируется развивать ее.
Ключевые слова: математическое программирование, линейное программирование, целочисленное программирование, оптимизация, алгоритм, покоординатный спуск

Введение. Задачи *целочисленного программирования* (ЦП) широко применяются в различных отраслях, таких как производство, логистика, экономика, информационные технологии и многих других [1–3]. Их актуальность обусловлена необходимостью оптимизации различных процессов и распределения ресурсов. Являясь частным случаем *линейного программирования* (ЛП) [4], ЦП имеет достаточно долгую историю [5, 6].

В данной работе развивается идея из [7], где предложен алгоритм для поиска целочисленного решения по аналогии с методом покоординатного спуска [8]. Представленный алгоритм, помимо итеративного округления координат на каждом шаге, проверяет выход за *область допустимых решений* (ОДР) и при необходимости возвращает текущее решение на ее границу.

Современное состояние предметной области

Одной из основных проблем при решении задач ЦП до сих пор является их высокая вычислительная сложность. Это обусловлено в основном тем, что точные методы решения данного рода задач, такие как метод отсечений и метод ветвей и границ [9] (также известный как метод Гомори [10]), тратят много вычислительных ресурсов впустую из-за своей природы: решают множество конфигураций одной и той же задачи (тем самым разветвляясь), что экспоненциально увеличивает время выполнения. Кроме того, они полагаются на достаточно затратную операцию нахождения оптимального решения с помощью симплекс-метода [11] на каждом шаге выполнения алгоритма, что приводит к значительным временным затратам.

Для исследования дискретных задач ЦП с логическими переменными имеется метод L-разбиений [12], который позволяет решать комбинаторные задачи, но не получать численные решения. С другой стороны, существуют эвристические алгоритмы, такие как поиск с восхождением к вершине [13] и алгоритм имитации отжига [14], являющийся примером метода Монте-Карло, а также другие подходы с применением алгоритмов машинного обучения [15]. Данные методы часто используются для решения задач ЦП из-за их более высокой скорости выполнения по сравнению с точными алгоритмами. Поскольку данные алгоритмы эвристические, в них изначально заложен элемент случайности, поэтому, используя их, приходится делать некоторые допущения в плане точности.

В одной из недавних работ [16] представлен подход к решению задач математического программирования с помощью R-функционального моделирования на основе построения вок-

сельных моделей геометрических объектов. Данный подход также увеличивает размерность пространства и в большей степени нацелен на задачи нелинейного программирования. В настоящий момент нет точных сведений о его работе с задачами ЦП.

Таким образом, можно сделать вывод о том, что существующие на данный момент методы решения задач ЦП имеют свои недостатки, обусловленные прежде всего выполнением достаточно большого количества излишних операций из-за увеличения размерности пространства решения или ограничения применения. Отличительной особенностью предлагаемого алгоритма является то, что он позволяет выполнить задачу без увеличения размерности пространства решения за счет избегания ввода новых ограничений, что потенциально может ускорить поиск.

Алгоритм поиска решения с помощью итеративного округления координат

В данной статье описывается точный алгоритм для поиска решения задачи ЦП, рассмотрены и экспериментально проверены несколько подходов. Основная идея каждого из них – смещение вглубь ОДР. Это достигается за счет построения луча, смотрящего внутрь ОДР в сторону уменьшения значения целевой функции (при задаче максимизации) и определяющего направление движения при смещении вглубь ОДР. Проверим некоторые предположения, а именно:

– определив новый базис в точке оптимального решения *O*, можно построить луч, который смотрел бы внутрь ОДР; данный луч можно использовать при смещении вглубь ОДР во время поиска целочисленного решения;

– поочередно округляя каждую из координат точки нецелочисленного оптимального решения *O* до ближайшего целого числа, можно найти целочисленное решение внутри ОДР.

Во всех случаях строится луч, выходящий из оптимальной точки *O*, считающейся начальной в данном алгоритме. После этого на построенном луче находится точка *P*, одна из координат которой является целочисленной. Из этой точки начинается итеративный процесс поиска целочисленного решения. Алгоритм состоит из нескольких шагов.

Определение ближайшего базиса

Для построения луча необходимо определить ближайшие смежные ограничения на точку *O*. Для этого необходимо решить уравнение ограничений вида

$$A\bar{x} - \bar{b} = 0, \quad (1)$$

определенное матрицей *A* и вектором $\bar{b}$ в соответствии с уравнением, где координатами $\bar{x}$ являются координаты точки *O*. Выполняющиеся уравнения и будут определять ограничения, которые образуют новый базис.

Допустим, даны следующие ограничения:

$$\begin{cases} -x_1 + 6x_2 \le 15, \\ 5x_1 - x_2 \le 11, \\ x_1 \ge 0, \\ x_2 \ge 0. \end{cases}$$

Тогда можно определить матрицу *A* и вектор $\bar{b}$:

$$A = \begin{pmatrix} 1 & -6 \\ -5 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} -15 \\ -11 \\ 0 \\ 0 \end{pmatrix}.$$

После этого, подставив точку оптимального нецелочисленного решения $O = \left(\frac{81}{29}, \frac{86}{29}\right)$ в уравнение (1), получим

$$AO - \bar{b} = \begin{pmatrix} 1 & -6 \\ -5 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 81/29 \\ 86/29 \end{pmatrix} - \begin{pmatrix} -15 \\ -11 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \frac{81}{29}\begin{pmatrix} 1 \\ -5 \\ 1 \\ 0 \end{pmatrix} + \frac{86}{29}\begin{pmatrix} -6 \\ 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} -15 \\ -11 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} -15 \\ -11 \\ 81/29 \\ 86/29 \end{pmatrix} - \begin{pmatrix} -15 \\ -11 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 \\ 0 \\ 81/29 \\ 86/29 \end{pmatrix}.$$

Видно, что элементы с номерами $k = 1, 2$ являются наименьшими (равны 0), а значит, соответствующие ограничения образуют новый базис в точке *O* (рис. 1):

$$\begin{cases} -x_1 + 6x_2 \le 15 \\ 5x_1 - x_2 \le 11 \end{cases} = \begin{cases} s_1 = 0, \\ s_2 = 0. \end{cases}$$

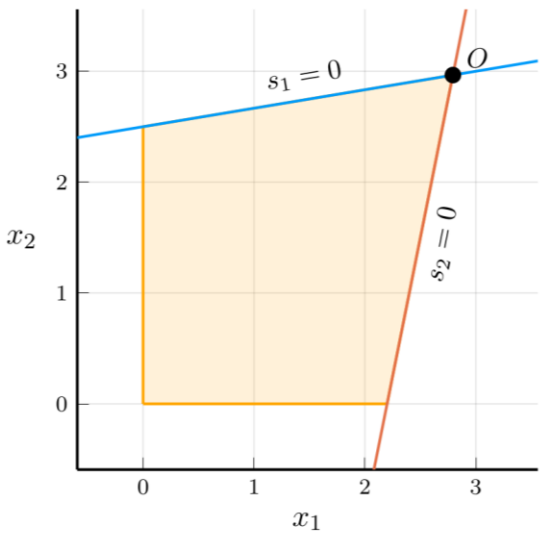


Рис. 1. Смежные ограничения, ближайшие к точке O в исходном базисе x_1x_2

Fig. 1. Intersecting constraints closest to the point O in primal basis x_1x_2

Новый базис s_1s_2 будет представлять собой прямоугольную систему координат с осями s_1 и s_2 соответственно.

Определим прямую функцию перехода от исходного базиса к новому:

$$L(\bar{x}) = M\bar{x} - \overline{b^k} \qquad (2)$$

и соответствующую ей обратную функцию перехода из нового базиса к исходному:

$$L^{-1}(\bar{x}) = M^{-1}(\bar{x} + \overline{b^k}),$$

где $\bar{x}$ – преобразуемая точка в исходном базисе; $\overline{b^k}$ – вектор k-х свободных членов; M – квадратная матрица перехода к новому базису со строками k из матрицы A:

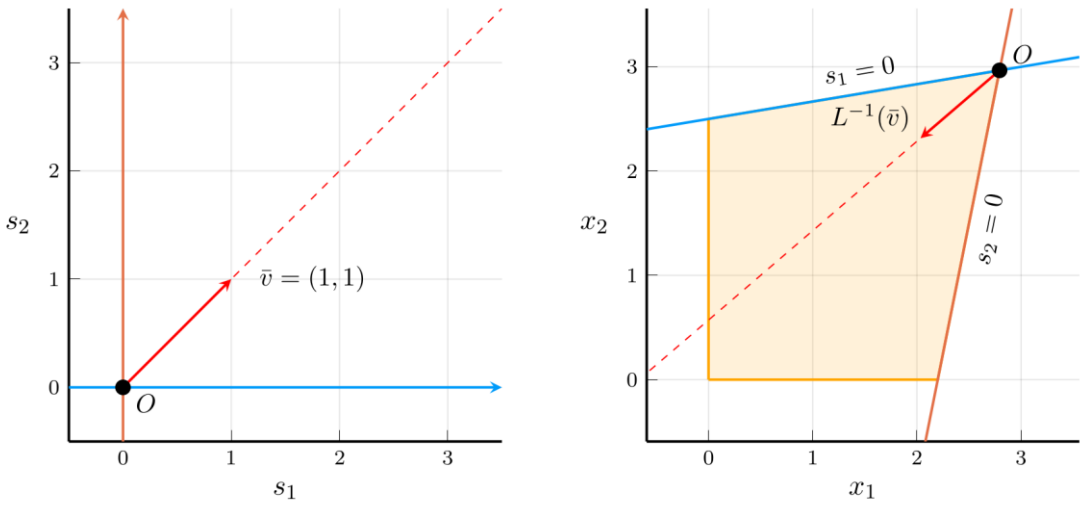
$$A = \begin{pmatrix} 1 & -6 \\ -5 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \end{matrix} \begin{matrix} \leftarrow k=1 \\ \leftarrow k=2 \\ \\ \\ \end{matrix} \rightarrow M = \begin{pmatrix} 1 & -6 \\ -5 & 1 \end{pmatrix},$$

$$\bar{b} = \begin{pmatrix} -15 \\ -11 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} \leftarrow k=1 \\ \leftarrow k=2 \\ \\ \\ \end{matrix} \rightarrow \overline{b^k} = \begin{pmatrix} -15 \\ -11 \end{pmatrix}.$$

Построение луча

Наивный метод. В новом базисе M построим вектор $\bar{v} = (1,1,...,n)$ длиной n, где n – число столбцов матрицы M (равное числу измерений). Данный вектор будет определять луч в новом базисе и делить область между ограничениями пополам (рис. 2а). Зная матрицу M и вектор $\bar{v}$, можно перенести данный луч в исходный базис (рис. 2б), используя уравнение (2). Назовем такой метод построения луча наивным, поскольку простой перенос луча в исходный базис не позволяет получить луч, который делил бы ОДР ровно пополам.

Разделение области между ограничениями ровно пополам. Поскольку наивный метод построения луча не позволяет получить такой



а) *б)*

Рис. 2. Луч в новом (а) и исходном (б) базисах

Fig. 2. Ray in new basis (a) and primal basis (б)

луч, который делил бы область между базисными ограничениями ровно пополам, необходимо модифицировать метод, с помощью которого строится луч. Это становится особенно заметным, если изменить ОДР, сделав ее более узкой.

Дело в том, что масштаб координатной сетки в новом базисе не равен масштабу в старом. Из-за этого, например, вектор (1,1) в исходном базисе будет перекошен в новом базисе и наоборот. Для избежания этого нужно построить вектор $\bar{v}$ таким образом, чтобы его координатами был шаг сетки по каждой из базисных осей (в данном случае – s_1 и s_2).

Шаг сетки базисной оси – это длина нормального вектора ограничения, образующего базис. Допустим, даны ограничения:

$$\begin{cases} 8x_1 - 6x_2 \le 1 \\ -4x_1 + 2x_2 \le 3 \end{cases} = \begin{cases} s_1 = 0, \\ s_2 = 0. \end{cases}$$

Тогда их нормальные векторы равны:

$$\overline{n_{s_1}} = (8, -6),$$

$$\overline{n_{s_2}} = (-4, 2),$$

длины этих векторов соответственно:

$$\left\|\overline{n_{s_1}}\right\| = 10,$$

$$\left\|\overline{n_{s_2}}\right\| = 2\sqrt{5},$$

а вектор, задающий луч в новом базисе, можно определить следующим образом:

$$\hat{v} = \begin{pmatrix} \left\|\overline{n_{s_1}}\right\| \\ \left\|\overline{n_{s_2}}\right\| \end{pmatrix}.$$

Для удобства представления вектор $\hat{v}$ можно нормализовать:

$$\hat{v} = \frac{\bar{v}}{\left|\bar{v}\right|}.$$

Определение точки *P* на луче

Для нахождения точки, из которой будет начинаться поиск, необходимо округлить координаты точки $O = \left(\frac{81}{29}, \frac{86}{29}\right)$ вниз (в задаче максимизации): $O' = \lfloor O \rfloor = (2, 2)$.

После этого через полученную точку нужно провести секущие плоскости $x_1 = 2$, $x_2 = 2$, параллельные осям координат, и найти точки пересечения P_i между плоскостями и лучом (рис. 3). Таким образом, получается набор потенциальных начальных точек внутри ОДР, в каждой из которых одна из координат является целочисленной. Затем необходимо выбрать наилучшую точку, рассчитав значения целевой функции $f(x)$ для каждой точки P_i.

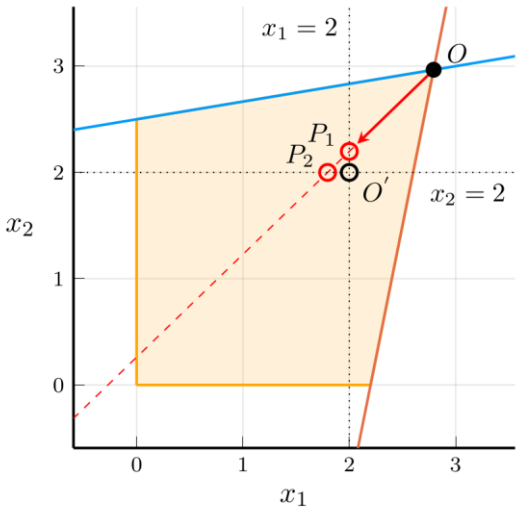


Рис. 3. Исправленный луч, секущие плоскости в точке O′ и потенциальные начальные точки P_i на луче

Fig. 3. Fixed ray, intersecting planes at point O' and potential starting points P_i on the ray

Допустим, задана целевая функция $f(\bar{x}) = 3x_1 + 2x_2$, тогда можно найти ее значения:

$$f(P_1) = 3 \cdot 2 + 2 \cdot \frac{16}{7} = \frac{74}{7} \approx 10.57,$$

$$f(P_2) = 3 \cdot \frac{5}{3} + 2 \cdot 2 = 9.$$

Значение целевой функции для P_1 наибольшее, значит, точка $P = P_1$ будет начальной.

Выбор оси координат для округления

После нахождения новой начальной точки $P = \left(2, \frac{16}{17}\right)$ необходимо определить ось, вдоль которой будет происходить смещение. Поскольку у точки P одна из координат уже является целочисленной, выбирается ось координат, по которой из других осей следует двигаться далее. Так как в данном случае всего две оси (x_1 и x_2) и известно, что координата x_1 точки P уже целочисленная, выбора не остается и смещение будет происходить вдоль оси x_2.

Смещение точки *P* вдоль выбранной оси координат

На первом этапе координата x_1 точки P целочисленная, следовательно, нужно округлять

по следующей доступной координате – x_2. Обозначим точку с округленной координатой как $\lceil P \rceil$. Перенесем точку $\lceil P \rceil$ в новый базис и обозначим как $L(\lceil P \rceil)$. Одна из ее координат (s_2) отрицательная. Это означает, что точка вышла за границы ОДР. Чтобы найти точку на границе ОДР, можно провести прямую между точками P и $L(\lceil P \rceil)$ и найти ее пересечение с осью s_1. Полученную точку обозначим $\lceil P \rceil'$ (рис. 4) и перенесем обратно в исходный базис, обозначив как $L^{-1}(\lceil P \rceil')$ – это будет следующей точкой для поиска решения.

Аналогично округлим начальную точку P вниз. Все ее координаты становятся целочисленными и положительными в новом базисе (рис. 5), следовательно, целочисленное решение для данного примера найдено.

После выполнения данных шагов координаты точки P проверяются на целочисленность. Если хотя бы одна из координат не является целочисленной, все шаги повторяются для новой точки P. Целочисленное решение будет найдено, когда все координаты точки P станут целочисленными. На рисунке 6 представлена общая блок-схема описанного алгоритма.

Непосредственно сама процедура одного смещения точки получает на вход точку P, индекс оси i, вдоль которой происходит смещение, а также направление округления – вверх или вниз. Процедура должна сделать копию точки P в памяти, которую обозначим Q, поскольку в дальнейшем понадобятся измененная точка Q и изначальная точка P. На первом шаге проверяется, является ли i-я координата точки Q целым числом. В случае целого числа

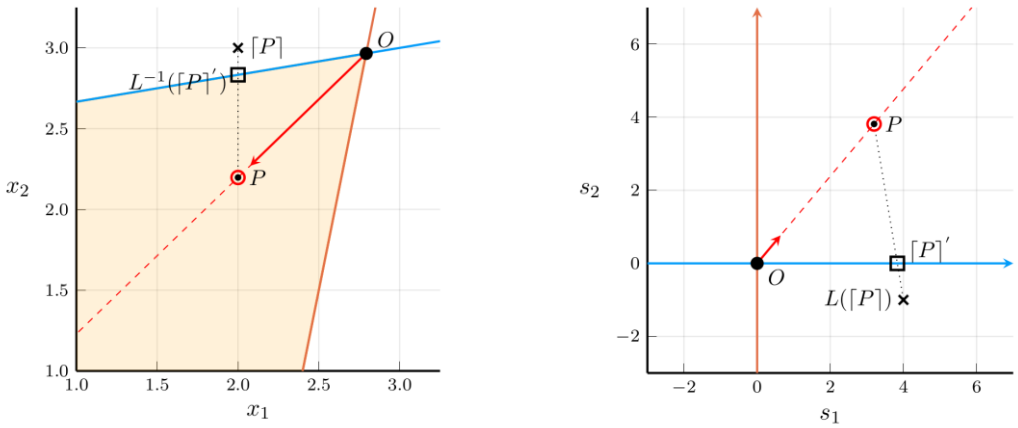


Рис. 4. Округление координаты x_2 точки P вверх и определение ее выхода за ОДР

Fig. 4. Rounding up coordinate x_2 of point P and checking if point P left the feasible region

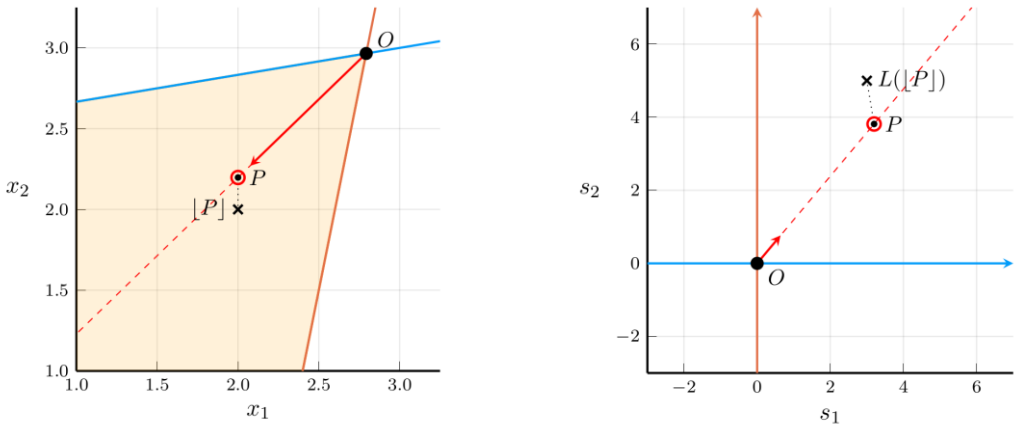


Рис. 5. Округление координаты x_2 точки P вниз и определение ее выхода за ОДР

Fig. 5. Rounding down coordinate x_2 of point P and checking if point P left the feasible region

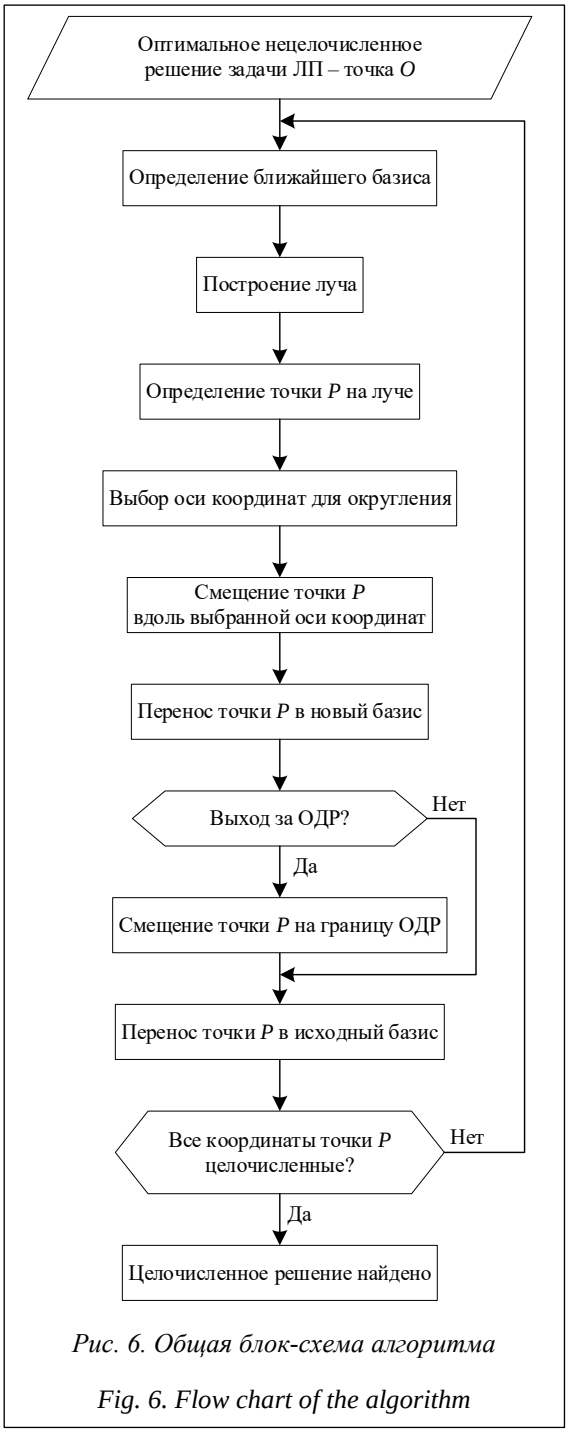


Рис. 6. Общая блок-схема алгоритма

Fig. 6. Flow chart of the algorithm

производится ее смещение на единицу в положительную сторону вдоль оси i в случае округления вверх или же в отрицательную в случае округления вниз. Если i-я координата точки Q не является целым числом, ее значение округляется либо вверх, либо вниз до ближайшего целого. После этого точки P и Q переносятся в новый базис, найденный в начале итерации (на первом шаге алгоритма). Обозначим новые точки как $P_T = L(P)$ и $Q_T = L(Q)$.

Затем проверяются значения координат точки Q_T на неотрицательность. Отрицательное значение указывает на выход за ОДР. В таком случае осуществляется поиск точки пересечения между прямой P_TQ_T и i-й осью координат в новом базисе. Обозначим данную точку как R и перенесем ее в исходный базис: $Q = L^{-1}(R)$. В результате получим точку Q, которая будет либо смещена на единицу вдоль оси i, либо находиться на границе ОДР на данной оси.

Представим процедуру выполнения одного шага алгоритма на языке программирования Julia:

```
function make_step(axis, point, round-
ing_mode = RoundUp)
   search_point = copy(point)
   rounding_up = rounding_mode == RoundUp

   if abs(search_point[axis] -
- round(search_point[axis])) <= 10^-10
      search_point[axis] =
   = search_point[axis] +
   +(rounding_up ? 1 : -1)
   else
      search_point[axis] =
   round(search_point[axis], round-
   ing_mode)
   end

   transformed_point = L(point)
   transformed_search_point =
= L(search_point)

   if minimum(transformed_search_point) <
< 0
      got_out_of_bounds = true

      plane_axis = argmin(trans-
   formed_search_point)
      plane_normal = zeros(length(ba-
   sis_indices))
      plane_normal[plane_axis] = 1.0

      search_point_intersection =
   =intersect_line_with_plane(
         transformed_search_point -
      transformed_point,
         transformed_point,
         plane_normal,
         [0.0, 0.0],
      )

      search_point =
   Linv(search_point_intersection)
   else
      got_out_of_bounds = false
   end

   return (search_point,
got_out_of_bounds)
end
```

Блок-схема данной процедуры изображена на рисунке 7.

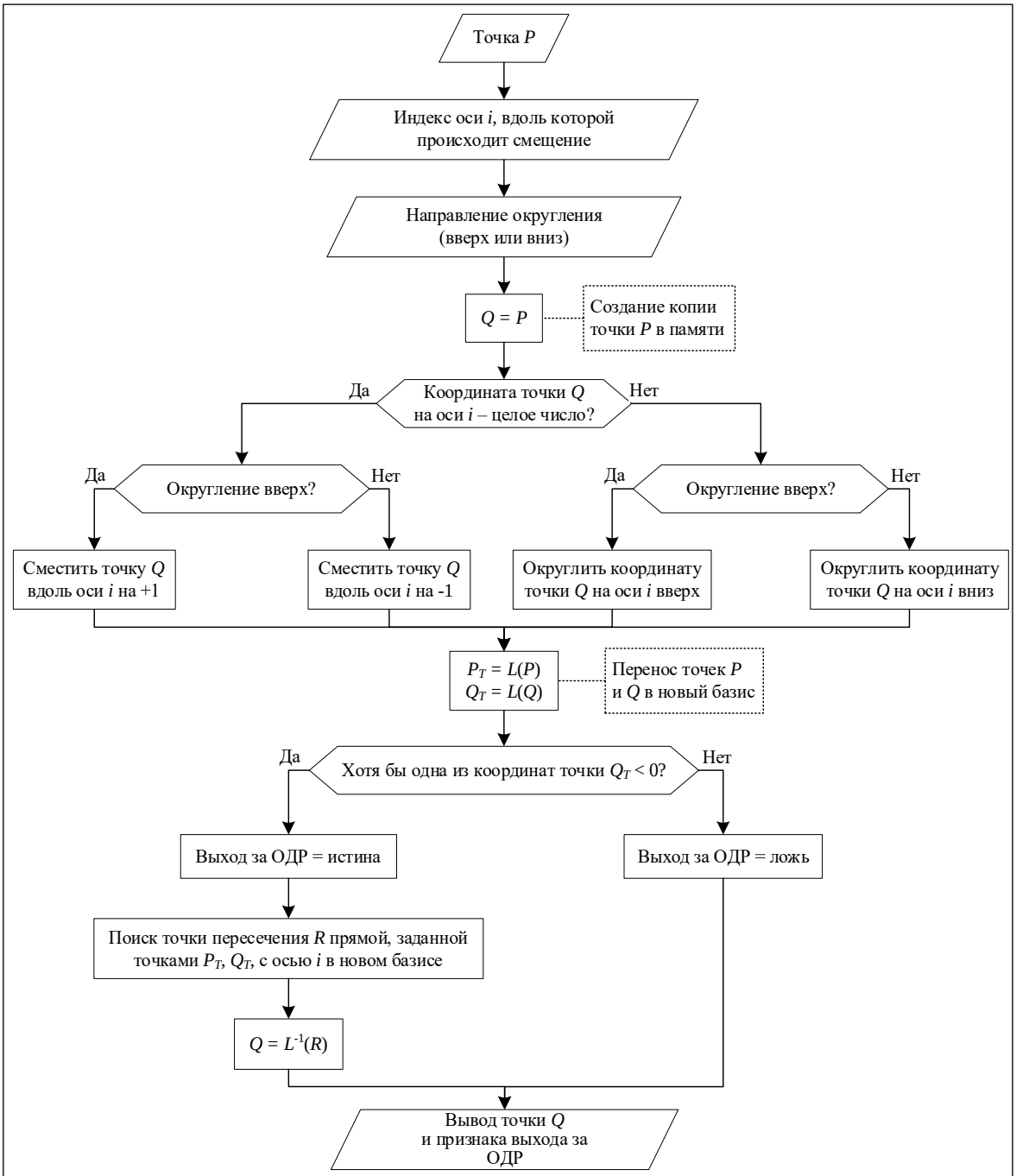


Рис. 7. Блок-схема алгоритма выполнения одного смещения точки

Fig. 7. Flow chart of the algorithm for a single step of the point

Наблюдение 1. Округление в сторону направления луча

Изменим ОДР, сместив и повернув второе ограничение, и аналогичным образом построим луч, направленный вглубь ОДР. Точка O' необязательно должна находиться внутри ОДР, поскольку через нее проводятся прямые, параллельные осям координат и пересекающиеся с лучом, направленным вглубь ОДР. Благодаря этому точка будут находиться внутри ОДР.

Когда луч направлен вниз и влево, округления вверх или вправо, как правило, не дают результата (за исключением, когда базисное ограничение параллельно оси координат).

Таким образом, если луч направлен влево, то есть первая координата вектора $\bar{v}$ отрица-

тельная, точку следует округлять вниз по оси x_1; если же первая координата вектора $\bar{v}$ положительная, соответствующую координату точки необходимо округлять вверх.

Аналогично для оси x_2 стоит руководствоваться знаком второй координаты вектора $\bar{v}$, определяющей луч. Так, если луч направлен вниз, то значение координаты x_2 у него отрицательное, следовательно, округлять координату точки необходимо также вниз. Если луч направлен вверх, то значение координаты x_2 у его вектора положительное и округлять координату точки необходимо вверх. Далее представлен процесс выполнения алгоритма поиска решения (рис. 8).

Исследование алгоритма при узкой ОДР

Изменим задачу так, чтобы ОДР была более узкой, а следовательно, целочисленное решение находилось бы дальше от оптимальной точки O. Из-за этого нужно будет выполнить больше шагов алгоритма, поэтапно округляя точку сначала по одной оси координат, а затем по другой. Используя уже представленный алгоритм, будем выполнять его итеративно до того момента, пока у точки P все координаты не окажутся целочисленными в исходном базисе. На рисунке 9 видно, что точка P как бы отскакивает от ограничений при выходе за пределы ОДР.

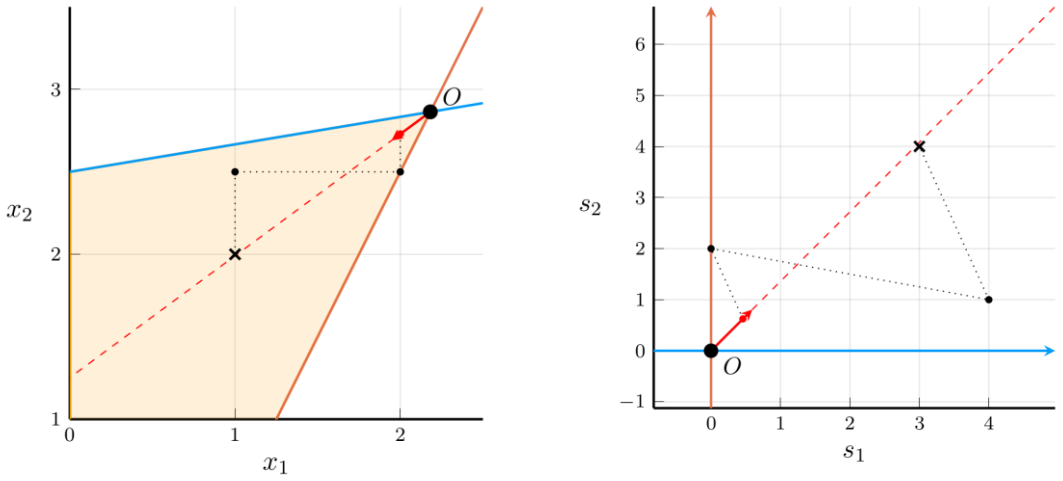


Рис. 8. Перемещение точки в процессе поиска решения

Fig. 8. Point movement during the search of a solution

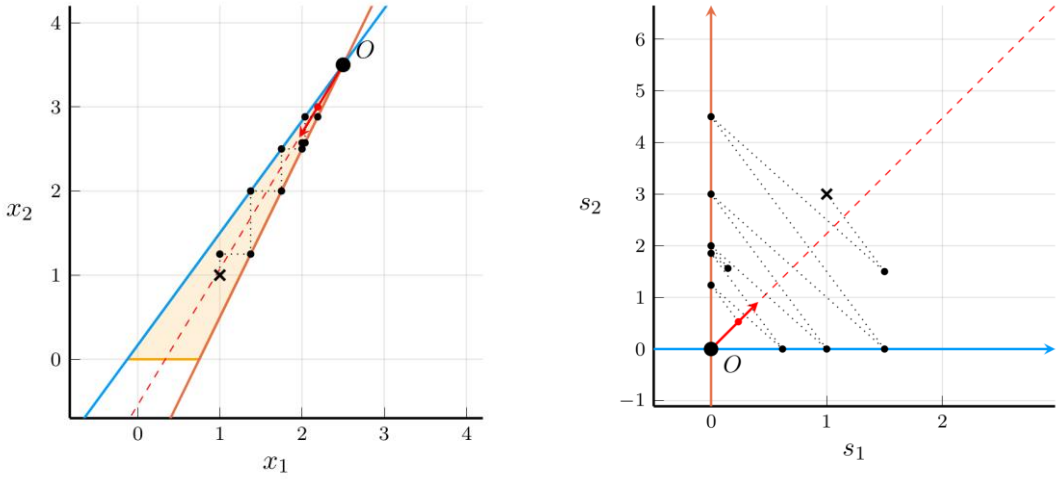


Рис. 9. Перемещение точки в процессе поиска решения для узкой ОДР

Fig. 9. Point movement during the search of a solution in the narrow feasible region

Исследование алгоритма при добавлении третьего ограничения

Введем третье ограничение, которое изменяло бы форму ОДР так, чтобы в ходе выполнения алгоритма точка, находящаяся внутри базиса между двумя ограничениями (s_1 и s_2), смогла выйти за пределы ОДР. В таком случае проверка на выход из ОДР и возврат точки обратно невозможны, и это приведет к тому, что алгоритм может не сойтись, хотя в текущем случае он находит решение (рис. 10).

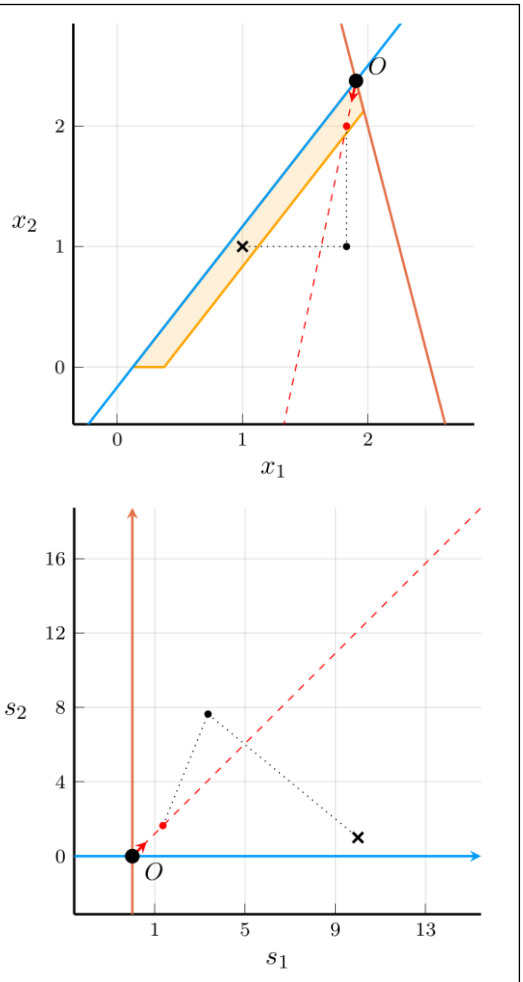


Рис. 10. Выход точки P за пределы ОДР при положительных координатах в базисе s_1s_2

Fig. 10. Point P leaves the feasible region while all of its coordinates in basis s1s2 are positive

Наблюдение 2. Определение последовательности округления координат

Используя вектор, задающий направление луча, можно определить последовательность осей координат, по которым стоит производить округление на следующих шагах. Дополняя наблюдение 1, определим веса $\overline{w}$ каждой из осей как соответствующие координаты нормализованного вектора $\hat{r}$, определяющего луч в исходном базисе:

$$\hat{r} = \frac{L^{-1}(\hat{v})}{\left|L^{-1}(\hat{v})\right|},$$

$$\overline{w} = \begin{pmatrix} |\hat{r}_1| \\ |\hat{r}_2| \end{pmatrix} \approx \begin{pmatrix} |-0.198| \\ |-0.980| \end{pmatrix} \approx \begin{pmatrix} 0.198 \\ 0.980 \end{pmatrix}.$$

Учитывая вектор $\overline{w}$, сначала нужно округлять по второй координате, а затем по первой, так как значение $|\hat{r}_2|$ больше $|\hat{r}_1|$. Аналогично можно определить направление округления, если использовать значения координат вектора $\hat{r}$ не по модулю, а по знаку:

$$\overline{d} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \end{pmatrix} \approx \begin{pmatrix} -0.198 \\ -0.980 \end{pmatrix} \begin{matrix} \downarrow \\ \downarrow \end{matrix}.$$

Таким образом, вектор $\overline{d}$ указывает, что по оси x_1 (первый элемент вектора) округлять нужно вниз (то есть влево на графике), как и по оси x_2 (второй элемент вектора) (то есть вниз на графике). Например, если бы вектор $\hat{r}$ содержал значения с разными знаками, округлять стоило бы в направлении знака по соответствующим осям координат, поскольку именно в ту сторону направлен луч:

$$\overline{d} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} \approx \begin{pmatrix} 0.37 \\ -0.65 \\ 0.76 \end{pmatrix} \begin{matrix} \uparrow \\ \downarrow \\ \uparrow \end{matrix}.$$

В таком случае округлять по осям x_1 и x_3 необходимо было бы вверх, а по оси x_2 – вниз.

Исследование алгоритма при изменении направления луча

Изменим ОДР, повернув ограничение s_2 так, чтобы луч смотрел не влево, а вправо. Тем самым продемонстрируем несостоятельность текущего способа построения луча, поскольку в таком случае алгоритм никогда не сойдется и целочисленное решение, лежащее внутри ОДР, не будет найдено (рис. 11).

Заключение

В статье был рассмотрен алгоритм поиска целочисленного решения, основанный на построении луча, смотрящего вглубь ОДР, и итеративном процессе округления координат

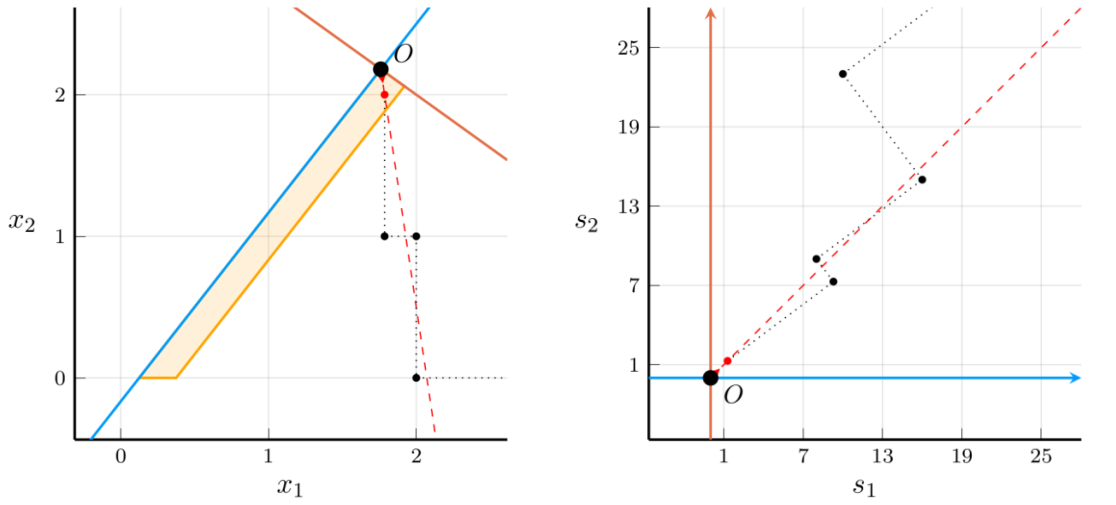


Рис. 11. Отдаление точки P от ОДР и расхождение алгоритма

Fig. 11. Point P moves away from the feasible region and the algorithm diverges

точки в направлении данного луча. Проведенные эксперименты показали работоспособность данного подхода, однако не всегда возможно построить луч, который смотрел бы вглубь ОДР. Кроме того, предложенный алгоритм может работать некорректно при введении дополнительных ограничений. Таким образом, в настоящий момент алгоритм подходит только для задач с двумя ограничениями.

Дальнейшие исследования могут быть направлены на модификацию алгоритма, чтобы он мог работать при большем числе ограничений. Также необходимо изменить способ построения луча. Например, его можно строить не среди ограничений, а между точками оптимального решения и начального базиса, что должно исключить возможность выхода точки за пределы ОДР.

Список литературы

1. Баусова З.И., Гамазина А Н., Дугина Ю.Н., Старикова А.Ю. Решение задач целочисленной линейной оптимизации // Вестн. ПензГУ. 2017. № 4. С. 117–121.
2. Арженовский С.В., Синявская Т.Г., Рудяга А.А. Решение задачи целочисленной оптимизации для фирмы // Учет и статистика. 2017. № 4. С. 36–43.
3. Легков К.Е., Нестеренко О.Е. Алгоритм формирования информационной структуры параллельных программ иерархической вычислительной системы // Наукоемкие технологии в космических исследованиях Земли. 2017. Т. 9. № 1. С. 52–59.
4. Гальмукова И.А. Линейное программирование в задачах экономики // Социально-экономические проблемы развития предпринимательства: региональный аспект: материалы VI Междунар. конф. 2017. С. 333–338.
5. Абдуахадов А.А. Краткая история развития математического программирования // Проблемы науки. 2021. № 6. С. 6–8.
6. Ямашкин Ю.В., Новокрещенова О.А. Системный подход к организации. Саранск, 2016. 195 с.
7. Матвеев Ю.Н., Иванов А.В. Использование метода покоординатного спуска для поиска решения задачи целочисленного программирования // Вестн. ТвГТУ. Сер. Технич. науки. 2023. Т. 17. № 1. С. 53–62.
8. Пинягина О.В. Метод покоординатного спуска для задачи рыночного равновесия с ценовыми группами // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. 2018. Т. 160. № 4. С. 718–730.
9. Мельников Б.Ф., Мельникова Е.А. О классической версии метода ветвей и границ // Компьютерные инструменты в образовании. 2021. № 1. С. 21–44. doi: 10.32603/2071-2340-2021-1-21-45.
10. Смагин Б.И., Машин В.В. Критический анализ решения задачи целочисленного линейного программирования методом Гомори // Наука и образование. 2022. Т. 5. № 1. URL: http://opusmgau.ru/index.php/see/article/view/4520/4552 (дата обращения: 13.04.2023).
11. Якубова У.Ш., Парпиева Н.Т., Мирходжаева Н.Ш. Некоторые применения графического и симплексного методов решения задач линейного программирования // Бюллетень науки и практики. 2022. Т. 8. № 4. С. 490–498. doi: 10.33619/2414-2948/77/57.
12. Адельшин А.В., Кучин А.К. Исследование L-структуры многогранника смешанной задачи максимальной выполнимости // ПДМ. 2017. № 38. С. 110–118. doi: 10.17223/20710410/38/9.

13. Атутова Н.Д. Применение эвристических методов для поиска булевых функций с криптографическими характеристиками // Прикладная дискретная математика. Приложение. 2022. № 15. С. 18–21. doi: 10.17223/2226308X/15/5.

14. Максимова Н.Н., Колтунов Н.С. Поиск оптимального кольцевого маршрута на карте г. Благовещенска с использованием алгоритма имитации отжига // Вестн. АмГУ. 2020. № 91. С. 3–8. doi: 10.22250/jasu.1.

15. Хамхоева Ф.Я. Нейронные сети в экономическом анализе: плюсы и минусы // Norwegian J. of Development of the Int. Sci. 2020. Т. 4. № 51. С. 72–75.

16. Толок А.В., Толок Н.Б. Решение задач математического программирования функционально-воксельным методом // Проблемы управления. 2017. № 3. С. 37–42.

Software & Systems doi: 10.15827/0236-235X.142.539-550 2023, vol. 36, no. 4, pp. 539–550

An integer programming solution: Iterative rounding of coordinates

Aleksey V. Ivanov
Yury N. Matveev

For citation
Ivanov, A.V., Matveev, Yu.N. (2023) 'An integer programming solution: Iterative rounding of coordinates', *Software & Systems*, 36(4), pp. 539–550 (in Russ.). doi: 10.15827/0236-235X.142.539-550
Article info
Received: 24.04.2023 After revision: 02.05.2023 Accepted: 17.05.2023

Abstract. The article proposes an algorithm for finding an integer solution using the idea of rounding coordinates of an optimal non-integer solution point and constructing a half-line directed deep into an acceptable solution area. The proposed algorithm is based on an iterative process of rounding coordinates of a point in the direction of a constructed half-line. The study educed that moving towards a half-line without going through all possible options simplifies the algorithm and avoids branching, which distinguishes this approach compared to other currently existing open methods, such as the method of clipping and the method of branches and boundaries. The aim of the work is to develop and study an algorithm for finding an optimal integer solution using the idea of rounding coordinates of a point of an optimal non-integer solution. During the study, the authors carried out description and experimental verification of this algorithm and the possibility of its application in different forms of an acceptable solution domain. The theoretical significance of the work is a development of a new algorithm that does not require performing a simplex-method at each stage, and uses a half-line instead of a plane at each algorithm step, which prevents an increase in problem spatial complexity compared to other methods. The study showed that the proposed algorithm has limitations; however, the main idea proved its operability, which is planned to be developed further.

Keywords: mathematical programming, linear programming, integer programming, optimization, algorithm, coordinate descent

References

1. Bausova, Z.I., Gamazina, A.N., Dugina, Yu.N., Starikova, A.Yu. (2017) 'Solving integer linear optimization problems', *Vestn. of Penza State University*, (4), pp. 117–121 (in Russ.).

2. Arzhenovskiy, S.V., Sinyavskaya, T.G., Rudyaga, A.A. (2017) 'Solving an integer optimization problem for a company', *Accounting and Statistics*, (4), pp. 36–43 (in Russ.).

3. Legkov, K.E., Nesterenko, O.E. (2017) 'The algorithm for forming the information structure of parallel programs of a hierarchical computing system', *High Tech in Earth Space Research*, 9 (1), pp. 52–59 (in Russ.).

4. Galmukova, I.A. (2017) 'Linear programming in economics', Proc. 6th Int. Conf. *Social and Economic Problems of Entrepreneurship Development: Regional Aspect*, pp. 333–338 (in Russ.).

5. Abduakhadov, A.A. 'Brief history of mathematical programming development', *Problems of Sci*., 2021, (6), pp. 6–8 (in Russ.).

6. Yamashkin, Yu.V., Novokreshchenova, O.A. (2016) *A Systematic Approach to the Organization*. Saransk, 195 p. (in Russ.).

7. Matveev, Yu.N., Ivanov, A.V. (2023) 'Usage of coordinate descent algorithm in search of a solution to an integer programming problem', *Vestn. of TvSTU. Ser. Tech. Sci.*, 17(1), pp. 53–62 (in Russ.).

8. Pinyagina, O.V. (2018) 'A coordinate descent method for market equilibrium problems with price groups', *Uchenye Zapiski Kazanskogo Universiteta. Ser. Fiziko-Matemat. Nauki*, 160 (4), pp. 718–730 (in Russ.).

9. Melnikov, B.F., Melnikova, E.A. (2021) 'On the classical version of the branch and bound method', *Computer Tools in Education*, (1), pp. 21–44 (in Russ.). doi: 10.32603/2071-2340-2021-1-21-45.

10. Smagin, B.I., Mashin, V.V. (2022) 'Critical analysis of the solution of the problem of integer linear programming by the Gomori method', *Sci. and Education*, 5 (1), available at: http://opusmgau.ru/index.php/see/article/view/4520/4552 (accessed April 13, 2023) (in Russ.).

11. Yakubova, U.Sh., Parpieva, N.T., Mirkhodzhaeva, N.Sh. (2022) 'Some applications of graphical and simplex methods for solving linear programming problems', *Bull. of Sci. and Pract*., 8 (4), pp. 490–498 (in Russ.). doi: 10.33619/2414-2948/77/57.

12. Adelshin, A.V., Kuchin, A.K. (2017) 'Analysis of L-structure of polyhedron in the partial max sat problem', *Applied Discrete Math.*, (38), pp. 110–118 (in Russ.). doi: 10.17223/20710410/38/9.

13. Atutova, N.D. (2022) 'Application of heuristic methods to search for Boolean functions with good cryptographic characteristics', *Applied Discrete Math. Supplement*, (15), pp. 18–21 (in Russ.). doi: 10.17223/2226308X/15/5.

14. Maksimova, N.N., Koltunov, N.S. (2020) 'Search for an optimal ring route on the Blagoveshchensk map using the annealing simulation algorithm', *Messenger AmSU*, (91), pp. 3–8 (in Russ.). doi: 10.22250/jasu.1.

15. Khamkhoeva, F.Ya. (2020) 'Neural networks in economic analysis: Pros and cons', *Norwegian J. of Development of the Int. Sci*., (51)4, pp. 72–75 (in Russ.).

16. Tolok, A.V., Tolok, N.B. (2017) 'Mathematical programming problems solving by functional voxel method', *Problemy Upravleniya*, (3), pp. 37–42 (in Russ.).

Авторы
Иванов Алексей Викторович [1], аспирант, lexuzieel@gmail.com
Матвеев Юрий Николаевич [1], д.т.н., профессор, matveev4700@mail.ru

[1] Тверской государственный технический университет, г. Тверь, 170026, Россия

Authors
Aleksey V. Ivanov [1], Postgraduate Student, lexuzieel@gmail.com
Yury N. Matveev [1], Dr.Sc. (Engineering), Professor, matveev4700@mail.ru

[1] Tver State Technical University, Tver, 170026, Russian Federation

УДК 004.9 doi: 10.15827/0236-235X.142.551-560 2023. Т. 36. № 4. С. 551–560

Автоматическая локализация ошибок времени выполнения в программном обеспечении с помощью нейронных сетей

А.М. Достовалова
А.А. Шерминская

Ссылка для цитирования
Достовалова А.М., Шерминская А.А. Автоматическая локализация ошибок времени выполнения в программном обеспечении с помощью нейронных сетей // Программные продукты и системы. 2023. Т. 36. № 4. С. 551–560. doi: 10.15827/0236-235X.142.551-560
Информация о статье
Поступила в редакцию: 02.05.2023 После доработки: 22.07.2023 Принята к публикации: 13.09.2023

Аннотация. Разработан метод автоматической локализации ошибок времени выполнения с помощью нейронной сети по данным трассировки осуществления функций программы. Метод сопоставляет каждой функции вероятность содержания ошибки, которая считается пропорциональной степени влияния значений параметров функции на результат выполнения программы. Влияние параметров определяется численной характеристикой (весом), вычисляемой по алгоритму Хашема. Метод применялся для отладки нескольких программ, различных по типам и причинам возникновения ошибок времени выполнения. Ошибки были расположены во вложенных функциях и проявлялись при определенных значениях входных данных. В каждой программе проведено сопоставление наиболее вероятных мест возникновения ошибок, которые определил метод, с их реальными местоположениями. Особенностями разработанного метода являются возможность работы с вложенными функциями, локализация множественных ошибок, а также ошибок, у которых место возникновения и место проявления в программе не совпадают. Во всех случаях параметры, содержащие ошибку, имели больший вес в сравнении с остальными, даже если ошибок в программе было несколько. При этом метод выделяет в программе полный путь ошибки, включающий в себя все параметры, связанные с ее возникновением. Благодаря этому с помощью предложенного метода можно определять положение логических ошибок в программах. Метод может применяться для отладки как программного, так и аппаратного обеспечения технических систем, поскольку логика его работы не зависит от источника исходных данных.

Ключевые слова: автоматическая локализация ошибок, ошибки времени выполнения, отладка ПО, нейронные сети, трассировка

Требования к функциональности современного программного и аппаратного обеспечения сложных технических систем с каждым годом ужесточаются. Это приводит к увеличению числа входящих в них устройств и программ. Усложнение системы ведет к увеличению возможного количества появляющихся в ней неисправностей, комплексная отладка такой системы становится более трудоемкой. Возникает необходимость автоматизации процесса поиска неисправностей, особенно ошибок времени выполнения, определение положения которых наиболее сложно.

Для ПО существуют методы автоматической локализации ошибок времени выполнения [1]. Эти методы можно разделить на две группы: параметрические и непараметрические. Первые (например, методы срезки и их модификации [1]) для локализации ошибок используют значения программных переменных. Непараметрические методы, к которым относятся спектральные, статистические и некоторые методы машинного обучения, определяют положение ошибки в коде по данным трассировки (трассам) программы – численным данным о результатах работы отслеживаемых участков кода (функций, инструкций), полученным во время выполнения программы, и по результатам выполнения программы (выполнилась она до конца или завершилась с ошибкой) [2]. При этом в непараметрических методах используются только данные об успешном или неудачном завершении выполнения участков программы.

Непараметрические методы обычно более производительные по сравнению с параметрическими, поскольку программист может сам назначить подозрительные на наличие ошибки участки кода, в том числе проводя поиск ошибки в несколько этапов, каждый раз измельчая исследуемые блоки. Но эти методы пригодны для ограниченного круга задач – для программ, в которых нет участков вложенных функций и местоположение ошибки времени выполнения совпадает с местом ее проявления (характерно для спектральных и статистических методов). Непараметрическими методами невозможно локализовать ошибки, проявляющиеся лишь при определенных значениях входных параметров, особенно если программа завершает свое выполнение в момент ее возникновения. Кроме того, при трассировке алгоритм судит

о наличии ошибки в функции по тому, соответствуют ли ее выходные параметры заданному диапазону. Такая информация сильно подвержена искажениям, поскольку при определении границ диапазона уже может быть допущена ошибка.

Основным затруднением в процессе поиска неисправностей как в ПО, так и в технической системе в целом, является сопоставление большого числа разнородных параметров, неявно связанных между собой. Для выявления подобных закономерностей перспективным представляется использование нейронных сетей. Преимуществом этого подхода является универсальность: нейронная сеть способна искать закономерности в данных независимо от их природы, то есть в общем случае в состав исследуемой системы могут входить как программные, так и аппаратные средства.

Была поставлена задача разработать метод автоматической локализации ошибок времени выполнения с помощью нейросети в программах по данным трассировки выполнения функций. Каждой функции метод должен сопоставлять вероятность содержания ошибки, которая считается пропорциональной степени влияния значений параметров функции на результат работы программы. Численная характеристика влияния параметра вычисляется с помощью алгоритма Хашема [3]. Новизна разработанного метода заключается в возможности работы с вложенными функциями и в локализации множественных ошибок, а также ошибок, у которых место возникновения и место проявления в программе не совпадают. Предложенный метод тестировался на наборе программ с заложенными в коде ошибками времени выполнения, находящимися внутри последовательности вложенных функций. Предполагаемые места возникновения ошибок сравнивались с их реальным местоположением.

Подходы к автоматизации процесса поиска неисправностей в ПО

Задача автоматизации поиска неисправностей в ПО имеет долгую историю. Одной из первых работ по данной тематике является [4], в которой описан метод статической срезки. В нем положение некорректных инструкций в программе определяется среди набора выделенных срезов – строк кода, содержащих переменные, связанные с появлением ошибки. В последние годы методы срезки использовались в гибридных подходах в связке с альтернативными методами автоматической локализации ошибок в программном коде [5, 6].

Кроме методов срезки, существует множество других подходов к локализации ошибок [1, 7]. Среди них можно выделить два активно развивающихся направления – спектральные методы и методы машинного обучения.

В спектральных методах программа разбивается на участки, подозрительные на наличие в них ошибок. При отладке для каждого участка отслеживаются правильность его работы и степень покрытия тестами и вычисляется его подозрительность – численное значение, характеризующее вероятность наличия ошибки [1]. Формула для вычисления подозрительности с течением времени претерпела изменения [1], в нее были добавлены вероятностные элементы [8, 9]. Созданы модификации спектральных методов, повышающие для больших программ точность определения участков с ошибкой [10, 11].

Методы машинного обучения для автоматической локализации ошибок разделяются на методы, направленные только на определение положения ошибки, и методы, и локализующие, и исправляющие найденную ошибку. Методы первой группы близки по идее к спектральным. С помощью полносвязной нейронной сети определяются взаимосвязи между покрытием тестами отслеживаемых участков кода и соответствующими им значениями подозрительности [12].

Методы второй группы рассматривают задачу поиска и исправления ошибки как задачу машинного перевода [13]. Такая постановка позволяет использовать архитектуры LLM, например, инструмент ChatGPT [14]. В работе [15] также был продемонстрирован потенциал языковых моделей к исправлению ошибок времени выполнения при дополнении набора для обучения сети данными трассировки программы.

Перечисленные подходы различны, но обладают общей чертой – каждый метод накладывает ограничения на обрабатываемые данные. Методы срезки требуют, чтобы в исследуемом участке кода присутствовала только одна ошибка. Спектральные методы не используют данные о значениях программных переменных, из-за чего их способность к локализации ошибок, проявляющихся только при определенных значениях входных данных, ограничена. Кроме того, оценка правильности прохождения участков трасс требует создания критериев оценива-

ния, при разработке которых также могут возникнуть ошибки. Соответственно, исходные данные для спектрального метода могут быть недостоверными.

Использование большинства языковых моделей для исправления ошибок времени выполнения ограничено случаями, когда ошибки не являются логическими и не проявляются только при определенных значениях входных данных. Добавление к обучающему набору данных трассировки программы расширяет возможности языковых моделей. Однако в работе [15] задача разметки подозрительных участков трасс перед обучением решена не была – разметка проводилась вручную.

Кроме того, при отладке программно-аппаратных комплексов часто возникает ситуация, когда исходного кода у программиста нет. Часть функционала может быть доступна только в формате библиотек с известным интерфейсом, но неизвестной реализацией. В таких библиотеках могут существовать ошибки, проявляющиеся только при их функционировании как части системы, но не отдельного блока.

Предложенный в данной работе метод не сталкивается с вышеперечисленными ограничениями. Для локализации ошибок он использует только данные трассировки программы, к которым легко получить доступ при отладке программно-аппаратных комплексов и программ, даже если исходный код последних не известен. Метод использует непосредственно значения переменных, отслеживаемых при отладке программы, поскольку эти данные не искажены дополнительными проверками правильности. Кроме того, применение метода не ограничено случаями, когда в программе присутствует только одна ошибка.

Общее описание метода локализации ошибок времени выполнения

Представляемый метод локализует ошибки времени выполнения в программе, состоящей из набора функций $f_1(\bar{\theta}_1)...f_n(\bar{\theta}_n)$, $n \geq 1$, где $\bar{\theta}_1...\bar{\theta}_n$ – векторы выходных параметров функций. При трассировке для этих функций отслеживаются значения входных и выходных параметров. Считается, что логика работы программы (блок-схема) не изменяется в зависимости от входных данных: число вызовов данных функций и их последовательность остаются неизменными.

Если в программе среди f_i нет вложенных функций, ее можно однозначно представить в виде одной функции $F(\bar{\theta})$, где $\bar{\theta} = (\bar{\theta}_1...\bar{\theta}_n)$ – вектор всех выходных параметров функций программы. Функция $F(\bar{\theta})$ принимает два значения: $\omega_1 = 1$ в случае успешного выполнения программы, когда все входящие в нее функции отработали без ошибок, и $\omega_2 = 0$ в противном случае. В пространстве параметров, где $F(\bar{\theta})$ является разделяющей поверхностью, возможно определить направления, по которым эта функция изменяется наиболее сильно – больше всего значения градиента $F(\bar{\theta})$. Эти направления позволяют определить набор параметров из вектора $(\bar{\theta})$, изменение которых сильнее всего влияет на результат выполнения программы – значение ω_i. Определив эти параметры и предположив, что вероятность содержания ошибки пропорциональна степени влияния значений параметров функций на результат выполнения программы, можно выделить функции программы, которые содержат ошибку.

Однако связи между значениями $\bar{\theta}$ и ω_i, а также вид зависимости (линейная, квадратичная и пр.) заранее не известны. Функция $F(\bar{\theta})$ задана набором точек ($\bar{\theta}$, ω_i), $i = 1,2$, каждая из которых описывает результат выполнения программы ω_i для заданных параметров функций $\bar{\theta} = (\bar{\theta}_1...\bar{\theta}_n)$. Чтобы вычислить градиент функции $F(\bar{\theta})$, вначале необходимо построить ее аппроксимацию. Для этого в представляемом методе используется полносвязная нейронная сеть. Выбор такого способа аппроксимации $F(\bar{\theta})$ обусловлен тем, что сеть является универсальным аппроксиматором. Согласно теореме Хехт-Нильсена [16], с помощью двухслойной нейронной сети можно аппроксимировать любую непрерывную функцию. Сеть обучается на наборе данных трассировки программы. Она учится определять для вектора, составленного из значений параметров $(\bar{\theta})$, метку ω_i – соответствующий этим значениям результат работы программы.

После того как сеть обучена, необходимо определить для каждого входного параметра нейросети степень его влияния на выходное значение метки ω_i – результат выполнения трассы. Для этого каждому параметру ставится в соответствие численная характеристика – вес параметра, вычисляемый по алгоритму Хаше-

ма [3] усреднением по обучающему набору значений градиента $F(\bar{\theta})$ в точках данных.

Обработка участков вложенных функций

Описание метода было сделано только для работы с программами, которые не содержат вложенных функций. Если в программе встречаются вложенные участки (что характерно для многих реальных задач), метод не сможет локализовать в ней ошибку из-за неспособности сформировать полный вектор параметров $(\bar{\theta})$. Если программа завершается с ошибкой на нижнем уровне вложенности, то функции верхнего уровня не отрабатывают до конца и значения их выходных параметров оказываются недоступными. Ошибку в таком случае можно локализовать только в функции нижнего уровня вложенности. Если фактически ошибка расположена в одной из функций верхнего уровня, определить ее местоположение становится невозможным: нет ни одного параметра, связанного с нужной функцией.

Чтобы преодолеть ограничение на наличие вложенности в обрабатываемых программах, представляемый метод дополняют еще одним шагом, предшествующим шагу составления вектора параметров $\bar{\theta}$. Данные трассировки программы обрабатываются алгоритмом разворачивания вложенности. Назначение этого алгоритма в том, чтобы заменить вызовы вложенных функций на последовательность вызовов обычных функций. Например, рассмотрим последовательность вложенных функций

$$f_1 \supset (f_2 \supset (f_3 \rightarrow f_4)), \quad (1)$$

где $\supset$ – вызов из левой функции вложенного правого участка; $\rightarrow$ – последовательные вызовы функций.

Функции f_3, f_4 принадлежат нижнему уровню вложенности и выполняются на нем последовательно друг за другом. Развернутая трасса этой последовательности имеет вид

$$f_{1_}f_2 \rightarrow f_{2_}f_3 \rightarrow f_3 \rightarrow f_4 \rightarrow f_{4_}f_2 \rightarrow f_{2_}f_1, \quad (2)$$

где функции $f_{1_}f_2$ и $f_{2_}f_3$ – части функций f_1, f_2 до вхождения во вложенный участок; $f_{4_}f_2$ и $f_{2_}f_1$ – части f_1, f_2 после выхода из вложенности (участки между несколькими последовательными уровнями вложенности внутри f_i обозначаются $f_{i_}f_{j_}f_k$). Различия между схемами вызовов исходной и развернутой трасс представлены на рисунке 1.

В развернутой трассе выходные параметры функций f_3, f_4, $f_{4_}f_2$, $f_{2_}f_1$ – это выходные параметры функций f_3, f_4, f_2, f_1 из исходной трассы. Для функций $f_{1_}f_2$, $f_{2_}f_3$ в исходной трассе аналогов нет. Для них выходными параметрами считаются входные параметры следующих после них функций из вложенных участков (соответственно, функций) $f_{2_}f_3$. Аналогично предлагается делать для функций, в которых программа завершается с ошибкой. Их выходные параметры неинформативны, поскольку содержат значение, приводящее к ошибке. Для развернутой трассы уже можно составить полный вектор параметров $(\bar{\theta})$ и определить среди них обладающие наибольшим влиянием на результат выполнения программы.

Определение параметров с наибольшим влиянием на результат выполнения программы

Для установления зависимости между значениями параметров $\bar{\theta} = (\bar{\theta}_1 ... \bar{\theta}_n)$ и результатом выполнения программы ω_i с помощью нейронной сети строилась аппроксимация функции $F(\bar{\theta})$. Влияние параметров на результат неравнозначно. Оценить степень этого влияния по построенной аппроксимации можно с помощью алгоритма Хашема [15]. В нем влияние

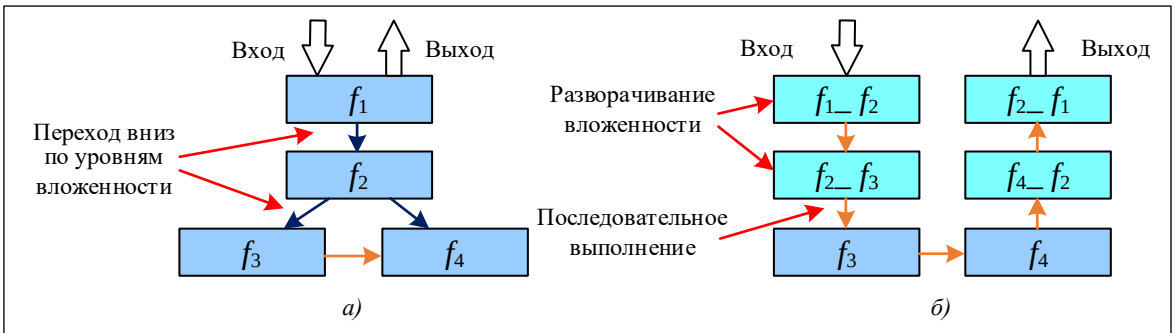


Рис. 1. Участок трассы следования программы: а) исходный; б) с разворачиванием вложенности

Fig. 1. Part of program trace: a) original view; б) view after unfolding nesting

входных параметров $\bar{\theta} = (\bar{\theta}_1 ... \bar{\theta}_n)$ на результат работы нейросети оценивается в ходе вычислений значений градиента $F(\bar{\theta})$ в точках данных. Обозначим O_k выходное значение k-го узла выходного N+1-го слоя, h_i^j – выходное значение i-го узла внутреннего слоя ($j = 1 \ldots N$) сети, I_i – значение на i-м входе нейросети, net_l^j – взвешенную сумму в l-м узле на j-м слое, w_{il}^j – весовой коэффициент, связывающий i-й узел с j-го слоя с l-м узлом на j+1-м, $j = 0 \ldots N$. Градиент в точке вычисляется с помощью алгоритма обратного распространения ошибки. Формулы в приведенных обозначениях имеют вид:

$$\frac{\partial O_k}{\partial h_i^N} = \frac{\partial O_k}{\partial net_k^{N+1}} \cdot \frac{\partial net_k^{N+1}}{\partial h_i^N} = O_k(1 - O_k) w_{ik}^N, \forall i,k, \quad (3)$$

$$\frac{\partial O_k}{\partial h_i^j} = \sum_l \frac{\partial O_k}{\partial h_l^{j+1}} \cdot \frac{\partial h_l^{j+1}}{\partial net_l^{j+1}} \cdot \frac{\partial net_l^{j+1}}{\partial h_i^j} = \\ = \sum_l \frac{\partial O_k}{\partial h_l^{j+1}} \cdot h_l^{j+1}(1 - h_l^{j+1}) \cdot w_{il}^j \quad \forall i,k, \quad (4)$$

$$\frac{\partial O_k}{\partial I_i} = \sum_l \frac{\partial O_k}{\partial h_l^1} \cdot \frac{\partial h_l^1}{\partial net_l^1} \cdot \frac{\partial net_l^1}{\partial I_i} = \\ = \sum_l \frac{\partial O_k}{\partial h_l^1} \cdot h_l^1(1 - h_l^1) \cdot w_{il}^0 \quad \forall i,k. \quad (5)$$

Для оценки приоритета входных параметров нейросети полученные значения градиента в точках данных усредняются по всему обучающему набору. Для каждого входа I_i нейронной сети вычисляется вес зависимости D_{ik} между ним и каждым выходным узлом O_k (N_{data} – число элементов в обучающем наборе) по формуле

$$D_{ik} = \frac{\sum_{s=1}^{N_{data}} \left| \frac{\partial O_k^s}{\partial I_i^s} \right|}{N_{data}}. \quad (6)$$

Эти значения нормируются и объединяются в матрицу весов DM (N_I – число узлов входного слоя, N_o – число узлов выходного слоя), для вычисления которой используется формула

$$DM = \begin{pmatrix} \frac{D_{11}}{\max_j(D_{j1})} & \ldots & \frac{D_{1k}}{\max_j(D_{jk})} \\ \ldots & \ldots & \ldots \\ \frac{D_{i1}}{\max_j(D_{j1})} & \ldots & \frac{D_{ik}}{\max_j(D_{jk})} \end{pmatrix} \forall j,k, \quad (7)$$

$j = 1 \ldots N_I, k = 1 \ldots N_O$.

Достоинство алгоритма Хашема в сравнении с альтернативными методами определения приоритета входных данных [3] в том, что он может работать с глубокими нейросетями, которые обычно приближают одну и ту же функцию более эффективно, чем неглубокие, даже если общее число нейронов у них совпадает [17]. Вычисленные с помощью алгоритма Хашема значения весов используют в предложенном методе локализации ошибок времени выполнения, чтобы ранжировать функции, составляющие программу, по вероятности содержания ошибки. Она тем выше, чем больше вес соответствующих функций параметров.

Экспериментальная отработка метода

Были исследованы пять тестовых программ. При трассировке отслеживались значения входных и выходных параметров функций. Завершение программы с ошибкой происходило, если значение хотя бы одного из параметров выполняемой функции выходило за пределы установленного для него диапазона.

Каждая исследуемая программа представляет собой последовательность вызовов вложенных функций, организованных в одну из двух структур (рис. 2), содержащих три уровня вложенности. На нижнем уровне функции выполняются последовательно. В каждой программе заложено условие, при выполнении которого она завершается с ошибкой в *func*5 или *func3*. Первопричина ошибки во всех случаях существует в предшествующих функциях: места возникновения и проявления ошибки не совпадают. Пять тестовых программ отличаются друг от друга характером ошибок, их количеством и причиной возникновения.

В **Программе 1** на вход *mod*1 подавался вектор длиной 10, состоящий из случайным образом сгенерированных чисел заданного диапазона. В этой функции создавался второй вектор, полученный из элементов первого умножением их значений на 2. В *mod*3 значения двух векторов еще раз умножались на 2, после чего генерировались два числа – *val*1 и *val*2. Первое принимало значения от 0 до 10, второе – 0 или 1. Оба вектора и два числа передавались в *mod*5. Далее в *func*4 вычислялся максимум из элементов векторов. В *func*5 происходила проверка значений *val*1 и *val*2: если второе число равнялось 1, а первое принимало любое значение, кроме 5 или 9, считалось, что программа завершилась с ошибкой. Программа 1 моделировала случай распределенной ошибки, проявляющейся при определенной комбинации значений параметров функций.

Программа 2 повторяла Программу 1, за исключением того, что в векторе входных па-

раметров не было значений *val*1 и *val*2. Программа 2 моделировала случай, когда ошибка не зависела от параметров, составляющих вектор параметров $(\bar{\theta})$.

Программа 3 тоже повторяла Программу 1, но в векторе параметров *val*1 и *val*2 повторялись три раза (названия переменных при этом были разными). Программа 3 моделировала случай, когда в векторе содержащие ошибку параметры повторялись несколько раз.

В **Программе 4** на вход *mod*2 подавался вектор длины 10, состоящий из заданных чисел (от 10 до 20). В этой функции создавался второй вектор, полученный из элементов первого умножением их значений на 2. В *mod*4 для элементов векторов вычислялись сумма значений *summVals* и максимум *maxVals*. Полученные значения передавались на вход *func*1, в которой вычислялся *res*1 = *summVals* + *maxVals* *c*1 и *res*2 = *res*1*c*2, *c*2 = 0; 1, *res*1 и *res*2 подавались на вход *func*2, в которой эти числа умножались на 1. В *func*3 программа завершалась ошибкой, если параметр *res*2 функции *func*2, передающийся ей на вход как параметр *inp*1, был отрицательным: ошибка была заложена в функции *func*1.

Программа 5 повторяла Программу 4 до момента выполнения функции *func*2. В Программе 5 *func*2 повторяла *func*1, за исключением условия, по которому *c*2 принимала отрицательное значение; *func*2 принимала на вход *maxVals* и *summVals* и возвращала значения *res*1 и *res*2; *func*3 принимала на вход значения и от *func*1, и от *func*2. Программа завершалась в этой функции с ошибкой при условии, что или *res*2 функции *func*1, или *res*2 функции *func*3 отрицательно.

Программа 5 моделировала случай, когда в ней присутствовало несколько ошибок. Программы 2 и 3 позволяли оценить влияние ошибок при составлении вектора параметров на результат обработки данных трассировки. В Программах 4 и 5, в отличие от Программ 1–3, ошибки проявлялись неявно: в вектор параметров входили только значения, на которые влияли параметры с ошибкой. Сами эти параметры в векторе $(\bar{\theta})$ не содержались.

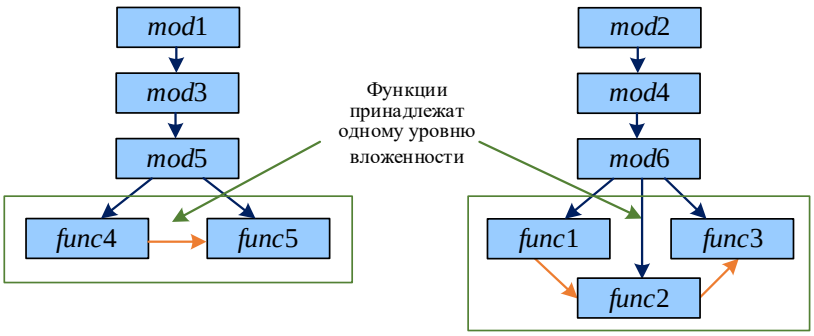


Рис. 2. Структуры исследуемых программ

Fig. 2. Structures of programs under study

Результаты локализации ошибок в исследуемых функциях

Для вычисления весов параметров функций программ 1–5 использовалась пятислойная полносвязная нейронная сеть с одним остаточным блоком [17]. Схема нейронной сети представлена на рисунке 3. Число нейронов в слоях сети изменялось в соответствии с длиной вектора данных *data_size*. В таблице для пяти тестовых программ приведены значения весов для параметров с ошибкой. Для параметров без ошибки были определены максимальное и среднее значения весов, а также их среднее квадратичное отклонение. Функции в программе ранжируются по вероятности содержания ошибки в соответствии с максимальными значениями весов их параметров. Соответствующий график для пяти тестовых программ представлен на рисунке 4.

На рисунке 4 каждой функции присвоен номер, определяющийся в соответствии с высотой столбца функции на графике. Столбцы наибольшей высоты соответствуют функциям, параметры которых обладают наибольшим весом.

В четырех из пяти случаев, для Программ 1, 3, 4 и 5, удалось выделить функции, содержащие ошибки. Параметры с ошибкой выделялись в общем множестве значений как пиковые значения: их веса были больше средних значений весов параметров без ошибки в 3–10 раз. У параметров без ошибки разброс значений весов относительно среднего, характеризующийся величиной среднеквадратичной ошибки, невелик.

Среди рассмотренных программ выделяется случай Программы 2, когда в векторе $(\bar{\theta})$ нет ни одного параметра, связанного с возник-

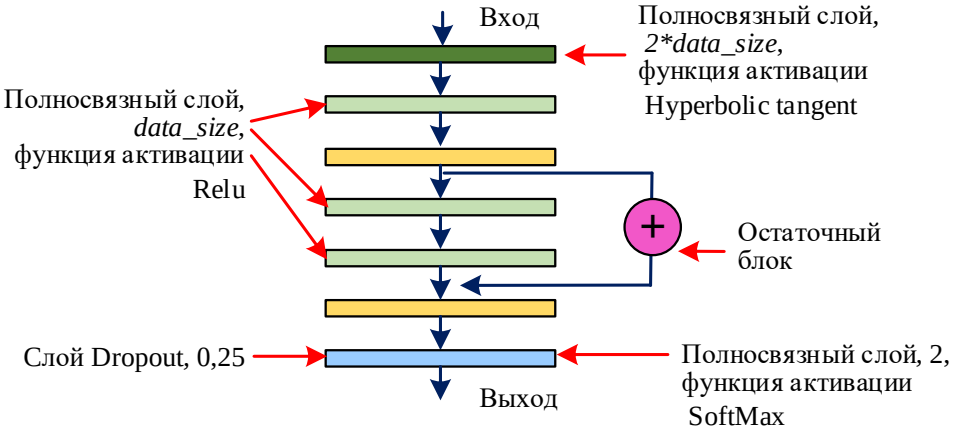


Рис. 3. Схема используемой нейронной сети

Fig. 3. Neural network scheme

новением ошибки. Функции этой программы возможно ранжировать по вероятности содержания ошибки. Но значения весов параметров всех функций близки друг к другу. Среди них нет ни одного яркого пикового значения, которое явно указывало бы на связь с появлением ошибки. Подобные случаи требуют дополнительного тестирования, в котором будут изменены отслеживаемые параметры или функции.

Интерес представляет случай Программы 4. В ней, кроме содержащей ошибку функции *func*1, выделяются еще функции *func*2 и *func*3, потому что у параметров *res*1 и вычисленных с его помощью *res*1 из *func*2 и у *inp*1 из *func*3 близкие значения весов. Метод может выделять лишние параметры, кроме тех, что действительно содержат ошибку. Однако все выделенные параметры оказываются связанными с возникновением ошибки: это параметры, значения которых являются условием появления ошибки, параметры, которым присваивается значение с ошибкой, и т.д. Для каждой ошибки можно проследить ее полный путь в программе – в данном случае от *func*1 до *func*3. В случае Программы 4 полный путь ошибки избыточен для определения ее местонахождения. Однако если допущенная ошибка является логической (например, условие, по которому она возникает, не должно выполняться вообще), исследование полного пути является единственным способом определения ее реального положения.

С помощью предложенного метода удалось локализовать ошибки в случае нескольких независимых ошибок в программном коде (Программа 5), а также когда значения, содержащие

Характеристики значений весов параметров функций тестовых программ

Weight values characteristics for test program function parameters

Номер программы	Параметры с ошибкой			Максимальный вес параметров без ошибки	Средний вес параметров без ошибки	Среднеквадратичная ошибка весов параметров без ошибки
	Функция	Имя	Вес			
1	*mod*5	*val*1	0.1883	0.026	0.0145	0.007
	*mod*5	*val*2	0.1408			
2	-	-	-	0.047	0.019	0.012
3	*mod*5	*val*1	0.0927	0.019	0.0084	0.005
	*mod*5	*val*1_0	0.0977			
	*mod*5	*val*1_1	0.0916			
	*mod*5	*val*2	0.0781			
	*mod*5	*val*2_0	0.0598			
	*mod*5	*val*2_1	0.0847			
4	*func*1	*res*1	0.1998	0.1853	0.015	0.009
5	*func*1	*res*1	0.08	0.12	0.02	0.089
	*func*2	*res*1	0.18			

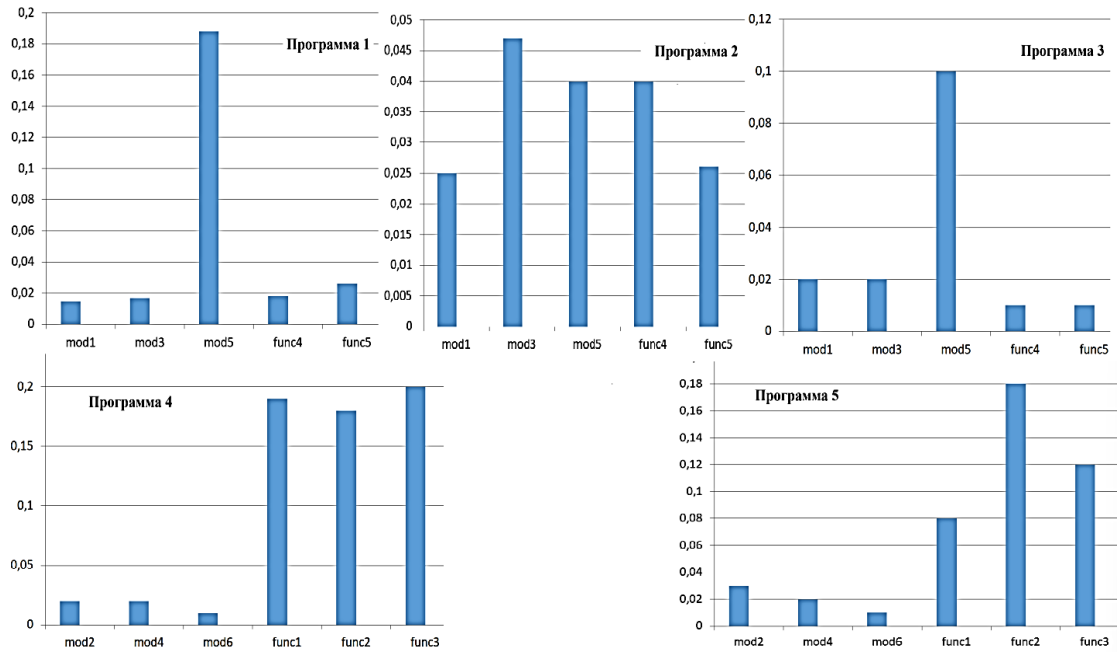


Рис. 4. Ранжирование функций в программе по весу (максимальной вероятности содержания ошибки) их параметров

Fig. 4. Ranking of program functions by parameter weights (maximum probability of containing errors)

ошибку, в векторе параметров повторялись (Программа 3). В обоих случаях значения весов параметров с ошибкой ярко выделялись среди остальных. При этом у повторяющихся параметров веса оказались близкими, так как значения градиентов по этим направлениям меняются схожим образом. Это является преимуществом предложенного метода, поскольку нет необходимости накладывать на исследуемые данные дополнительные ограничения на число независимых ошибок в трассе или то, чтобы каждый параметр встречался в векторе ровно один раз. При обработке реальных данных это позволит увеличить число случаев, в которых метод даст возможность определить местоположение ошибки.

Заключение

В результате исследования разработан метод автоматической локализации ошибки времени выполнения с использованием нейронных сетей в программах по данным трассировки выполнения функций. Предложенный метод производителен, как и непараметрические методы автоматической локализации ошибки времени выполнения, но при этом лишен их главного недостатка: с каждой функцией в трассе связаны неискаженные данные, по значениям которых производится локализация ошибки.

Метод применен для обработки данных трассировки пяти тестовых программ, различных по числу и природе возникающих в них ошибок времени выполнения. С его помощью удалось определить параметры, связанные с возникновением ошибки, когда они присутствовали в векторе параметров, подающемся на вход нейросети. Метод способен выделять ошибки в программах с вложенными функциями, в том числе и когда их несколько, а также определять положение ошибок, возникающих только при определенных значениях параметров функций. Он имеет потенциал в качестве средства определения положения логических ошибок в коде. Метод можно применить для отладки как аппаратного обеспечения, так и ПО сложных технических систем, поскольку для него не важна природа исследуемых данных.

Список литературы

1. Wong W.E., Gao R., Li Y., Abreu R., Wotawa F. A survey on software fault localization. IEEE Transactions on Software Eng., 2016, vol. 42, no. 8, pp. 707–740. doi: 10.1109/TSE.2016.2521368.

2. Kraft J., Wall A., Kienle H. Trace recording for embedded systems: Lessons learned from five industrial projects. Proc. RV. LNPSE, 2010, vol. 6418, pp. 315–329. doi: 10.1007/978-3-642-16612-9_24.

3. Комаров Я.Б., Шерминская А.А., Николаев А.А. Методы определения приоритета входных параметров функционально-математических библиотек с закрытым исходным кодом // Тр. КГНЦ. 2021. Спец. № 2. С. 91–96.

4. Weiser M. Program slicing. IEEE Trans. Soft. Eng., 1984, vol. SE-10, no. 4, pp. 352–357. doi: 10.1109/TSE.1984.5010248.

5. Reis S., Abreu R., d'Amorim M. Demystifying the combination of dynamic slicing and spectrum-based fault localization. Proc. IJCAI, 2019, pp. 4760–4766.

6. Soremekun E., Kirschner L., Böhme M., Zeller A. Locating faults with program slicing: An empirical analysis. Empirical Software Engineering, 2021, vol. 26, no. 3, art. 51. doi: 10.1007/s10664-020-09931-7.

7. Zakari A., Lee S.P., Alam K.A., Ahmad R. Software fault localisation: A systematic mapping study. IET Software, 2019, vol. 13, no. 1, pp. 60–74. doi: 10.1049/iet-sen.2018.5137.

8. Yang Y., Deng F., Yan Y., Gao F. A fault localization method based on conditional probability. Proc. IEEE 19th Int. Conf. on QRS-C, 2019, pp. 213–218. doi: 10.1109/QRS-C.2019.00050.

9. Landsberg D., Barr Earl T. Doric: Foundations for statistical fault localization. ArXiv, 2018, art. 1810.00798. URL: https://arxiv.org/abs/1810.00798 (дата обращения: 13.07.2023).

10. Keller F., Grunske L., Simon H., Filieri A., van Hoorn A., Lo D. A critical evaluation of spectrum-based fault localization techniques on a large-scale software system. Proc. IEEE Int. Conf. QRS, 2017, pp. 114–125. doi: 10.1109/QRS.2017.22.

11. Kong L., Wang J., Zhou S., Wang M. A multiple-fault localization method for embedded software with applications in engineering. Math. Problems in Engineering, 2021, vol. 2021, pp. 1–17, art. 7038979. doi: 10.1155/2021/7038979.

12. Heris S.R., Keyvanpour M. Effectiveness of weighted neural network on accuracy of software fault localization. Proc. ICWR, 2019, pp. 100–104. doi: 10.1109/ICWR.2019.8765262.

13. Chalambous Y., Tihanyi N., Jain R., Sun Y. et.al. A new era in software security: Towards self-healing software via large language models and formal verification. ArXiv, 2023, art. 2305.14752. URL: https://arxiv.org/abs/2305.14752 (дата обращения: 13.07.2023).

14. Haque M.A., Li S. The potential use of ChatGPT for debugging and bug fixing. EAI Endorsed Trans AI Robotics, 2023, vol. 2. doi: 10.4108/airo.v2i1.3276.

15. Bouzenia I., Ding Y., Pei K., Ray B., Pradel M. TraceFixer: Execution trace-guided program repair. ArXiv, 2023, art. 2304.12743. URL: https://arxiv.org/abs/2304.12743 (дата обращения: 13.07.2023).

16. Hecht-Nielsen R. Kolmogorov's mapping neural network existence theorem. Proc. IEEE Int. Conf. on Neural Networks, 1987, vol. 3, pp. 11–14.

17. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. СПб: Питер, 2018. 480 с.

Software & Systems doi: 10.15827/0236-235X.142.551-560 2023, vol. 36, no. 4, pp. 551–560

Automatic runtime error localization for software debugging using neural networks

Anastasiya M. Dostovalova
Anna A. Sherminskaya

For citation
Dostovalova, A.M., Sherminskaya, A.A. (2023) 'Automatic runtime error localization for software debugging using neural networks', *Software & Systems*, 36(4), pp. 551–560 (in Russ.). doi: 10.15827/0236-235X.142.551-560
Article info
Received: 02.05.2023 After revision: 22.07.2023 Accepted: 13.09.2023

Abstract. A new automatic runtime error localization method using a neural network was developed. The method determines bug position by program functions tracing data. The method matches an error containing probability for each function. This probability is proportional to the effect of each function parameter on a program execution result. A numerical characteristic calculated by Hashem algorithm determines a parameter effect. The method was applied for debugging several programs with nested functions. The most probable bug locations defined by the method were compared with their real positions. The developed method has some features. It is able to localize multiple bugs, nested function bugs, and bugs with unequal manifestation and location places. In all cases, bugged parameters had more weight in comparison with rest parameters. At the same time, method determines a full error trace for each bug. The trace consists of all program parameters related to a bug. Therefore, the method is able to localize logical errors in program. The method can be used for debugging both technical system software and hardware since method logic does not depend on source data.
Keywords: program debugging, neural network, program tracing, automatic bug localization, runtime error

References

1. Wong, W.E., Gao, R., Li, Y., Abreu, R., Wotawa, F. (2016) 'A survey on software fault localization', *IEEE Transactions on Software Eng*., 42(8), pp. 707–740. doi: 10.1109/TSE.2016.2521368.
2. Kraft, J., Wall, A., Kienle, H. (2010) 'Trace recording for embedded systems: Lessons learned from five industrial projects', *Proc. RV. LNPSE*, 6418, pp. 315–329. doi: 10.1007/978-3-642-16612-9_24.
3. Komarov, Ya.B., Sherminskay, A.A., Nikolaev, A.A. (2021) 'Methods for determining the priority of input parameters of functional-mathematical libraries with a closed source code', *Transactions of KSRC*, (spec. 2), pp. 91–96 (in Russ.).
4. Weiser, M. (1984) 'Program slicing', *IEEE Trans. Soft. Eng*., SE-10(4), pp. 352–357. doi: 10.1109/TSE.1984.5010248.
5. Reis, S., Abreu, R., d'Amorim, M. (2019) 'Demystifying the combination of dynamic slicing and spectrum-based fault localization', *Proc. IJCAI*, pp. 4760–4766.
6. Soremekun, E., Kirschner, L., Böhme, M., Zeller, A. (2021) 'Locating faults with program slicing: An empirical analysis', *Empirical Software Engineering*, 26(3), art. 51. doi: 10.1007/s10664-020-09931-7.
7. Zakari, A., Lee, S.P., Alam, K.A., Ahmad, R. (2019) 'Software fault localisation: A systematic mapping study', *IET Software*, 13(1), pp. 60–74. doi: 10.1049/iet-sen.2018.5137.
8. Yang, Y., Deng, F., Yan, Y., Gao, F. (2019) 'A fault localization method based on conditional probability', *Proc. IEEE 19th Int. Conf. on QRS-C*, pp. 213–218. doi: 10.1109/QRS-C.2019.00050.
9. Landsberg, D., Barr E.T. (2018) 'Doric: Foundations for statistical fault localization', *ArXiv*, art. 1810.00798, available at: https://arxiv.org/abs/1810.00798 (accessed July, 2023).
10. Keller, F., Grunske, L., Simon, H., Filieri, A., van Hoorn, A., Lo, D. (2017) 'A critical evaluation of spectrum-based fault localization techniques on a large-scale software system', *Proc. IEEE Int. Conf. QRS*, pp. 114–125. doi: 10.1109/QRS.2017.22.
11. Kong, L., Wang, J., Zhou, S., Wang, M. (2021) 'A multiple-fault localization method for embedded software with applications in engineering', *Math. Problems in Engineering*, 2021, pp. 1–17, art. 7038979. doi: 10.1155/2021/7038979.
12. Heris, S.R., Keyvanpour, M. (2019) 'Effectiveness of weighted neural network on accuracy of software fault localization', *Proc. ICWR*, pp. 100–104. doi: 10.1109/ICWR.2019.8765262.
13. Chalambous, Y., Tihanyi, N., Jain, R., Sun, Y. et.al (2023) 'A new era in software security: Towards self-healing software via large language models and formal verification', *ArXiv*, art. 2305.14752, available at: https://arxiv.org/abs/2305.14752 (accessed July, 2023).
14. Haque, M.A., Li, S. (2023) 'The potential use of ChatGPT for debugging and bug fixing', *EAI Endorsed Trans AI Robotics*, 2. doi: 10.4108/airo.v2i1.3276.
15. Bouzenia, I., Ding, Y., Pei, K., Ray, B., Pradel, M. (2023) 'TraceFixer: Execution trace-guided program repair', *ArXiv*, art. 2304.12743, available at: https://arxiv.org/abs/2304.12743 (accessed July, 2023).
16. Hecht-Nielsen, R. (1987) 'Kolmogorov's mapping neural network existence theorem', *Proc. IEEE Int. Conf. on Neural Networks*, 3, pp. 11–14.
17. Nikolenko, S., Kadurin, A., Arkhangelskaya, E. (2018) *Deep Learning*. St. Petersburg, 480 p. (in Russ.).

Авторы
Достовалова Анастасия Михайловна [1, 2],
техник-программист,
инженер-исследователь,
dost.bmstu99@gmail.com
Шерминская Анна Алексеевна [1],
начальник сектора, SherminskayaAA@yandex.ru

[1] Концерн «Моринсис-Агат»,
г. Москва, 105275, Россия
[2] Федеральный исследовательский центр
«Информатика и управление» РАН,
г. Москва, 119333, Россия

Authors
Anastasiya M. Dostovalova [1, 2],
Technical Fellow Programmer,
Research-Engineer
dost.bmstu99@gmail.com
Anna A. Sherminskaya [1], Head of Sector,
SherminskayaAA@yandex.ru

[1] Morinsis-Agat Concern JSC,
Moscow, 105275, Russian Federation
[2] Federal Research Center
"Computer Science and Control" RAS,
Moscow, 119333, Russian Federation

УДК 004.42, 004.051 doi: 10.15827/0236-235X.142.561-572 2023. Т. 36. № 4. С. 561–572

Создание инструментария для векторизации тела плоского цикла с помощью векторных инструкций AVX-512

А.А. Рыбаков
А.Н. Швиндт

Ссылка для цитирования
Рыбаков А.А., Швиндт А.Н. Создание инструментария для векторизации тела плоского цикла с помощью векторных инструкций AVX-512 // Программные продукты и системы. 2023. Т. 36. № 4. С. 561–572. doi: 10.15827/0236-235X.142.561-572
Информация о статье
Поступила в редакцию: 23.08.2023 После доработки: 01.09.2023 Принята к публикации: 04.09.2023

Аннотация. Работа направлена на изучение подходов по сокращению времени исполнения программ с помощью инструкций из набора AVX-512 для повышения эффективности векторизации программного кода. Векторизация является основной низкоуровневой оптимизацией, с помощью которой возможно кратно ускорить выполнение программы, а набор инструкций AVX-512 обладает рядом уникальных особенностей, позволяющих применять векторизацию в сложных программных контекстах. В работе исследуется программный контекст специального вида – плоский цикл, который при удовлетворении ряда требований может быть векторизован с помощью инструкций AVX-512 при практически произвольной сложности тела. Однако довольно часто оптимизирующий компилятор не в состоянии выполнить автоматическую векторизацию плоских циклов по причине наличия в них сложного управления, вызовов функций, гнезд циклов и других конструкций. Это приводит к тому, что векторизацию приходится выполнять вручную с использованием специальных функций-интринсиков. Эффективность векторизации напрямую зависит от плотности масок векторных операций, которые оказываются разреженными при сильно разветвленном управлении внутри тела плоского цикла. В работе предлагается инструментарий, позволяющий параллельно создавать скалярную и векторную версии плоского цикла и анализировать эффективность выполненной векторизации. При этом поддерживаются два режима сборки тестируемого кода: скалярная версия с эмуляцией и мониторингом векторных инструкций и векторная версия для исполнения на целевой машине. С одной стороны, это позволяет контролировать корректность выполненных преобразований кода, а с другой – отслеживать пути исполнения программы с низкой вероятностью для их локализации или выноса из тела цикла. С помощью созданного инструментария удалось выделить горячие плоские циклы ряда реальных приложений и повысить эффективность их векторизации.

Ключевые слова: векторизация, плоский цикл, AVX-512, низкоуровневая оптимизация, компьютерные приложения
Благодарности. Работа выполнена в МСЦ РАН в рамках государственного задания по теме FNEF-2022-0016. В исследованиях использован суперкомпьютер МВС-10П

Введение. Векторизация является основной низкоуровневой оптимизацией, с помощью которой можно достичь кратного ускорения компьютерных приложений. Использование векторных инструкций позволяет применять одинаковые операции сразу к нескольким наборам входных данных, упакованных в векторные регистры. На сегодняшний день наиболее продвинутым набором векторных инструкций является AVX-512, поддержку которого можно встретить в микропроцессорах Intel и AMD. Инструкции AVX-512 работают с векторными регистрами размера 512 бит, каждый из которых вмещает в себя 8 элементов в формате вещественных чисел двойной точности, что с учетом комбинированных операций приводит к возможности выполнения 16 операций сложения и умножения за одну векторную команду. Из этого следует, что на микропроцессорах с поддержкой AVX-512 без использования этих инструкций даже теоретически невозможно добиться производительности более 6,25 % от пиковой. Особенности векторных инструкций AVX-512 позволяют применять их для векторизации достаточно сложного программного контекста, однако оптимизирующий компилятор не всегда успешно справляется с этой задачей. Для обеспечения программисту возможности прямого использования векторных инструкций существует набор функций-интринсиков, которые в дальнейшем заменяются компилятором на конкретные векторные инструкции или последовательности инструкций. Использование интринсиков значительно упрощает разработку векторизованного кода, однако этот процесс все еще остается слишком трудоемким.

Можно считать, что впервые набор векторных инструкций AVX-512 был поддержан в 2016 году в микропроцессорах Intel Xeon Phi Knights Landing (так как более раннее поколение Intel Xeon Phi Knights Corner представляет собой ускоритель и векторный код для него не является x86 совместимым). С этого момента

вопросы векторизации приложений активно обсуждаются в научных статьях. Отметим наиболее заметные актуальные работы.

В качестве практического руководства по созданию векторизованного программного кода с помощью функций-интринсиков для языка программирования C++ и с помощью языка ассемблера для широкого спектра практических задач, включая задачи линейной алгебры, обработки изображений, реализацию сверток и другие, можно рассматривать работу [1].

Среди научно-исследовательских статей в различных областях применения можно отметить [2], посвященную разработке векторизованной версии быстрой сортировки (vqsort) с помощью наборов векторных инструкций AVX2 и AVX-512, превышающей по эффективности реализацию из стандартной библиотеки (qsort) более чем в 10 раз. В работе [3] рассматриваются реализация бессеточного метода решения задачи газовой динамики и векторизация данного метода. Использование векторных инструкций AVX-512 позволило ускорить исходный код в 6,3 раза. В статье [4] описано успешное применение векторизации для алгоритма анализа генетических вариаций. Использование векторных инструкций AVX-512 позволило добиться ускорения расчетов от 7 до 12 раз по сравнению с оригинальным алгоритмом (алгоритм базируется на обработке данных формата Byte, что объясняет такое серьезное ускорение). В работе [5] рассматривается прямое применение инструкций AVX-512 для задачи обработки временных рядов, что привело к ускорению результирующего кода в 4 раза по сравнению с автоматической векторизацией, выполняемой оптимизирующим компилятором. Статья [6] посвящена оптимизации алгоритмов шифрования с открытым ключом, в частности, рассматривается оптимизация блочного варианта умножения Монтгомери. Было достигнуто ускорение основных операций от 1,5 до 1,9 раза. В статье [7] показана реализация алгоритма блочного шифрования на различных архитектурах, включая GPU и CPU. В частности, в рамках данной работы алгоритм блочного шифрования был реализован с помощью инструкций AVX-512, что позволило ускорить его в 9,5 раза по сравнению с оригиналом. В статье [8] рассматривается применение векторных инструкций для оптимизации математических операций, используемых в протоколе обмена ключами с применением суперсингулярных изогений (SIKE). В результате выполненных усовершенствований зафиксировано ускорение отдельных операций в диапазоне от 1,5 до 3,5 раза. В работе [9] описано использование инструкций AVX-512 для оптимизации быстрого преобразования Фурье. С помощью функций-интринсиков для векторных инструкций было достигнуто ускорение примерно в полтора раза по сравнению с оригинальной версией. Работа [10] посвящена применению подмножества инструкций Integer Fused Multiply-Add (AVX-512 IFMA) для повышения эффективности реализации деления больших целых чисел. В результате оптимизации было достигнуто ускорение требуемого функционала на 25–35 %. В работе [11] продемонстрирован практический подход к векторизации расчета попарного взаимодействия множества частиц. Использование инструкций AVX-512 в явном виде привело к ускорению исполнения программы в 3,3 раза.

В статье [12] предлагается подход к развитию средств автоматической векторизации расчетных циклов, основанный на анализе и эквивалентных преобразованиях графа зависимостей между операциями, расположенными в теле цикла. Этот подход позволяет переупорядочить операции внутри цикла, снизив их количество и укоротив критический путь исполнения [13], и сгруппировать для объединения в векторные инструкции. При векторизации тела циклов с помощью набора инструкций AVX-512 количество и вид используемых операций не являются препятствием для векторизации. Основная проблема – наличие в теле цикла операций передачи управления и вызовов функций. Однако набор инструкций AVX-512 позволяет оперировать даже таким программным контекстом [14].

Векторные инструкции AVX-512

Набор инструкций AVX-512 – это 512-битное расширение 256-битных AVX-инструкций из набора команд Intel x86, поддержанное в семействах микропроцессоров, начиная с Intel Xeon Skylake и Intel Xeon Phi Knights Landing. Поддержка данного набора инструкций внедряется и для микропроцессоров AMD.

Множество инструкций AVX-512 состоит из нескольких подмножеств, их перечень расширяется с появлением новых поколений микропроцессоров (при этом в различных поколениях микропроцессоров поддержаны разные наборы подмножеств инструкций AVX-512) [15]. Перечислим только некоторые из этих подмножеств. Основной набор векторных инструкций

с поддержкой маскирования – AVX-512F (Foundation) – включает в себя векторные аналоги для большинства арифметических операций, которые используются в расчетах. Инструкции предварительной подкачки данных из памяти – AVX-512PF (PreFetch) – были поддержаны в микропроцессорах Intel Xeon Phi KNL для оптимизации загрузки данных. Команды для вычисления экспоненты и обратных значений – AVX-512ER (Exponential and Reciprocal) – крайне полезный набор, позволяющий векторизовать вычисления, содержащие, в частности, операции возведения в степень. Инструкции для определения конфликтов, которые используются для анализа пересечения адресов обращений в память, – AVX-512CD (Conflict Detection). Дополнительные инструкции для работы с данными размера Byte, Word, Double-word, Quad-word – AVX-512BW, AVX-512DQ.

Из других подмножеств можно упомянуть команды для работы с 52-битными целочисленными значениями, специальные команды для работы с нейросетями и AES-шифрованием, реализацию арифметики полей Галуа, имплементацию специальных битовых операций, а также новый класс комбинированных операций, позволяющий еще больше повысить пиковую производительность процессоров.

Для поддержки выборочного применения операций над упакованными данными к конкретным элементам вектора используются маски. Большинство инструкций из набора AVX-512 могут использовать специальные регистры масок. Всего таких регистров 8 (k0–k7). Длина каждой маски составляет 64 бита. Маски k0–k7 используются в командах для осуществления условной операции над элементами упакованных данных (если соответствующий бит выставлен в 1, то операция выполняется, а точнее – результат операции записывается в соответствующий элемент вектора назначения) или для слияния элементов данных в регистр назначения. Также маски можно использовать для выборочного чтения из памяти и записи в память элементов векторов, для аккумулирования результатов логических операций над элементами векторов.

По схеме работы можно выделить несколько групп операций AVX-512. Упакованные операции с одним операндом zmm (512-битный вектор) и одним результатом zmm получают на вход один вектор и применяют к каждому его элементу конкретную функцию, получая результат того же размера, который по маске записывается в выходной вектор. Примерами таких операций являются получение абсолютного значения, извлечение корня, округление, операции сдвигов на фиксированное количество разрядов и другие. Упакованные операции с двумя операндами zmm и одним результатом zmm отличаются от предыдущего класса только тем, что применяемая функция является бинарной. К данной группе относятся операции поэлементного сложения, вычитания, умножения, деления, сдвига на переменное количество разрядов и другие. Упакованные операции с двумя операндами zmm и результатом маской выполняют поэлементное сравнение двух векторов и записывают полученный результат в маску. Операции конвертации предназначены для преобразования элементов вектора из одного формата в другой, к ним относятся наборы команд cvt и pack. Упакованные комбинированные операции принимают на вход сразу три zmm-вектора, a, b, c, и поэлементно вычисляют значения вида $\pm a \cdot b \pm c$, которые по маске записываются в выходной вектор. Операции перестановок не выполняют арифметические действия, а только переставляют части вектора в произвольном порядке, определяемом типом операции и дополнительными параметрами. Данная группа представлена большим набором разнообразных операций – unpck, shuf, align, blend, perm. Операции пересылок предназначены для перемещения последовательных данных между регистрами, а также между памятью и регистром. Поддержаны также операции пересылки элементов данных, расположенных не последовательно, а с произвольными смещениями от заданного базового адреса в памяти (операции gather и scatter), а также операции пересылки с дублированием элементов, позволяющие переместить одно значение сразу в несколько элементов вектора. Операции предварительной подкачки данных используются для увеличения вероятности того, что к моменту исполнения команды данные уже будут в кэше памяти. Кроме того, поддержаны другие операции с более сложной логикой, среди которых определение класса вещественного числа, реализация логических функций от трех аргументов, операции определения конфликтов и другие.

С целью упрощения векторизации исходного кода для компилятора icc разработаны специальные функции – интринсики, определенные в заголовочном файле immintrin.h. Они покрывают практически все инструкции AVX-512, избавляют от необходимости вруч-

ную писать ассемблерный код и позволяют использовать встроенные типы данных для 512-битных векторов (_m512, _m512i, _m512d) и масок (_mmask8, _mmask16 и другие). Некоторые интринсики соответствуют не отдельной команде, а целой последовательности. Примером могут служить функция сложения всех элементов вектора (_mm512_reduce_add_ps) и другие reduce-операции. Из множества интринсиков можно выделить различные группы функций, схожих по структуре. Для большинства операций AVX-512 реализованы соответствующие инстринсики, раскрывающиеся в одну конкретную операцию. Среди них арифметические и побитовые операции, операции чтения из памяти и записи в память, операции конвертации, слияние двух векторов, нахождение обратных значений, получение минимума и максимума из двух значений, операции сравнения, операции с масками, комбинированные операции и другие. Некоторые интринсики, особенно предназначенные для выполнения упакованных трансцендентных операций, раскрываются просто в вызов библиотечной функции (например, _mm512_log_ps, _mm512_hypot_ps, тригонометрические функции). Полный перечень всех доступных функций-интринсиков размещена по адресу https://www.laruence.com/sse/.

Понятие плоского цикла

Векторизация программного контекста автоматически не может быть применена к коду произвольного вида. Определим подходящий для векторизации программный контекст специального вида – плоский цикл – и опишем его свойства.

На рисунке 1 слева продемонстрирована функции fun, которая получает на вход три аргумента – a, b и c и на их основании формирует результат, состоящий из трех значений, – *x*, *y*, *z*. Будем считать, что результат выполнения этой функции зависит только от значений a, b и c (например, отсутствуют побочные эффекты через глобальную память или операции ввода-вывода). Если рассмотреть вместо одного вызова функции fun несколько вызовов (*w* штук) с разными наборами входных параметров, то их можно трактовать как вызов некоторой функции FUN, входными параметрами которой являются векторы A, B и C длины *w*, а результатом – набор из векторов X, Y и Z также длины *w*. В данном случае можно говорить, что функция FUN представляет собой реализацию векторизованной функции fun при ширине векторизации *w* (рис. 1 справа). В этом случае семантику функции FUN можно записать следующим образом:

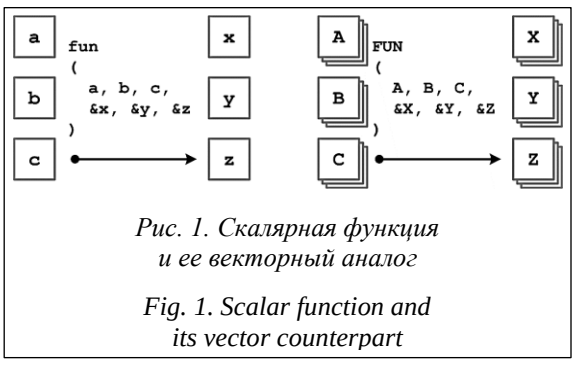


Рис. 1. Скалярная функция и ее векторный аналог

Fig. 1. Scalar function and its vector counterpart

```
// Плоский цикл в скалярном виде.
for (int i = 0; i < w; i++)
{
  fun(A[i], B[i], C[i], X[i], Y[i], Z[i];
}

// Плоский цикл в векторном виде.
FUN(A, B, C, X, Y, Z);
```

Цикл такого вида будем называть плоским. Определим характеристики, присущие плоскому циклу. Будем считать, что плоский цикл – это цикл for, индуктивная переменная которого меняется от 0 до $w - 1$, где w – ширина векторизации (например, при работе с набором инструкций AVX-512 и с вещественными значениями формата single precision ширина векторизации равна 16, то есть один zmm-регистр вмещает 16 значений), а также, что внутри плоского цикла на i-й итерации все обращения в память (и вообще все обращения к глобальным данным) имеют вид a[i]. Такое ограничение гарантирует отсутствие конфликтов между итерациями при работе с памятью. Таким образом, итерации плоского цикла становятся независимыми между собой, то есть могут выполняться в произвольном порядке, в том числе и одновременно.

В случае нарушения требования на вид обращения в память будем называть цикл квазиплоским. Для таких циклов все еще доступны некоторые приемы векторизации. Так, в [16] рассматривается подход к векторизации сортировки Шелла, реализованной в виде гнезда из трех циклов. В данной реализации сортировки запись в память имеет вид a[i + off]. Это приводит к необходимости использовать векторную операцию множественной записи в память scatter, что в совокупности с нерегулярным количеством итераций вложенных циклов приводит к очень скромным результатам по ускоре-

нию (примерно в 2 раза). В данной работе такие квазиплоские циклы не рассматриваются.

Плоские циклы, обладающие описанными свойствами, представляют собой удобный контекст для векторизации, и в большинстве случаев они могут быть векторизованы с помощью векторных инструкций AVX-512 путем перевода тела цикла в предикатное представление [17] и замены скалярных инструкций векторными аналогами, реализованными с помощью интринсиков. Многие практические вычислительные задачи состоят из выполнения однотипных вычислений, применяемых к разным наборам данных, которые можно сгруппировать, трансформировав в плоский цикл, как это продемонстрировано на рисунке 1 на примере функции fun.

Слияние ветвей исполнения при векторизации

Основной причиной потери производительности при векторизации плоского цикла является наличие в теле цикла операций передачи управления (конструкции if, else, switch и другие). Наличие команд передачи управления означает, что внутри итерации цикла некоторые инструкции, до которых управление не дойдет, не должны исполняться. В векторизованной версии цикла это означает, что некоторые векторные инструкции должны обрабатывать не все свои элементы, а только часть их, что регулируется масками. Если назвать плотностью маски отношение количества в ней единичных битов к общей длине маски, то можно отметить, что эффективность векторизации коррелирует с плотностью маски (при низкой плотности масок векторных операций эффективность векторного кода не может быть высокой, обратное не всегда верно).

Рассмотрим векторизацию простого программного контекста – тела плоского цикла, состоящего из двух линейных участков (left и right), выполняющихся под противоположными условиями (предикатами). Пусть длины линейных участков равны t и αt, где под длиной линейного участка можно понимать количество операций в нем, стоящих на критическом пути исполнения (рис. 2). Пусть вероятность выполнения условия для перехода на первый линейный участок равна p, тогда вероятность условия для перехода на второй линейный участок равна $(1 - p)$ (рис. 2а). Если количество итераций рассматриваемого цикла равно w, то общее время выполнения скалярного варианта цикла $T_1 = (p + (1 - p)\alpha)tw$.

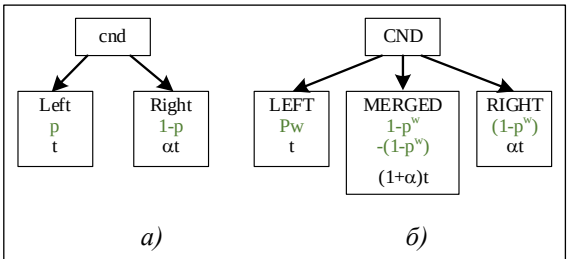


Рис. 2. Слияние двух линейных участков при векторизации

Fig. 2. Merging two linear sections during vectorization

Пусть необходимо векторизовать рассматриваемый цикл с длиной векторизации w. В общем случае в блоке CND должна быть сформирована маска длины w, каждый бит которой соответствует выполнению блока left или right в скалярной версии цикла. После этого для всех скалярных инструкций из блоков left и right должны быть взяты векторные аналоги под противоположными масками и помещены в единый блок MERGED. Тогда длина блока MERGED, а значит, и время выполнения векторизованного цикла, будет $T_w = (1 + \alpha)t$. В этом случае теоретическая эффективность векторизации составит

$$e = \frac{T_1}{T_w w} = \frac{p + (1 - p)\alpha}{1 + \alpha}.$$

Заметим, что в случае одинаковых по длине участков left и right ($\alpha = 1$) эффективность векторизации равна 0,5 даже при стремлении вероятности выполнения одного из этих участков к нулю ($p = 0$ или $p = 1$).

Если присутствует дополнительная информация об условии выбора линейного участка в скалярной версии цикла, то при векторизации можно произвести дополнительную оптимизацию. Зачастую в расчетных приложениях условия выбора линейного участка на разных итерациях плоского цикла не являются независимыми. Более того, данные условия могут меняться достаточно медленно при переходе от одной итерации к другой. Это приводит к тому, что маска, сформированная в блоке CND, нередко оказывается пустой или, наоборот, полной. Это означает, что в скалярном варианте цикла на каждой из w итераций всегда выполняется линейный участок left или right. Для повышения эффективности векторной версии цикла может быть создан векторизованный блок LEFT, который будет выполняться с вероятностью p^w. Аналогично для линейного участка right может быть создан векторизован-

ный блок RIGHT, который будет выполняться с вероятностью $(1 - p)^w$. Таким образом, вероятность выполнения векторизованного участка MERGED, содержащего все векторизованные операции тела цикла, составит $1 - p^w - (1-p)^w$. Посчитав математическое ожидание длины векторизованного тела цикла, содержащего все три участка – LEFT, MERGED и RIGHT, получим теоретическую эффективность векторизации:

$$e = \frac{T_1}{T_w w} = \frac{p + \alpha(1-p)}{\left(1-(1-p)^w\right) + \alpha\left(1-p^w\right)}.$$

В этом случае при одинаковых по длине участках left и right и при стремлении вероятности выполнения одного из них к нулю эффективность векторизации стремится к единице. С другой стороны, такое преобразование требует введения дополнительных проверок маски на пустоту или полноту (маски можно сравнивать как обычные беззнаковые целые числа), а также дублирования кода, поэтому выполнять его следует ограниченно и при наличии информации мониторинга о вероятностях переходов между линейными участками.

Отдельно следует отметить, что набор векторных инструкций AVX-512 содержит ряд инструкций, логика которых соответствует конструкциям с передачей управления. Это операции abs (реализуется с помощью обнуления знакового бита), min, max, blend (см. таблицу).

Инструкции, семантика которых отражает передачу управления

Instructions whose semantics reflect control transfer

Инструкция	Семантика
abs(x)	(x > 0) ? x : -x
min(x, y)	(x <= y) ? x : y
max(x, y)	(x >= y) ? x : y
blend(cnd, x, y)	cnd ? x : y

Указанные в таблице инструкции с семантикой, содержащей передачу управления, имеют векторные аналоги в AVX-512, поэтому с их помощью можно векторизовать другие похожие операции. Например, операция to_bounds(x, lo, hi) приведения величины в заданный диапазон может быть выражена как to_bounds(x, lo, hi) = = min(hi, max(lo, x)), а значит, и векторизована. Конечно, наиболее ценной среди этих операций является blend, так как с ее помощью можно производить слияние различных линейных участков с произвольной степенью разветвленности управления.

Пример выполнения векторизации кода с сильно разветвленным управлением

В качестве примера векторизации плоского цикла с сильно разветвленным управлением рассмотрим цикл, тело которого включает нахождение минимального положительного корня квадратного уравнения (рис. 3, слева). Данный программный контекст с обработкой всех краевых случаев содержит большое количество линейных участков при их небольшой длине, поэтому прямое слияние всех этих участков крайне неэффективно.

Для анализа возможностей векторизации тела данного плоского цикла рассмотрим его граф потока управления [18] – ориентированный граф, узлами которого являются линейные участки программы, связанные между собой операциями передачи управления. В графе потока управления выделен начальный узел (start node), в который не входит ни одна дуга, а также конечный (stop node), из которого не выходит ни одна дуга. Таким образом, граф потока управления является гамаком. На рисунке 4 слева представлен граф потока управления рассматриваемого фрагмента кода. Можно заметить, что большое количество линейных участков рассматриваемого графа может быть удалено. Например, все линейные участки, обнуляющие величину *h*, могут быть удалены при условии, что переменная *h* инициализирована нулем в стартовом узле. Также некоторые линейные участки могут быть непосредственно слиты с помощью семантики операций max и blend (на рисунке 4 сливаемые в один узел регионы отмечены пунктиром). После выполнения преобразований по уменьшению количества линейных участков получим граф потока управления в более компактном виде (рис. 4, справа).

Следующим шагом векторизации является запись полученного кода в предикатной форме, то есть без операций передачи управления, где для каждой операции указано условие ее выполнения (рис. 3, справа сверху). После записи кода в предикатной форме остается один механический шаг – замена всех скалярных инструкций векторными аналогами. При этом предикаты скалярных операций при векторизации трансформируются в векторные маски. Результирующий код представлен на рисунке 3 (справа внизу).

Векторизация плоских циклов с помощью инструмента flatvec

Подход к векторизации плоского цикла путем перевода скалярной версии в предикатную

```
// Исходный скалярный
код тела цикла.
if (a == 0.0)
{
    if (b == 0.0)
    {
        h = 0.0;
    }
    else
    {
        h = -c / b;
        if (h < 0.0)
        {
            h = 0.0;
        }
    }
}
else
{
    b /= a;
    c /= a;
    double d =
= b * b - 4.0 * c;

    if (d < 0.0)
    {
        h = 0.0;
    }
    else
    {
        double sd = sqrt(d);
        double h1 =
= 0.5 * (-b - sd);
        if (h1 > 0.0)
        {
            h = h1;
        }
        else
        {
            double h2 =
= 0.5 * (-b + sd);
            if (h2 > 0.0)
            {
                h = h2;
            }
            else
            {
                h = 0.0;
            }
        }
    }
}
```

```
// Код в предикатной форме для векторизации.
h = 0.0;
p1 = (a == 0.0);
p2 = p1 && (b != 0.0);
h = std::max(-c / b, 0.0);            [?  p2]
b /= a;                               [? !p1]
c /= a;                               [? !p1]
d = b * b - 4.0 * c;                  [? !p1]
p4 = !p1 && (d >= 0.0);
sd = sqrt(d);                         [?  p4]
h1 = 0.5 * (-b - sd);                 [?  p4]
h = (h1 > 0.0)
    ? h1
    : std::max(0.5 * (-b + sd), 0.0); [?  p4]


// Векторный код с использованием
функций-интринсиков.
h = zero;

__mmask16 p1 = _mm512_cmpeq_ps_mask(a, zero);
__mmask16 p2 = _mm512_mask_cmpneq_ps_mask(p1,
b, zero);

h = _mm512_maskz_max_ps(p2,
        _mm512_maskz_div_ps(p2,
_mm512_sub_ps(zero, c), b),
        zero);

__mmask16 np1 = _mm512_knot(p1);

b = _mm512_mask_div_ps(b, np1, b, a);
c = _mm512_mask_div_ps(c, np1, c, a);

__m512 d = _mm512_sub_ps(_mm512_mul_ps(b, b),
                _mm512_mul_ps(four, c));

__mmask16 p4 = _mm512_mask_cmpge_ps_mask(np1,
d, zero);

__m512 sd = _mm512_mask_sqrt_ps(sd, p4, d);

__m512 h1 = _mm512_mul_ps(half,
                _mm512_sub_ps(zero,
_mm512_add_ps(b, sd)));

h = _mm512_mask_mov_ps(h, p4,
        _mm512_mask_blend_ps(
            _mm512_cmpgt_ps_mask(h1, zero),
            _mm512_max_ps(
                _mm512_mul_ps(half,
_mm512_sub_ps(sd, b)),
                 zero),
            h1));
```

Рис. 3. Пример векторизации плоского цикла

Fig. 3. Flat loop vectorization example

форму с последующей заменой скалярных инструкций на векторные аналоги является достаточно трудоемкой задачей, при выполнении которой велика вероятность внесения ошибки. Для упрощения описанного процесса предлагается инструментарий, реализация которого ведется в открытом репозитории (https://github.com/r-aax/flatvec).

Репозиторий flatvec состоит из отдельных тестов по векторизации программного контекста плоских циклов (http://www.swsys.ru/uploaded/image/2023-4/16.jpg). Имя теста имеет структуру case_<имя>, в качестве параметров функция принимает количество тестируемых векторов, а также повторений теста. Внутри теста выполняется инициализация мас-

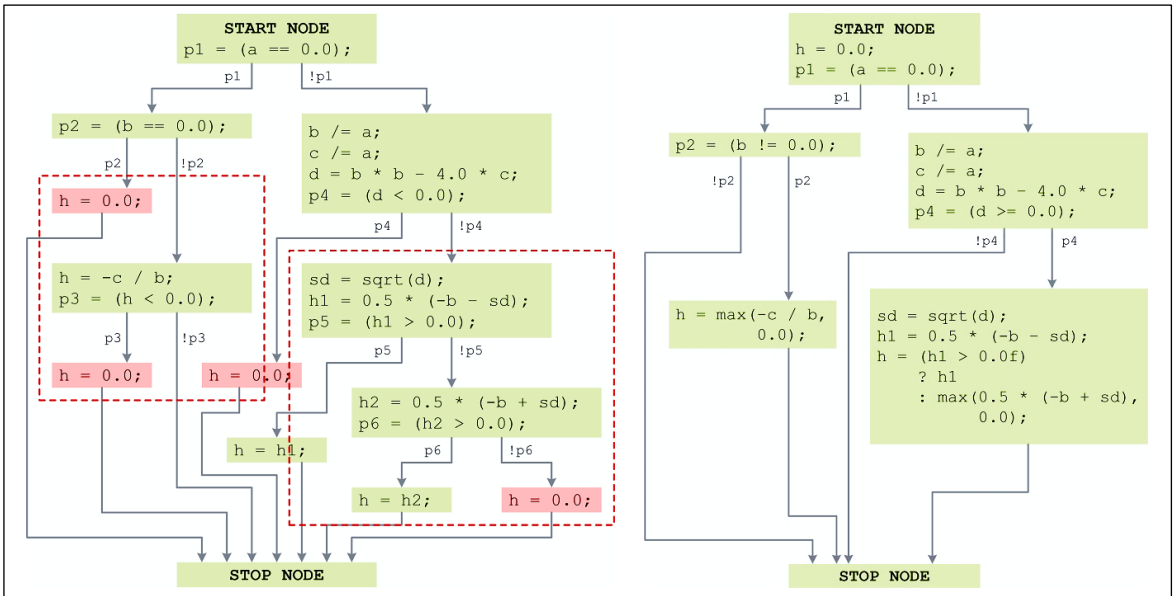


Рис. 4. Граф потока управления плоского цикла

Fig. 4. Control flow graph of flat loop

сивов входных данных, содержащих требуемое количество проверяемых векторов. Инициализация массивов входных данных может выполняться с помощью генерации псевдослучайных чисел или загрузки данных из текстовых файлов. После этого массивы данных обрабатываются с помощью скалярной версии цикла (scase_<имя>) и векторной версии того же цикла (vcase_<имя>). После выполнения вычислений скалярной и векторной версиями цикла выполняется проверка на совпадение результатов вычислений. В случае совпадения векторизация теста считается верной. Функция scase_<имя>_1 содержит скалярную реализацию одной итерации исходного цикла, в данной функции осуществляется работа с атомарными типами данных. Функция vcase_<имя>_1 содержит векторную реализацию *w* итераций исходного цикла, в данной функции работа ведется с векторными регистрами (для примера, работающего с входными данными типа float, значение *w* равно 16, так как в одном векторном 512-битном регистре помещаются 16 значений типа float).

Все помещенные в репозиторий тесты могут быть скомпилированы в двух режимах. Первый – режим исполнения и замера ускорения от векторизации. В данном режиме используемые функции-интринсики раскрываются в инструкции из набора AVX-512, на выходе получается исполняемый векторизованный код. Второй – режим эмуляции функций-интринсиков для отладки и мониторинга. В этом режиме векторные регистры и маски представлены классами fv::ZMM и fv::Mask, а функции-интринсики полностью эмулируются. В режиме эмуляции доступны мониторинг содержимого векторных регистров и масок, а также сбор информации по плотности масок и эффективности векторизации.

Сравним результаты работы некоторых тестов в двух различных режимах сборки. Результат работы в режиме исполнения:

```
guessp    : OK
Global stat:
  vector opers              : 0
  scalar opers              : 0
  mask opers                : 0
  theoretical acceleration  : -nan
  mean masks density        : -nan
  real time acceleration    : 9.354823
--------------------------
prefun    : OK
Global stat:
  vector opers              : 0
  scalar opers              : 0
  mask opers                : 0
  theoretical acceleration  : -nan
  mean masks density        : -nan
  real time acceleration    : 11.446910
--------------------------
sample    : OK
Global stat:
  vector opers              : 0
  scalar opers              : 0
  mask opers                : 0
  theoretical acceleration  : -nan
  mean masks density        : -nan
     real time acceleration   : 3.022369
```

Можно отметить, что в этом режиме недоступна никакая статистическая информация, кроме непосредственного измеренного ускорения от векторизации (real time acceleration).

Представим результат работы в режиме эмуляции:

```
guessp    : OK
Global stat:
  vector opers             : 610
  scalar opers             : 8285
  mask opers               : 60
  theoretical acceleration : 13.581967
  mean masks density       : 0.848873
  real time acceleration   : 0.058976
-------------------------
prefun    : OK
Global stat:
  vector opers             : 260
  scalar opers             : 3200
  mask opers               : 30
  theoretical acceleration : 12.307692
  mean masks density       : 0.769231
  real time acceleration   : 0.013309
-------------------------
sample    : OK
Global stat:
  vector opers             : 590
  scalar opers             : 7615
  mask opers               : 50
  theoretical acceleration : 12.906780
  mean masks density       : 0.806674
     real time acceleration   : 0.01155
```

В этом режиме, наоборот, доступна вся информация, которая собирается в процессе эмуляции, но время фактического ускорения векторизации не имеет смысла, поскольку эмуляция векторных инструкций заведомо выполняется значительно дольше исходного скалярного кода. В выводимой статистике можно отметить, что у всех тестов наблюдается достаточно высокое теоретическое ускорение от векторизации (после theoretical acceleration в режиме эмуляции), тогда как реальное изменяется в широких пределах (поле real time acceleration в режиме исполнения). Это связано с тем, что теоретическое ускорение рассчитывается через среднюю плотность масок в векторных инструкциях. Однако средняя плотность масок при векторизации кода может быть завышена. Для примера можно рассмотреть плоский цикл с телом c[i] = cnd ? (a[i] + b[i]) : (a[i] – b[i]). Теоретическое ускорение векторизованного кода, записанного в виде _mm512_mask_blend_ps(cnd, _mm512_add_ps(a, b), _mm512_sub_ps(a, b)), будет максимальным, так как все векторные инструкции обрабатывают все элементы своих аргументов. Но из полученных результатов инструкций add и sub для получения конечного результата используется только половина (вторая половина будет проигнорирована при выполнении инструкции blend). Таким образом, для получения более адекватной оценки теоретического ускорения необходимо проставлять маски во всех векторных инструкциях, хотя на реальную производительность это никак не влияет.

Рассмотренный подход к векторизации плоских циклов с помощью функций-интринсиков для инструкций AVX-512 был опробован на ряде реальных задач. В статье [19] приведено описание векторизации для матричных операций, а также для гнезд циклов с нерегулярным числом итераций на примере сортировки Шелла. В работе [20] в качестве объекта векторизации рассматривается программный контекст точного римановского решателя, в котором содержатся гнезда вложенных циклов, вызовы функций и сложное управление. Использование инструкций AVX-512 позволило добиться ускорения на данном программном коде примерно в 7 раз.

Заключение

Набор векторных инструкций AVX-512 оказывает существенное влияние на производительность суперкомпьютерных приложений. Наличие специальных векторных операций с выборочной обработкой отдельных элементов позволяет выполнять векторизацию достаточно сложного программного контекста. В работе рассмотрена специальная конструкция, называемая плоским циклом. На плоский цикл накладывается ряд ограничений, касающихся его структуры и порядка обращений в память. При выполнении этих ограничений плоский цикл может быть успешно векторизован с помощью специальных функций-интринсиков, объявленных в заголовочном файле immintrin.h. На первый взгляд может показаться, что ограничения, накладываемые на плоские циклы, слишком жесткие, однако значительная часть программного кода в расчетных приложениях может быть представлена в виде композиции плоских циклов без особых усилий. Для облегчения процесса векторизации плоского цикла путем записи программного кода в предикатном виде с последующей заменой всех операций на векторные аналоги был разработан инструмент flatvec, в котором поддержана возможность сборки тестов в двух режимах. Первый режим предназначен для по-

лучения результирующего кода для целевого микропроцессора, тогда как во втором режиме осуществляется эмуляция всех векторных инструкций, что позволяет выполнить отладку и мониторинг создаваемого векторного кода. Предложенный подход был использован для векторизации кода в ряде различных расчетных приложений для целевых микропроцессоров Intel Xeon Phi Knights Landing и Intel Xeon Cascade Lake. На сегодня существует проблема завышения показателя теоретического ускорения от векторизации в режиме эмуляции, что может приводить к излишне оптимистичным ожиданиям ускорения в реальных вычислениях. В дальнейшем планируется справиться с этой проблемой путем учета тех результатов векторных инструкций, которые фактически не участвуют в дальнейших вычислениях. Также одной из задач, которые необходимо решить для развития инструмента, является автоматическое выделение маловероятных ветвей исполнения из тела плоского цикла для выноса их из векторной версии путем проверки масок на пустоту.

Список литературы

1. Kusswurm D. Modern parallel programming with C++ and Assembly Language. X86 SIMD development using AVX, AVX2, and AVX-512. CA, Apress Berkeley Publ., 2022, 633 p. doi: 10.1007/978-1-4842-7918-2.

2. Blacher M., Giesen J., Sanders P., Wassenberg J. Vectorized and performance-portable Quicksort. ArXiv, 2022, art. 2205.05982, pp. 1–21. URL: https://arxiv.org/abs/2205.05982 (дата обращения: 14.06.2023).

3. Long S., Fan X., Chao L., Yi L., Fan S., Guo X.-W., Yang C. VecDualSPHysics: A vectorized implementation of Smoothed Particle Hydrodynamics method for simulating fluid flows on multi-core processors. J. of Computational Phys., 2022, vol. 463, art. 111234. doi: 10.1016/j.jcp.2022.111234.

4. Ponte-Fernández C., González-Domínguez J., Martín M.J. A SIMD algorithm for the detection of epistatic interactions of any order. Future Generation Comput. Sys., 2022, vol. 132, pp. 108–123. doi: 10.1016/j.future.2022.02.009.

5. Quislant R., Fernandez I. Time series analysis acceleration with advanced vectorization extensions. The J. of Supercomputing, 2023, vol. 79, no. 9, pp. 10178–10207. doi: 10.1007/s11227-023-05060-2.

6. Buhrow B., Gilbert B., Haider C. Parallel modular multiplication using 512-bit advanced vector instructions. J. of Cryptographic Engineering, 2022, vol. 12, pp. 95–105. doi: 10.1007/s13389-021-00256-9.

7. Choi H., Seo S.C. Efficient parallel implementations of PIPO Block Cipher on CPU and GPU. IEEE Access, 2022, vol. 10, pp. 85995–86007. doi: 10.1109/ACCESS.2022.3198707.

8. Cheng H., Fotiadis G., Großschädl J., Peter Y.A. Ryan. Highly vectorized SIKE for AVX-512. IACR Transactions on Cryptographic Hardware and Embedded Sys., 2022, no. 2, pp. 41–68. doi: 10.46586/tches.v2022.i2.41-68.

9. Sansone G., Cococcioni M. Experiments on speeding up the recursive fast Fourier transform by using AVX-512 SIMD instructions. Proc. ApplePies. LNEE, 2023, vol. 1036, pp. 255–263. doi: 10.1007/978-3-031-30333-3_34.

10. Edamatsu T., Takahashi D. Fast multiple-precision integer division using Intel AVX-512. IEEE Transactions on Emerging Topics in Computing, 2023, vol. 11, no. 1, pp. 224–236. doi: 10.1109/TETC.2022.3196147.

11. Медакин П.О., Никулин Р.Н., Авдеюк О.А., Королева И.Ю., Павлова Е.С., Лемешкина И.Г. Векторизация и распараллеливание метода «частица-частица» // Инженерный вестник Дона. 2021. № 1. URL: http://ivdon.ru/ru/magazine/archive/n1y2021/6800 (дата обращения: 30.06.2023).

12. Tayeb H., Paillat L., Bramas B. Autovesk: Automatic vectorization of unstructured static kernels by graph transformations. ArXiv, 2023, art. 2301.01018, pp. 1–21. URL: https://arxiv.org/abs/2301.01018 (дата обращения: 30.06.2023).

13. Laukemann J., Hammer J., Hager G., Wellein G. Automatic throughput and critical path analysis of x86 and ARM assembly kernels. Proc. IEEE/ACM PMBS, 2019, pp. 1–6. doi: 10.1109/PMBS49563.2019.00006.

14. Shabanov B.M., Rybakov A.A., Shumilin S.S. Vectorization of high-performance scientific calculations using AVX-512 instruction set. Lobachevskii J. of Math., 2019, vol. 40, no. 5, pp. 580–598. doi: 10.1134/S1995080219050196.

15. Intel 64 and IA-32 architectures software developer's manual. Combined volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. Intel Corporation, 2017, no. 325462-079US. URL: https://archive.org/details/Intel_64_and_IA-32_Architectures_Software_Developer_s_Manual_2017_07 (дата обращения: 30.06.2023).

16. Рыбаков А.А., Шумилин С.С. Исследование эффективности векторизации гнезд циклов с нерегулярным числом итераций // Программные системы: Теория и алгоритмы. 2019. Т. 10. № 4. С. 77–96. doi: 10.25209/2079-3316-2019-10-4-77-96.

17. Carminati A., Starke R.A., de Oliveira R.S. Combining loop unrolling strategies and code predication to reduce the worst-case execution time of real-time software. Applied Computing and Informatics, 2017, vol. 13, no. 2, pp. 184–193. doi: 10.1016/j.aci.2017.03.002.

18. Sahin V.H. Turna: A control flow graph reconstruction tool for RISC-V architecture. Computing, 2023, vol. 105, pp. 1821–1825. doi: 10.1007/s00607-023-01172-y.

19. Shabanov B.M., Rybakov A.A., Shumilin S.S. Vectorization of high-performance scientific calculations using AVX-512 instruction set. Lobachevskii J. of Math., 2019, vol. 40, no. 5, pp. 580–598. doi: 10.1134/S1995080219050196.

20. Рыбаков А.А., Шумилин С.С. Векторизация римановского решателя с использованием набора инструкций AVX-512 // Программные системы: Теория и алгоритмы. 2019. Т. 10. № 3. С. 59–80. doi: 10.25209/2079-3316-2019-10-3-59-80.

Software & Systems doi: 10.15827/0236-235X.142.561-572 2023, vol. 36, no. 4, pp. 561–572

Tools for vectorizing a flat loop body using AVX-512 vector instructions

Alexey A. Rybakov
Anthony N. Shvindt

For citation
Rybakov, A.A., Shvindt, A.N. (2023) 'Tools for vectorizing a flat loop body using AVX-512 vector instructions', *Software & Systems*, 36(4), pp. 561–572 (in Russ.). doi: 10.15827/0236-235X.142.561-572
Article info
Received: 23.08.2023 After revision: 01.09.2023 Accepted: 04.09.2023

Abstract. The paper studies approaches to reducing program execution time using instructions from AVX-512 set to increase program code vectorization efficiency. Vectorization is a basic low-level optimization that can speed up program execution by several times; AVX-512 instruction set has a number of unique features that allow applying vectorization in complex program contexts. The paper investigates a special type program context – a flat loop, which can be vectorized using AVX-512 instructions with almost arbitrary body complexity if meeting a number of requirements. However, quite often an optimizing compiler fails to automatically vectorize flat loops due to complex control, function calls, nested loops, and other constructs. This leads to manual vectorization using special intrinsic functions. Vectorization efficiency directly depends on the density of vector operation masks, which turn out to be sparse with a highly branched control inside a flat loop body. The paper proposes a toolkit that allows simultaneous creating a scalar and vector version of a flat loop and analyzing performed vectorization effectiveness. At the same time, there are two modes of assembling the code under test: a scalar version with emulation and monitoring of vector instructions and a vector version for execution on a target machine. On the one hand, this allows controlling code transformations to be correct; on the other hand, this allows tracking program execution paths with a low probability for their localization or removal from a loop body. The created tools enabled identifying hot flat loops of some real applications and increasing their vectorization efficiency.
Keywords: vectorization, flat loop, AVX-512, low-level optimization, computer applications
Acknowledgements. The work was carried out at the JSC RAS in terms of FNEF-2022-0016 state assignment. The research used MVS-10P supercomputer

References

1. Kusswurm, D. (2022) *Modern Parallel Programming with C++ and Assembly Language. X86 SIMD Development Using AVX, AVX2, and AVX-512*. CA, Apress Berkeley Publ., 633 p. doi: 10.1007/978-1-4842-7918-2.
2. Blacher, M., Giesen, J., Sanders, P., Wassenberg, J. (2022) 'Vectorized and performance-portable Quicksort', *ArXiv*, art. 2205.05982, pp. 1–21, available at: https://arxiv.org/abs/2205.05982 (accessed June 14, 2023).
3. Long, S., Fan, X., Chao, L., Yi, L., Fan, S., Guo, X.-W., Yang, C. (2022) 'VecDualSPHysics: A vectorized implementation of Smoothed Particle Hydrodynamics method for simulating fluid flows on multi-core processors', *J. of Computational Phys.*, 463, art. 111234. doi: 10.1016/j.jcp.2022.111234.
4. Ponte-Fernández, C., González-Domínguez, J., Martín, M.J. (2022) 'A SIMD algorithm for the detection of epistatic interactions of any order', *Future Generation Comput. Sys.*, 132, pp. 108–123. doi: 10.1016/j.future.2022.02.009.
5. Quislant, R., Fernandez, I. (2023) 'Time series analysis acceleration with advanced vectorization extensions', *The J. of Supercomputing*, 79(9), pp. 10178–10207. doi: 10.1007/s11227-023-05060-2.
6. Buhrow, B., Gilbert, B., Haider, C. (2022) 'Parallel modular multiplication using 512-bit advanced vector instructions', *J. of Cryptographic Engineering*, 12, pp. 95–105. doi: 10.1007/s13389-021-00256-9.
7. Choi, H., Seo, S.C. (2022) 'Efficient parallel implementations of PIPO Block Cipher on CPU and GPU', *IEEE Access*, 10, pp. 85995–86007. doi: 10.1109/ACCESS.2022.3198707.
8. Cheng, H., Fotiadis, G., Großschädl, J., Peter Y.A. Ryan. (2022) 'Highly vectorized SIKE for AVX-512', *IACR Transactions on Cryptographic Hardware and Embedded Sys.*, (2), pp. 41–68. doi: 10.46586/tches.v2022.i2.41-68.
9. Sansone, G., Cococcioni, M. (2023) 'Experiments on speeding up the recursive fast Fourier transform by using AVX-512 SIMD instructions', *Proc. ApplePies. LNEE*, 1036, pp. 255–263. doi: 10.1007/978-3-031-30333-3_34.
10. Edamatsu, T., Takahashi, D. (2023) 'Fast multiple-precision integer division using Intel AVX-512', *IEEE Transactions on Emerging Topics in Computing*, 11(1), pp. 224–236. doi: 10.1109/TETC.2022.3196147.
11. Medakin, P.O., Nikulin, R.N., Avdeyuk, O.A., Koroleva, I.Yu., Pavlova, E.S., Lemeshkina, I.G. (2021) 'Vectorization and parallelization of the particle-particle method', *Engineering J. of Don*, (1), available at: http://ivdon.ru/ru/magazine/archive/n1y2021/6800 (accessed June 30, 2023) (in Russ.).
12. Tayeb, H., Paillat, L., Bramas, B. (2023) 'Autovesk: Automatic vectorization of unstructured static kernels by graph transformations', *ArXiv*, art. 2301.01018, pp. 1–21, available at: https://arxiv.org/abs/2301.01018 (accessed June 30, 2023).
13. Laukemann, J., Hammer, J., Hager, G., Wellein, G. (2019) 'Automatic throughput and critical path analysis of x86 and ARM assembly kernels', *Proc. IEEE/ACM PMBS*, pp. 1–6. doi: 10.1109/PMBS49563.2019.00006.
14. Shabanov, B.M., Rybakov, A.A., Shumilin, S.S. (2019) 'Vectorization of high-performance scientific calculations using AVX-512 instruction set', *Lobachevskii J. of Math.*, 40(5), pp. 580–598. doi: 10.1134/S1995080219050196.
15. Intel 64 and IA-32 architectures software developer's manual. Combined volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. (2017) *Intel Corporation*, 325462-079US, available at: https://archive.org/details/Intel_64_and_IA-32_Architectures_Software_Developer_s_Manual_2017_07 (accessed June 30, 2023).

16. Rybakov, A.A., Shumilin, S.S. (2019) 'Study of the vectorization efficiency of loop nests with an irregular number of iterations', *Program Systems: Theory and Applications*, 10(4), pp. 77–96 (in Russ.). doi: 10.25209/2079-3316-2019-10-4-77-96.

17. Carminati, A., Starke, R.A., de Oliveira, R.S. (2017) 'Combining loop unrolling strategies and code predication to reduce the worst-case execution time of real-time software', *Applied Computing and Informatics*, 13(2), pp. 184–193. doi: 10.1016/j.aci.2017.03.002.

18. Sahin, V.H. (2023) 'Turna: A control flow graph reconstruction tool for RISC-V architecture', *Computing*, 105, pp. 1821–1825. doi: 10.1007/s00607-023-01172-y.

19. Shabanov, B.M., Rybakov, A.A., Shumilin, S.S. (2019) 'Vectorization of high-performance scientific calculations using AVX-512 instruction set', *Lobachevskii J. of Math.*, 40(5), pp. 580–598. doi: 10.1134/S1995080219050196.

20. Rybakov, A.A., Shumilin, S.S. (2019) 'Vectorization of the Riemann solver using the AVX-512 instruction set', *Program Systems: Theory and Applications*, 10(3), pp. 41–58. doi: 10.25209/2079-3316-2019-10-3-41-58.

Авторы

Рыбаков Алексей Анатольевич [1], к.ф.-м.н., ведущий научный сотрудник, rybakov@jscc.ru
Швиндт Антоний Николаевич [2], к.т.н., докторант, shvindt@ispras.ru

[1] МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, г. Москва, 117218, Россия
[2] Институт системного программирования им. В.П. Иванникова РАН (ИСП РАН), г. Москва, 109004, Россия

Authors

Alexey A. Rybakov [1], Cand. of Sci. (Physics and Mathematics), Leading Researcher, rybakov@jscc.ru
Anthony N. Shvindt [2], Cand. of Sci. (Engineering), Doctoral Student, shvindt@ispras.ru

[1] JSCC RAS – branch of SRISA RAS, Moscow, 117218, Russian Federation
[2] Ivannikov Institute for System Programming of the Russian Academy of Sciences (ISP RAS), Moscow, 109004, Russian Federation

УДК 004.4'2 doi: 10.15827/0236-235X.142.573-581 2023. Т. 36. № 4. С. 573–581

Разработка модели жизненного цикла мобильных приложений для людей с ограниченными интеллектуальными возможностями

В.В. Гурьев
О.А. Шабалина
Н.П. Садовникова
А.А. Воронина

Ссылка для цитирования
Гурьев В.В., Шабалина О.А., Садовникова Н.П., Воронина А.А. Разработка модели жизненного цикла мобильных приложений для людей с ограниченными интеллектуальными возможностями // Программные продукты и системы. 2023. Т. 36. № 4. С. 573–581. doi: 10.15827/0236-235X.142.573-581
Информация о статье
Поступила в редакцию: 31.11.2022 После доработки: 08.06.2023 Принята к публикации: 13.06.2023

Аннотация. В статье рассмотрены модели жизненного цикла мобильных приложений и программные средства поддержки их разработки. Описаны особенности создания мобильных приложений, доступных для людей с ограниченными интеллектуальными возможностями. Одним из способов повышения доступности приложений является их адаптируемость. Такая настройка может потребовать поддержки динамического контента приложения, поэтому данные возможности необходимо предусматривать уже на этапах проектирования и реализации. Предложена модель жизненного цикла мобильных приложений для людей с ограниченными интеллектуальными возможностями, обеспечивающая сквозную поддержку этапов проектирования, реализации и эксплуатации. Разработана технология проектирования мобильных приложений на основе многомодульной архитектуры, включающая применение готовых архитектурных шаблонов для генерации базового кода приложения и повторное использование успешных проектных решений. Представлена технология проектирования адаптируемого интерфейса мобильных приложений, включающая разработку шаблонов экранов приложения, разделение элементов шаблонов на обязательные и опциональные элементы, сопоставление каждому элементу предопределенного набора возможных визуальных представлений на экране и прикрепление всех элементов по месту. Разработана система управления мобильными приложениями на основе CALS-технологии сквозной поддержки этапов проектирования, реализации и эксплуатации адаптируемых мобильных приложений. Предложен способ адаптации интерфейса мобильного приложения с помощью конфигурационной панели, встраиваемой в систему управления. Описан пример применения предложенных моделей, технологий и средств к разработке мобильной игры для тренировки навыков совершения покупок для людей с ограниченными интеллектуальными возможностями. Показано, что применение разработанной системы управления мобильными приложениями позволяет существенно сократить временные затраты на разработку приложений. Возможность применения шаблонов архитектур и готовых решений расширяет круг пользователей системы и делает ее доступной разработчикам, профессиональная деятельность которых связана с поддержкой людей с ограниченными интеллектуальными возможностями.

Ключевые слова: жизненный цикл мобильных приложений, модель жизненного цикла, MADLC-модель, пользователи с ограниченными интеллектуальными возможностями, адаптируемое мобильное приложение

Введение. Одним из требований предварительного стандарта к качеству и безопасности *мобильных приложений* (МП) (https://docs.cntd.ru/document/1200159701) является адаптируемость приложения к использованию людьми с ограниченными возможностями здоровья, самую большую группу из которых составляют люди с ограниченными интеллектуальными возможностями (Intellectual Disabilities, ID). Разработка МП для пользователей с ID в первую очередь требует решения проблемы обеспечения понятности интерфейса [1–3]. Возможности и ограничения людей с ID сильно различаются, поэтому ориентация разработчиков МП на широкий круг пользователей с ID может привести к тому, что для некоторых из них приложение будет практически недоступным. В то же время ориентация разработчиков на вполне определенные категории пользователей с ID существенно ограничивают возможности МП.

Одним из способов повышения доступности МП является их адаптируемость [4, 5]. Многие МП предусматривают те или иные возможности для настройки под конечного пользователя, в большинстве случаев это настройки интерфейса (шрифтов, цветовой палитры, звукового сопровождения и т.д.). При этом предполагается, что настраивать интерфейс должен сам пользователь. Однако для пользователей с ID может потребоваться более персонализированная адаптация интерфейса (возможность изменения количества элементов управления, выбора различных пиктограмм для отображения элементов управления, использования или неиспользования текста и т.д.). Такая адаптация требует поддержки динамического контента МП, поэтому данные возможности необ-

ходимо предусматривать уже на этапах проектирования и реализации приложений.

В последнее время появились исследования, посвященные разработке МП для людей с ID с адаптивным интерфейсом [6, 7], настраиваемым самим приложением для каждого пользователя. Такие приложения основаны на построении модели пользователя, получающей в процессе работы информацию о его персональных характеристиках, и генерации оптимального интерфейса для этого пользователя на основе этих характеристик [8]. Однако применительно к разработке МП для людей с ID информация, получаемая непосредственно от пользователя, далеко не всегда может быть адекватно интерпретирована.

Работа посвящена исследованию особенностей *жизненного цикла* (ЖЦ) МП, доступных для людей с ID, и разработке новых моделей, технологий и специализированных программных средств, обеспечивающих сквозную поддержку ЖЦ разработки МП на этапах проектирования, реализации и эксплуатации с учетом специфических особенностей этой категории пользователей.

Модели и программные средства поддержки ЖЦ разработки МП

В условиях непрерывного роста сложности и разнообразия ПО выбор правильной стратегии поддержки его ЖЦ обеспечивает существенное повышение эффективности процесса разработки и в итоге качества самого ПО [9–11]. Однако исследования возможности применения моделей ЖЦ ПО для разработки МП показывают [12, 13], что классические и традиционные модели и их методы без существенных изменений не могут быть применены непосредственно к разработке приложений. Обоснованием таких заключений являются характеристики, специфичные для разработки МП. Поэтому в современных исследованиях модель ЖЦ мобильных приложений (Mobile Application Development Lifecycle, MADLC) рассматривается как самостоятельная категория моделей ЖЦ ПО, отличающаяся содержанием и даже количеством этапов [14].

Для управления разработкой МП сейчас применяют системы MADM (Mobile Application Development Management), типовой функционал которых включает конструктор интерфейсов, функции доработки программного кода и публикации обновлений приложений. Использование MADM-системы позволяет снизить трудоемкость при разработке МП, а также повысить его качество. Для администрирования МП, разработанных с целью использования в различных видах профессиональной деятельности, и управления пользователями таких приложений на этапе эксплуатации применяют системы MAM (Mobile Application Management). Типовой функционал MAM-системы: адаптация интерфейса МП под конечного пользователя, управление аккаунтом пользователя, сбор статистики [15]. Применение MAM-системы позволяет упростить управление пользователями и конфигурацию МП. Системы MADM обеспечивают поддержку МП на этапе разработки, а MAM – на этапе эксплуатации.

Разработку МП можно вести с использованием платформ MADP (Mobile Application Development Platform), которые включают функции управления приложениями при их размещении на сервисе для скачивания пользователями, а также инструменты управления МП для их развертывания и защиты.

Большая часть средств поддержки ЖЦ МП, применяемых при их разработке, поддерживают только один этап ЖЦ, остальные этапы реализуются в других системах или программах, не связанных между собой. Существуют средства, которые поддерживают два этапа ЖЦ МП (реализация, эксплуатация): appery.io, TheAppBuilder, iBuildApp. Такие системы ориентированы на разработку МП бизнес-направленности, соответственно, поддержка МП на этапе эксплуатации в основном заключается в сборе статистики, управлении правами доступа пользователей и в выполнении других функций, предназначенных для администраторов МП.

Модель ЖЦ МП для пользователей с ID

Разработка МП для людей с ID сопряжена с высокими рисками, обусловленными невозможностью заранее предусмотреть, как такие пользователи будут реагировать на те или иные интерфейсные решения и смогут ли они правильно выбрать требуемые действия. Успешные интерфейсные решения, реализованные в МП и апробированные с пользователями с ID для разработки новых приложений, могут существенно повысить качество таких МП и уверенность разработчиков в их доступности. Необходимость учета особенностей пользователей с ID также может потребовать разработки специальных функций, не зависящих от назначения создаваемого МП. Повторное ис-

пользование кода этих функций позволит упростить процесс разработки и повысить скорость сборки проекта.

В зависимости от характера и проявления интеллектуальных ограничений потенциальные пользователи могут сталкиваться с различными проблемами, такими как, например, невозможность прочитать текст на экране МП, понять назначение элементов управления и других объектов интерфейса, выбрать и переместить объекты приложения и т.д. Чтобы обеспечить доступность для пользователей с различными ID, МП должно предоставлять возможность настройки экранов под конечного пользователя без потери полноценного доступа к функционалу. При этом необходимо предусмотреть возможность настройки интерфейса приложения опекающими людьми, которые хорошо знают возможности и ограничения своих подопечных и могут адаптировать мобильное приложение для конкретного пользователя.

Таким образом, модель ЖЦ МП для людей с ID с учетом особенностей такой категории пользователей должна включать этапы идентификации, проектирования, реализации, прототипирования с тестированием, развертывания, эксплуатации.

Этапы и процедуры модели MADLC показаны на рисунке 1.

Технологии разработки МП для людей с ID

Для реализации предложенной модели ЖЦ МП разработаны технологии и программные средства сквозной поддержки этапов проектирования, реализации и эксплуатации.

Технология разработки реализует принципы модуляризации [16] применительно к МП для пользователей с ID. Разработан шаблон многомодульной архитектуры, включающей три уровня модулей: модуль приложения (APP), функциональные модули (Feature), библиотечные модули (Library). Feature-модули соответствуют полноэкранным согласованным функциональным возможностям пользователя в приложении; APP-модуль связывает вместе функциональные модули; Library-модули предоставляют общий функционал, который повторно используется в нескольких или во всех функциях. Такой подход стандартизирует разработку МП для людей с ID, что положительно влияет на скорость и качество разработанных решений. Модули могут сохраняться в хранилище как компоненты для повторного использования в других МП. Разработка приложения с применением предложенной технологии позволяет разработчикам использовать готовые решения, успешно реализованные в предыдущих своих и чужих проектах, и добавлять но-

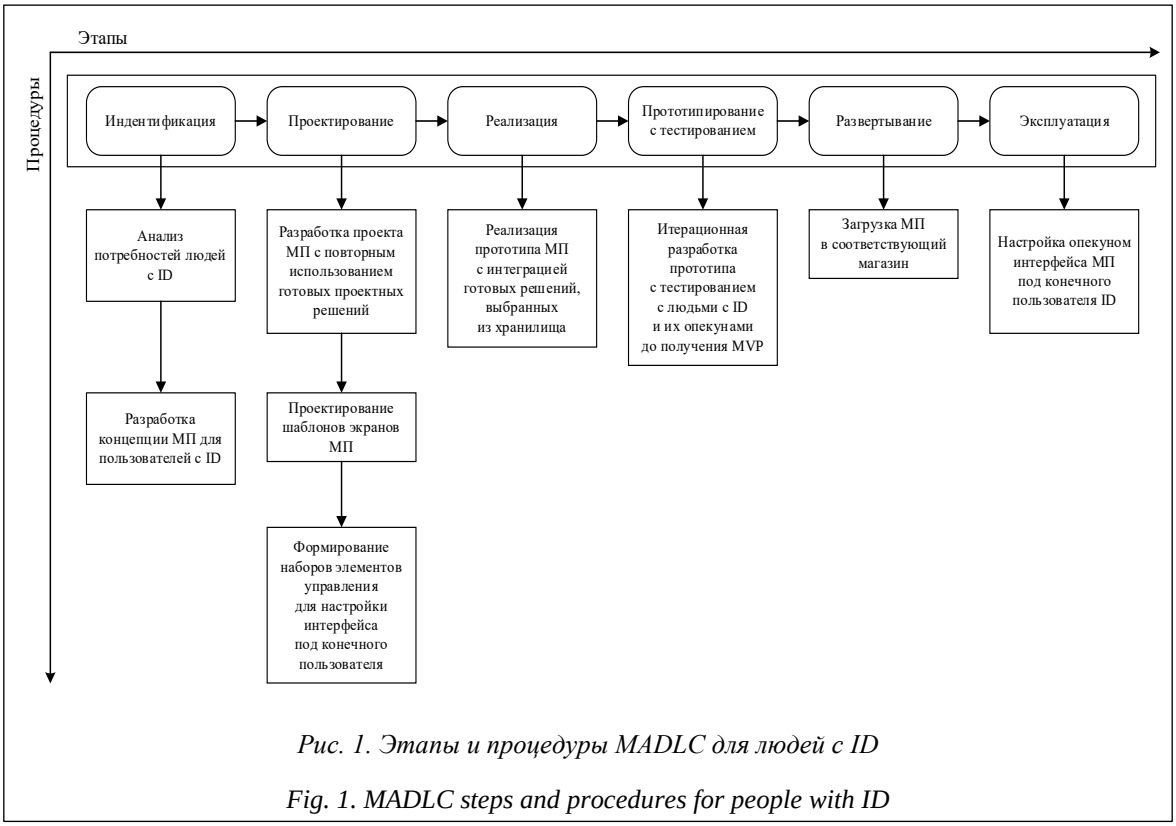


Рис. 1. Этапы и процедуры MADLC для людей с ID

Fig. 1. MADLC steps and procedures for people with ID

вые успешные решения в хранилище модулей. При этом разработчик применяет уже готовый шаблон архитектуры для генерации базового кода приложения.

Технология разработки адаптируемого интерфейса МП для людей с ID основана на концепции Pattern-Based. Процесс проектирования интерфейса заключается в разработке шаблонов всех экранов МП, на которых размещаются элементы интерфейса, требуемые для доступа к функционалу приложения. Технология адаптации интерфейса основана на разделении элементов шаблонов на обязательные и опциональные элементы, возможности выбора изображений элементов из предопределенного набора изображений (или добавления новых изображений) и прикреплении элементов по месту. Обязательными элементами являются те, которые необходимы для работы отдельно взятого приложения и, соответственно, должны включаться в интерфейс каждого пользователя МП. Опциональные элементы могут включаться или не включаться в интерфейс в зависимости от возможности пользователя работать с этими элементами. Каждому элементу управления интерфейса сопоставляется область на экране, в которой этот элемент может быть размещен (привязка по месту). При настройке интерфейса МП для каждого пользователя имеется возможность выбрать изображения всех элементов интерфейса из имеющихся в БД или отключить элемент интерфейса вовсе, если он относится к категории необязательных. Такую настройку интерфейса может осуществлять либо сам конечный пользователь с ID, либо его опекун, знающий возможности и ограничения своего подопечного.

Система управления

Для управления МП для людей с ID разработана система на основе CALS-технологии поддержки этапов проектирования, реализации и эксплуатации МП.

Архитектура системы включает подсистемы управления (MADM) и эксплуатации (MAM), БД, API для приложения, само приложение. Основной функционал MADM-подсистемы включает просмотр хранилища модулей, выбор и подключение одиночного модуля в проект разрабатываемого МП, добавление файла описания шаблонов страниц приложения, генерацию и редактирование конфигурационной панели для МП. После тестирования разработанного приложения модули, определенные разработчиком как применимые для повторного использования, могут добавляться в хранилище. Подсистема MAM реализует конфигурирование шаблонов экранов МП под конкретного пользователя с помощью конфигурационной модели, сгенерированной для этого МП.

Система построена на архитектурном стиле REST, который устанавливает однозначное соответствие между операциями, используемыми при работе с персистентными хранилищами данных, и HTTP-методами и обеспечивает таким образом независимость способа подключения МП к БД. Архитектурная схема в нотации ArchiMate представлена на рисунке 2.

Для конфигурирования интерфейса МП под конечного пользователя используется конфигурационная панель, разрабатываемая на этапе реализации МП и встраиваемая в систему управления для каждого МП. Конфигурационная панель включает область настроек, которая содержит виджеты для настройки элементов интерфейса конфигурируемого МП, и область, в которой отображается текущая страница интерфейса МП. Настройка интерфейса включает выбор элементов для отображения их изображений на экране МП (пиктограммы и/или другие графические иллюстрации). Процесс генерации панели строится на валидации, парсинге описания интерфейса МП и построении блока элементов управления и блока элементов представления на основе тегов и информации, содержащейся в них. Редактор, также являющийся частью разработанного модуля, позволяет предварительно сгенерировать конфигурационную панель на основе составленного описания и проверить файл описания на наличие ошибок. Данный способ адаптации интерфейса с помощью панели не требует изменения исходного кода МП и может осуществляться лицом, не владеющим навыками разработки МП.

Роли пользователей системы и их действия на этапах ЖЦ МП показаны в таблице.

Пример панели, разработанной для конфигурирования интерфейса МП для тестирования МП AIT, предназначенного для оценки профессиональных предпочтений пользователей с ID при выборе опекунами наиболее подходящей для них работы, приведен на рисунке 3.

Генерация интерфейсов МП в веб-приложении имеет следующие возможности: добавление приложения в систему (выполняется разработчиком МП) с названием и описанием приложения, а также с файлом, описывающим интерфейс; описание интерфейса с примене-

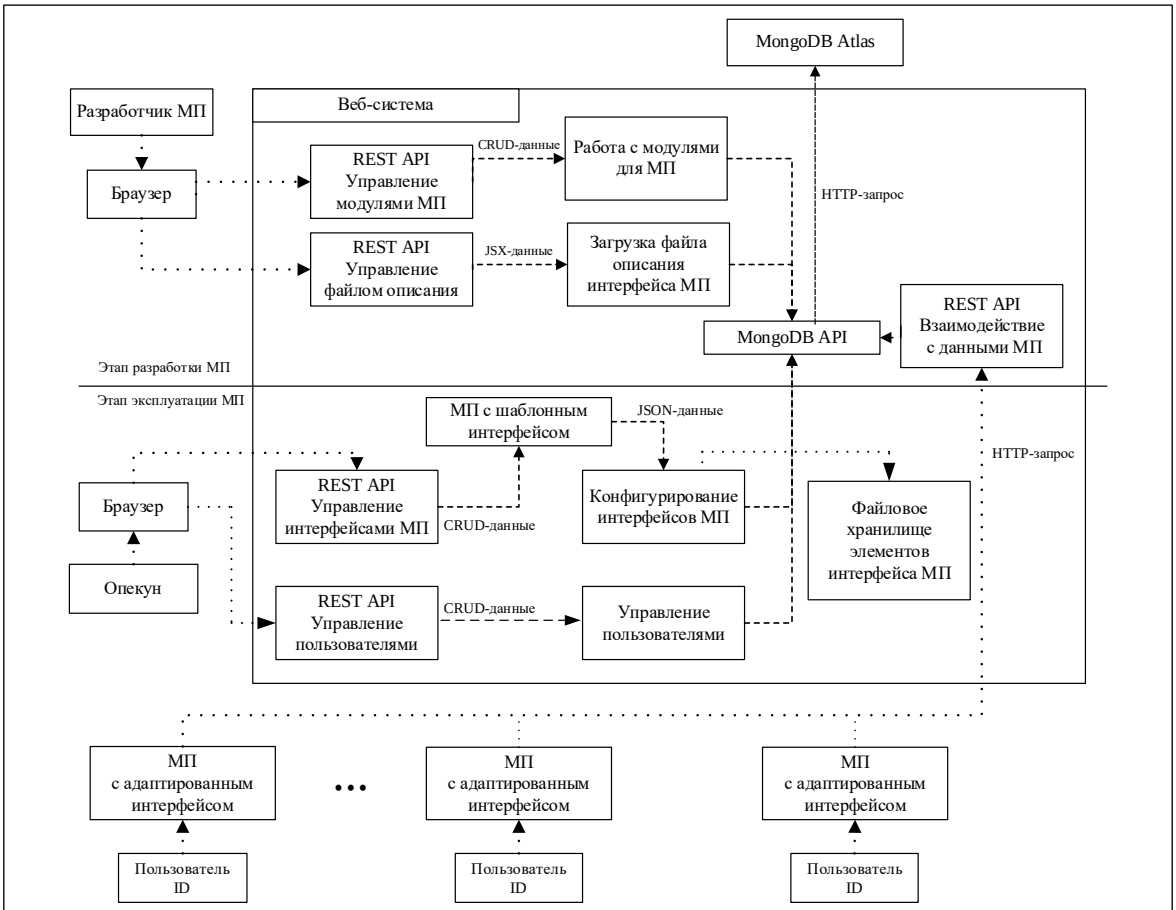


Рис. 2. Архитектурная схема в нотации ArchiMate

Fig. 2. Architectural diagram in ArchiMate notation

нием JSX; редактирование загруженного приложения; просмотр списка приложений в системе; удаление приложения.

Система реализована как веб-приложение. Для бэкенд-разработки выбрана платформа Node.js, для фронтенд-разработки применены HTML, CSS, JavaScript. В качестве БД использована MongoDB, для обмена данными между сервером и МП – протокол HTTP. Для поддержки разработки МП использованы специализированные наборы заранее разработанных компонентов и библиотек, реализованы поддержка наследования и ограничение доступа к определенным компонентам системы. Разработан интерфейс клиентской части, предназначенный для визуального управления процессом поддержки создания МП и адаптации его под конечного пользователя.

Система была апробирована на разработке МП Money Game для тренировки навыков совершения покупок для людей с ID [17]. Проект МП создан на основе многомодульной архитектуры. Для самостоятельной авторизации пользователей в проект включен разработанный в рамках другого проекта модуль авторизации пользователя с применением пикто-

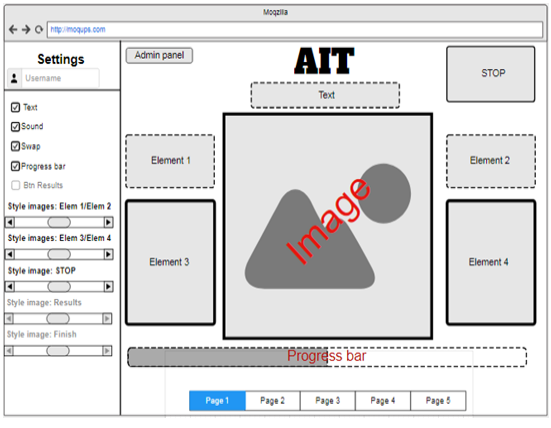


Рис. 3. Экранная форма с примером КП для конфигурирования интерфейса МП AIT для пользователей с ID

Fig. 3. Screenshot of an example of MA AIT configuration panel

Действия пользователей в системе управления МП

User activities in MA management system

Этап ЖЦ	Действия пользователя в системе	Роль
Проектирование	Разработка функционала МП. Разработка шаблонов интерфейса МП. Создание проекта МП. Поиск модулей в хранилище готовых решений, реализующих требуемые функции разрабатываемого МП. Подключение выбранных модулей в проект МП	Разработчик МП. Опекун
Реализация	Реализация проекта МП. Разработка конфигурационной панели под реализованное МП. Загрузка файла описания шаблонов интерфейса для генерации конфигурационной панели. Добавление модулей в хранилище готовых решений, применимых для повторного использования	Разработчик МП
Прототипирование с тестированием	Итерационная доработка интерфейса на основе информации о результатах тестирования МП на целевых группах пользователей с ID	Разработчик МП
Эксплуатация	Конфигурирование интерфейса МП под конечного пользователя с ID	Опекун

грамм, выбранный из хранилища моделей. Экранные формы МП и модуля авторизации показаны на рисунке 4.

Тестирование финального прототипа было организовано в MPI Oosterlo (https://www.mpi-oosterlo.be/), реабилитационном центре для людей с интеллектуальными ограничениями во Фландрии, Бельгия. Опекуном каждого пользователя с ID был сконфигурирован интерфейс, который он определил как наиболее соответствующий возможностям этого пользователя. Некоторые этапы процесса тестирования показаны на фотографиях (http://www.swsys.ru/uploaded/image/2023-4/2.jpg).

Тестирование финального прототипа на целевой группе показало, что МП в целом доступно таким пользователям, понятно и привлекательно как мобильная игра.

Заключение

В отличие от предложенных другими исследователями разработанная модель ЖЦ МП учитывает специфику пользователей с ID, обеспечивая возможность адаптации интерфейса под каждого в соответствии с его возможностями и ограничениями.

Для разработки МП для людей с ID на основе предложенной модели создана система управления приложением, реализующая CALS-технологию поддержки ЖЦ МП. Применение системы для разработки МП для пользователей с ID

Рис. 4. Экранные формы МП Money Game

Fig. 4. Money Game screenshots

позволяет существенно сократить временные затраты на разработку за счет использования унифицированной многомодульной архитектуры и включения в проект МП готовых модулей, а также генерации конфигурационной панели для адаптации интерфейса МП. Возможность применения шаблонов архитектур и готовых решений расширяет круг пользователей системы и делает ее доступной разработчикам, профессиональная деятельность которых связана с поддержкой людей с ограниченными интеллектуальными возможностями.

Привлечение разработчиков МП к созданию приложений для людей с ID с применением системы позволит пополнять хранилище модулей, доступных для подключения в новые проекты, и таким образом расширять возможности системы. В рамках дальнейших исследований предполагаются разработка метода мониторинга действий пользователей системы (разработчиков МП для людей с ID) и построение пользовательских шаблонов проектирования МП для последующей интеграции в систему.

Список литературы

1. Braun M., Wölfel M., Renner G., Menschik C. Accessibility of different natural user interfaces for people with intellectual disabilities. Proc. Int. Conf. on CW, 2020, pp. 211–218. doi: 10.1109/CW49994.2020.00041.

2. Zaina L.A.M., Fortes R.P.M., Casadei V. Preventing accessibility barriers: Guidelines for using user interface design patterns in mobile applications. J. of Systems and Software, 2022, vol. 186, art. 111213. doi: 10.1016/j.jss.2021.111213.

3. da Silva L.F., Parreira Junior P.A., Freire A.P. Mobile user interaction design patterns: A systematic mapping study. Information, 2022, vol. 13, no. 5, art. 236. doi: 10.3390/info13050236.

4. Shabalina O., Guriev V., Kosyakov S., Voronina A., Moffat D.C. Adaptable mobile software for supporting daily activities of people with intellectual disabilities. Proc. CIT&DS, 2019, pp. 474–484. doi: 10.1007/978-3-030-29750-3_37.

5. Perez C. Adaptable user interfaces for people with autism: A transportation example. Proc. W4A, 2018, no. 12, pp. 1–2. doi: 10.1145/3192714.3196318.

6. Kultsova M., Potseluico A., Zhukov I., Skorikov A., Romanenko R. A two-phase method of user interface adaptation for people with special needs. Proc. CIT&DS, 2017, pp. 805–821. doi: 10.1007/978-3-319-65551-2_58.

7. Heumader P., Miesenberger K., Murillo-Morales T. Adaptive user interfaces for people with cognitive disabilities within the easy reading framework. In: LNISA. Proc. ICCHP, 2020, vol. 12377, pp. 53–60. doi: 10.1007/978-3-030-58805-2_7.

8. Miraz D., Ali M., Excell P. Adaptive user interfaces and universal usability through plasticity of user interface design. Comput. Sci. Review, 2021, vol. 40, pp. 1–26. doi: 10.1016/j.cosrev.2021.100363.

9. Shokhista E., Kruglov A. Software development life cycle early phases and quality metrics: A systematic literature review. J. Phys.: Conf. Ser., 2020, vol. 1694, no. 1, art. 012007. doi: 10.1088/1742-6596/1694/1/012007.

10. Kyeremeh K. Overview of system development life cycle models. SSRN Electron. J., 2019. URL: https://ssrn.com/abstract=3448536 (дата обращения: 20.05.2023). doi: 10.2139/ssrn.3448536.

11. Жидков И.В., Зубарев И.В., Хабибуллин И.В. Выбор рациональной модели разработки безопасного программного обеспечения // Вопросы кибербезопасности. 2021. № 5. С. 21–29. doi: 10.21681/2311-3456-2021-5-21-29.

12. Nataraj R., Jagatheesan S.M. Lack of SDLC models and frameworks in mobile application development – a systematic literature review and study. J. of Xi'an University of Architecture & Tech., 2021, vol. XIII, no. 8, pp. 250–258.

13. Vagrani A., Singh N., Vigneswara P. Decline in mobile application life cycle. Procedia Comput. Sci., 2017, vol. 122, pp. 957–964. doi: 10.1016/j.procs.2017.11.460.

14. Kaur A., Kaur K. Suitability of existing software development life cycle (SDLC) in context of mobile application development life cycle (MADLC). IJCA, 2015, vol. 116, no. 19, pp. 1–6. doi: 10.5120/20441-2785.

15. Hayes D., Cappa F., Le-Khac N. An effective approach to mobile device management: Security and privacy issues associated with mobile applications. Digital Business, 2020, vol. 1, no. 1, art. 100001. doi: 10.1016/j.digbus.2020.100001.

16. Гурьев В.В., Косяков С.В., Сквазников Д.Е., Шабалина О.А. Web система адаптации интерфейсов мобильных приложений для людей с ограниченными интеллектуальными возможностями // Информационные технологии в науке, управлении, социальной сфере и медицине: сб. тр. 2018. Т. 2. С. 187–191.

17. Shabalina O., Voronina A., Davtian A., Peeters E., Hensbergen R. A mobile game for training shopping skills for people with intellectual disabilities. Proc. ECGBL, 2018, vol. 12, pp. 565–573.

Software & Systems doi: 10.15827/0236-235X.142.573-581 2023, vol. 36, no. 4, pp. 573–581

Developing a mobile application life cycle model for people with intellectual disabilities

Vladislav V. Guryev
Olga A. Shabalina
Natalya P. Sadovnikova
Angelina A. Voronina

For citation
Guryev, V.V., Shabalina, O.A., Sadovnikova, N.P., Voronina, A.A. (2023) 'Developing a mobile application life cycle model for people with intellectual disabilities', *Software & Systems*, 36(4), pp. 573–581 (in Russ.). doi: 10.15827/0236-235X.142.573-581
Article info
Received: 31.11.2022 After revision: 08.06.2023 Accepted: 13.06.2023

Abstract. The article discusses life cycle models and software tools for supporting mobile application development. It also analyzes peculiarities of developing mobile applications for people with intellectual disabilities. One way to increase the accessibility of applications is to make them adaptable. Such adaptation requires supporting an application dynamic content, therefore, these features must be already provided at life cycle design and implementation stages. The paper proposes a mobile application development lifecycle (MADLC) model that provides end-to-end support for the stages of design, implementation and maintenance of mobile applications for users with intellectual disabilities. There is a developed mobile application design technology based on a multi-module architecture, which includes developing ready-made architectural templates for generating a basic application code and reuse of successful design solutions. The paper describes the developed adaptable interface design technology based on Pattern-Based user Interface Design Approach. The technology includes developing screen templates, separating template elements into mandatory and optional ones, matching each GUI element with a predefined set of its possible visual representation, and attaching elements in place. A mobile application management system implements CALS technology of end-to-end support for the stages of mobile application design, implementation and operation. The paper proposes a method for adapting a mobile application interface using a configuration panel built into the system. There is an example of developing a mobile game for training shopping skills for people with intellectual disabilities. It is shown that the mobile application management system can significantly reduce time spent on developing mobile applications. Using architecture templates and ready-made solutions expands the range of system users and makes it available to developers whose professional activities are related to supporting people with intellectual disabilities.

Keywords: mobile application life cycle, mobile application life cycle model, MADLC model, user with intellectual disabilities, adaptable mobile application, CALS technology, adaptable interface, mobile application management system

References

1. Braun, M., Wölfel, M., Renner, G., Menschik, C. (2020) 'Accessibility of different natural user interfaces for people with intellectual disabilities', *Proc. Int. Conf. on CW*, pp. 211–218. doi: 10.1109/CW49994.2020.00041.
2. Zaina, L.A.M., Fortes, R.P.M., Casadei, V. (2022) 'Preventing accessibility barriers: Guidelines for using user interface design patterns in mobile applications', *J. of Systems and Software*, 186, art. 111213. doi: 10.1016/j.jss.2021.111213.
3. da Silva, L.F., Parreira Junior, P.A., Freire, A.P. (2022) 'Mobile user interaction design patterns: A systematic mapping study', *Information*, 13(5), art. 236. doi: 10.3390/info13050236.
4. Shabalina, O., Guriev, V., Kosyakov, S., Voronina, A., Moffat, D.C. (2019) 'Adaptable mobile software for supporting daily activities of people with intellectual disabilities', *Proc. CIT&DS*, pp. 474–484. doi: 10.1007/978-3-030-29750-3_37.
5. Perez, C. (2018) 'Adaptable user interfaces for people with autism: A transportation example', *Proc. W4A*, (12), pp. 1–2. doi: 10.1145/3192714.3196318.
6. Kultsova, M., Potseluico, A., Zhukov, I., Skorikov, A., Romanenko, R. (2017) 'A two-phase method of user interface adaptation for people with special needs', *Proc. CIT&DS*, pp. 805–821. doi: 10.1007/978-3-319-65551-2_58.
7. Heumader, P., Miesenberger, K., Murillo-Morales, T. (2020) 'Adaptive user interfaces for people with cognitive disabilities within the easy reading framework', in *LNISA. Proc. ICCHP*, 12377, pp. 53–60. doi: 10.1007/978-3-030-58805-2_7.
8. Miraz, D., Ali, M., Excell, P. (2021) 'Adaptive user interfaces and universal usability through plasticity of user interface design', *Comput. Sci. Review*, 40, pp. 1–26. doi: 10.1016/j.cosrev.2021.100363.
9. Shokhista, E., Kruglov, A. (2020) 'Software development life cycle early phases and quality metrics: A systematic literature review', *J. Phys.: Conf. Ser.*, 1694 (1), art. 012007. doi: 10.1088/1742-6596/1694/1/012007.
10. Kyeremeh, K. (2019) 'Overview of system development life cycle models', *SSRN Electron. J.*, available at: https://ssrn.com/abstract=3448536 (accessed May 20, 2023). doi: 10.2139/ssrn.3448536.
11. Zhidkov, I.V., Zubarev, I.V., Khabibullin, I.V. (2021) 'Choosing a rational model for development secure software', *Voprosy kiberbezopasnosti*, (5), pp. 21–29 (in Russ.). doi: 10.21681/2311-3456-2021-5-21-29.

12. Nataraj, R., Jagatheesan, S.M. (2021) 'Lack of SDLC models and frameworks in mobile application development – a systematic literature review and study', *J. of Xi'an University of Architecture & Tech.*, XIII (8), pp. 250–258.

13. Vagrani, A., Singh, N., Vigneswara, P. (2017) 'Decline in mobile application life cycle', *Procedia Comput. Sci.*, 122, pp. 957–964. doi: 10.1016/j.procs.2017.11.460.

14. Kaur, A., Kaur, K. (2015) 'Suitability of existing software development life cycle (SDLC) in context of mobile application development life cycle (MADLC)', *IJCA*, 116 (19), pp. 1–6. doi: 10.5120/20441-2785.

15. Hayes, D., Cappa, F., Le-Khac, N. (2020) 'An effective approach to mobile device management: Security and privacy issues associated with mobile applications', *Digital Business*, 1 (1), art. 100001. doi: 10.1016/j.dig-bus.2020.100001.

16. Guryev, V.V., Kosyakov, S.V., Skvaznikov, D.E., Shabalina, O.A. (2018) 'Web system for adapting interfaces of mobile applications for people with intellectual disabilities', *Proc. Inform. Tech. in Sci., Management, Social Sphere and Medicine*, 2, pp. 187–191 (in Russ.).

17. Shabalina, O., Voronina, A., Davtian, A., Peeters, E., Hensbergen, R. (2018) 'A mobile game for training shopping skills for people with intellectual disabilities', *Proc. ECGBL*, 12, pp. 565–573.

Авторы

Гурьев Владислав Витальевич [1], аспирант, vladgurjev@mail.ru
Шабалина Ольга Аркадьевна [1], к.т.н., доцент, o.a.shabalina@gmail.com
Садовникова Наталья Петровна [1], д.т.н., профессор, n_sadovnikova@vstu.ru
Воронина Ангелина Андреевна [1], аспирант, angelina.vaa@gmail.com

[1] Волгоградский государственный технический университет, г. Волгоград, 400005, Россия

Authors

Vladislav V. Guryev [1], Postgraduate Student, vladgurjev@mail.ru
Olga A. Shabalina [1], Cand. of Sci. (Engineering), Associate Professor, o.a.shabalina@gmail.com
Natalya P. Sadovnikova [1], Dr.Sc. (Engineering), Professor, n_sadovnikova@vstu.ru
Angelina A. Voronina [1], Postgraduate Student, angelina.vaa@gmail.com

[1] Volgograd State Technical University, Volgograd, 400005, Russian Federation

УДК 004.942 doi: 10.15827/0236-235X.142.582-589 2023. Т. 36. № 4. С. 582–589

Моделирование динамики гидроприводов в системах виртуального окружения

Е.В. Страшнов
И.Н. Мироненко

Ссылка для цитирования
Страшнов Е.В., Мироненко И.Н. Моделирование динамики гидроприводов в системах виртуального окружения // Программные продукты и системы. 2023. Т. 36. № 4. С. 582–589. doi: 10.15827/0236-235X.142.582-589
Информация о статье
Поступила в редакцию: 19.06.2023 После доработки: 07.07.2023 Принята к публикации: 10.07.2023

Аннотация. Системы виртуального окружения активно применяются во многих областях деятельности человека. В частности, широкое распространение получили виртуальные (компьютерные) тренажеры для моделирования и управления сложными механическими системами. Некоторые из таких систем оснащены гидроприводами в качестве исполнительных устройств. В данной работе рассматривается задача моделирования динамики гидроприводов в системах виртуального окружения в масштабе реального времени. Предлагаемое решение этой задачи базируется на подходе, когда осуществляется совместное моделирование динамики гидропривода и системы шарнирно связанных тел. Для этого задействован метод последовательных импульсов, в рамках которого получены ограничения относительно координат и скоростей поршня и двух связанных тел. Идея такого подхода состоит в том, что сначала путем интегрирования уравнений гидравлики с помощью полунеявной схемы Эйлера вычисляется скорость движения поршня без учета нагрузки, а затем эта скорость корректируется с помощью импульса, который обеспечивает ограничения скоростей поршня и двух связанных тел. Предлагаемый подход является итерационным и позволяет реализовать динамику систем шарнирно связанных тел с гидроприводом в масштабе реального времени. Основной результат заключается в том, что разработанный подход является универсальным и позволяет обрабатывать все ограничения в системе тел единым образом. В отличие от других аналогичных методов он не требует громоздких матричных вычислений и решения систем линейных уравнений большой размерности. Апробация в созданном программном комплексе VirSim показала адекватность и эффективность предложенных решений. Практическая значимость работы заключается в том, что предложенные методы и подходы для моделирования динамики гидроприводов могут быть использованы при реализации тренажеров, предназначенных для обучения операторов навыкам управления роботами, машинами различного типа, механизмами, техникой и т.д.
Ключевые слова: моделирование динамики, гидропривод, гидрораспределитель, метод последовательных импульсов, полунеявная схема Эйлера, система виртуального окружения
Благодарности. Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2022-0012 «Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации»

Введение. Гидравлический привод (гидропривод) представляет собой совокупность устройств для приведения в движение рабочих органов машин и механизмов посредством энергии жидкости. В настоящей работе рассматриваются гидроприводы поступательного движения (гидроцилиндры), состоящие из емкости, внутри которой находится поршень со штоком. Гидроцилиндры используются в строительно-дорожных, землеройных, подъемно-транспортных машинах, авиации, робототехнике и т.д.

Развитие электроники обусловило появление бульдозеров, экскаваторов, ковшовых погрузчиков, кранов и проч., снабженных управлением по принципу джойстика. Это повлекло за собой некоторые проблемы, одна из них – обучение операторов навыкам управления такими машинами. Использование реальных машин для обучения операторов сопряжено с материальными затратами и может привести к поломке дорогостоящего оборудования. Альтернативное решение проблемы заключается в применении систем виртуального окружения. Такой подход подразумевает замену реальных объектов на их виртуальные аналоги с реализацией моделирования этих объектов в виртуальной среде. Для этого активно создаются компьютерные тренажеры и симуляторы [1], на которых обучение операторов осуществляется с помощью органов управления, аналогичных органам управления реальных машин. При этом качество обучения непосредственно зависит от адекватности моделирования динамики виртуальных объектов. В процессе обучения у операторов не должны возникнуть ложные навыки управления рабочими органами машин, которые содержат гидроприводы. Кроме того, системы виртуального окружения накладывают ограничения, в рамках которых все расчеты должны проводиться в масштабе реального времени. Поэтому разработка эффективных методов и алгоритмов моделирования динамики гидроприводов в системах виртуального

окружения является важной и актуальной задачей.

Среди существующих решений для моделирования динамики гидроприводов условно можно выделить два подхода. Первый подход [2, 3] подразумевает разработку сложной математической модели гидропривода без учета динамики всей механической системы. Динамические процессы в гидросистеме описываются системой нелинейных дифференциальных уравнений, которые формируются в зависимости от ее структуры. Распространенным решением в рамках данного подхода является использование библиотеки гидравлики Simscape SimHydraulics (https://www.mathworks.com/products/simscape-fluids.html) среды Simulink, предназначенной для моделирования и симуляции гидравлических систем. В работах [4–6] из блоков данной библиотеки построены модели, позволяющие осуществлять имитационное моделирование гидроприводов различной сложности. Второй подход предполагает совместное моделирование гидропривода с динамикой системы многих тел [7–10]. Идея такого подхода заключается в том, что на каждом шаге моделирования динамика систем многих тел сводится к дифференциально-алгебраическим уравнениям большой размерности. Один из методов решения этих уравнений базируется на итерационном алгоритме с применением множителей Лагранжа. Недостаток этого подхода состоит в том, что из-за большого числа матричных вычислений и необходимости решения систем линейных уравнений обеспечение реального времени моделирования динамики систем тел становится затруднительным.

В настоящей работе для реализации динамики гидропривода предлагается подход, основанный на итерационном методе последовательных импульсов. Его основная идея заключается в том, что вводится специальное ограничение, связывающее координаты поршня и двух тел механической системы, относительным движением которых управляет гидропривод. В рамках такого подхода на каждом шаге моделирования сначала реализуется динамика гидропривода без нагрузки, а потом эта нагрузка учитывается при обеспечении введенного ограничения. Апробация предлагаемых в статье решений была проведена в разработанном в НИИСИ РАН программном комплексе виртуального окружения VirSim [11] на примере моделирования динамики роботов, которые содержат гидроприводы.

Математическая модель гидропривода

Рассмотрим гидропривод, который имеет схему, показанную на рисунке 1. В этой схеме *гидроцилиндр* (ГЦ) двустороннего действия перемещает груз массой m, к которому приложена внешняя сила F_l. Положение поршня гидроцилиндра задается координатой x_p, для которой заданы ограничения $x_p^{\min} \le x_p \le x_p^{\max}$, где $x_p^{\min} \ge 0$. Управление движением поршня гидроцилиндра осуществляется с помощью золотникового *гидравлического распределителя* (ГР). В рассматриваемой схеме в нейтральном положении золотника распределителя обе полости гидроцилиндра будут соединены со сливом в гидравлический *бак* (Б). Перемещение золотника x_v осуществляется посредством *сервоклапана* (СК) с электрическим управлением, где входным сигналом является напряжение u. При смещении золотника одна полость гидроцилиндра будет связана с линией высокого давления, создаваемого *насосом* (Н), а другая со сливом. Подача жидкости в полости гидроцилиндра по гидролиниям приводит к тому, что будут возникать давления p_A и p_B, вызывая движение поршня. При этом в качестве допущения принимается, что давление насоса p_S и давление гидробака p_T в процессе моделирования постоянны. В рассматриваемой схеме расходы жидкости Q_A и Q_B для удобства направлены внутрь гидроцилиндра. Это означает, что в процессе движения поршня они будут иметь разные знаки.

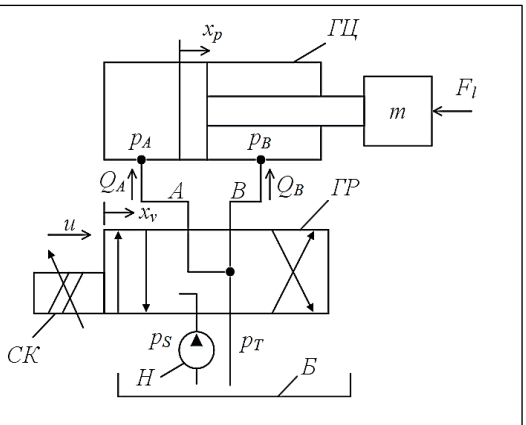


Рис. 1. Схема гидропривода

Fig. 1. Hydraulic actuator scheme

Согласно второму закону Ньютона, уравнение движения поршня будет иметь вид

$$m\dot{v}_p = p_A S_A - p_B S_B - k v_p + F_f + F_l, \qquad (1)$$

где v_p – скорость движения поршня; S_A и S_B – площади поршневой и штоковой полостей гид-

роцилиндра; k – коэффициент вязкого трения; F_f – сила сухого трения, модуль которой не превышает $F_f^{\max}$.

Изменение давления в полостях гидроцилиндра подчиняется уравнениям баланса расхода жидкости:

$$\dot{p}_A = \frac{\beta}{V_A + x_p S_A}(Q_A - v_p S_A),$$
$$\dot{p}_B = \frac{\beta}{V_B - x_p S_B}(Q_B + v_p S_B), \quad (2)$$

где β – модуль объемной упругости жидкости; V_A и V_B – начальные объемы полостей гидроцилиндра.

При смещении золотника относительно нейтрального положения изменяются площади отверстий, через которые жидкость поступает в полости гидроцилиндра. Расход жидкости через отверстие определяется известным из гидравлики соотношением [12]:

$$Q(x_v, \Delta p) = c_v x_v \operatorname{sgn}(\Delta p)\sqrt{|\Delta p|},$$

где $c_v = \pi d_v \alpha_d \sqrt{\frac{2}{\rho}}$ – коэффициент расхода жидкости; x_v – положение золотника; Δp – перепад давления на концах плунжера золотника; d_v – диаметр золотника; α_d – коэффициент расхода канала в золотнике; ρ – плотность жидкости.

В данной статье рассматривается модель золотника с отрицательным перекрытием на слив в нейтральном положении. Также задано положительное перекрытие для подачи жидкости от насоса. Пусть положение золотника является нормализованным ($-1 \le x_v \le 1$). Тогда перекрытия определяются с помощью следующих функций:

$$f_{SA} = \begin{cases} x_v - c_1, & x_v \ge c_1 \\ 0, & x_v < c_1 \end{cases}, \quad f_{SB} = \begin{cases} -x_v + c_1, & x_v \le -c_1 \\ 0, & x_v > -c_1 \end{cases},$$
$$f_{TA} = \begin{cases} -x_v + c_2, & x_v \le c_2 \\ 0, & x_v > c_2 \end{cases}, \quad f_{TB} = \begin{cases} x_v + c_2, & x_v \ge -c_2 \\ 0, & x_v < -c_2 \end{cases},$$

в которых c_1 и c_2 – величины положительного и отрицательного перекрытий соответственно.

Расходы жидкости Q_A и Q_B через отверстия золотника будут вычисляться как

$$Q_A = c_v f_{SA}(x_v)\operatorname{sgn}(p_S - p_A)\sqrt{|p_S - p_A|} + \\ + c_v f_{TA}(x_v)\operatorname{sgn}(p_T - p_A)\sqrt{|p_T - p_A|}, \quad (3)$$

$$Q_B = c_v f_{SB}(x_v)\operatorname{sgn}(p_S - p_B)\sqrt{|p_S - p_B|} + \\ + c_v f_{TB}(x_v)\operatorname{sgn}(p_T - p_B)\sqrt{|p_T - p_B|}. \quad (4)$$

Если пренебречь гистерезисом и чувствительностью срабатывания клапана гидравлического распределителя, то его динамику можно описать с помощью дифференциального уравнения второго порядка:

$$\ddot{x}_v + 2D_v\omega_v\dot{x}_v + \omega_v^2 x_v = K_v\omega_v^2 u, \quad (5)$$

где K_v – коэффициент усиления электрического сигнала клапана; ω_v – собственная частота; D_v – коэффициент демпфирования [2].

Дифференциальные уравнения (1)–(5) описывают математическую модель динамики гидропривода. Предлагаемое решение для моделирования динамики гидропривода заключается в том, что сначала вычисляется скорость поршня v_p без учета внешней силы F_l и силы сухого трения F_f, а затем полученная скорость корректируется на основе динамики всей системы тел.

Динамика шарнирно связанных тел

Гидроприводы применяются для управления механическими устройствами, которые представляют собой систему шарнирно связанных тел. Для моделирования динамики таких систем ранее был разработан итерационный метод последовательных импульсов [13, 14], который обеспечивает выполнение ограничений различного типа, накладываемых на координаты и скорости тел. Примером служат голономные ограничения шарниров, контакта, удара и трения тел, которые задаются в виде равенств и неравенств. Рассмотрим ограничение специального типа для реализации динамики гидропривода.

На рисунке 2 изображено поршневое соединение двух тел (с номерами i и j) механической системы, относительное движение в котором управляется с помощью гидропривода. Точки C_i и C_j являются центрами масс двух тел. Движение в шарнире задается с помощью точек P_i и P_j, а также единичной осью $\mathbf{n}_i$, которые жестко связаны с соответствующими телами.

Для учета динамики гидропривода рассмотрим ограничение

$$G = \mathbf{P}_i\mathbf{P}_j\mathbf{n}_i - x_p = 0. \quad (6)$$

Продифференцируем его по времени:

$$\dot{G} = (\mathbf{v}_j + \boldsymbol{\omega}_j \times \mathbf{r}_j - \mathbf{v}_i - \boldsymbol{\omega}_i \times \mathbf{r}_i)\mathbf{n}_i + \\ + \mathbf{P}_i\mathbf{P}_j\frac{d\mathbf{n}_i}{dt} - v_p = 0, \quad (7)$$

где $\mathbf{v}_i$ и $\mathbf{v}_j$ – линейные скорости двух тел; $\boldsymbol{\omega}_i$ и $\boldsymbol{\omega}_j$ – угловые скорости двух тел; $\mathbf{r}_i = \mathbf{C}_i\mathbf{P}_i$, $\mathbf{r}_j = \mathbf{C}_j\mathbf{P}_j$.

Так как векторы $\mathbf{P}_i\mathbf{P}_j$ и $\mathbf{n}_i$ параллельны, применяя формулу Бура [15] $\dot{\mathbf{n}}_i = \boldsymbol{\omega}_i \times \mathbf{n}_i$ и свойство смешанного произведения $\mathbf{a}(\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{c} \times \mathbf{a}) = \mathbf{c}(\mathbf{a} \times \mathbf{b})$, получим следующее:

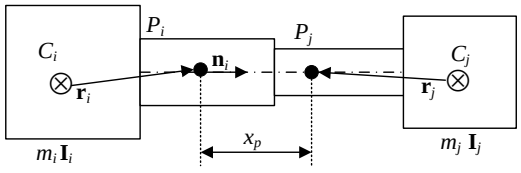


Рис. 2. Скользящее (поршневое) соединение двух тел

Fig. 2. Slider (piston) joint of two bodies

$$\mathbf{P}_i\mathbf{P}_j\frac{d\mathbf{n}_i}{dt}=\mathbf{P}_i\mathbf{P}_j(\boldsymbol{\omega}_i\times\mathbf{n}_i)=\boldsymbol{\omega}_i(\mathbf{n}_i\times\mathbf{P}_i\mathbf{P}_j)=0.$$

Тогда (7) преобразуется к виду

$$\dot{G}=(\mathbf{v}_j-\mathbf{v}_i)\mathbf{n}_i+\boldsymbol{\omega}_j\mathbf{t}_j-\boldsymbol{\omega}_i\mathbf{t}_i-v_p=0, \qquad (8)$$

где $\mathbf{t}_i=\mathbf{r}_i\times\mathbf{n}_i$, $\mathbf{t}_j=\mathbf{r}_j\times\mathbf{n}_i$.

Задача заключается в том, чтобы обеспечить ограничения вида (8). Для ее решения воспользуемся методом последовательных импульсов. Сформируем импульс $\mathbf{p}_n=p\mathbf{n}_i$, при применении которого скорости тел и поршня будут изменяться следующим образом:

$$\mathbf{v}_i'=\mathbf{v}_i-m_i^{-1}p\mathbf{n}_i,$$
$$\boldsymbol{\omega}_i'=\boldsymbol{\omega}_i+\mathbf{I}_i^{-1}\mathbf{r}_i\times(-p\mathbf{n}_i)=$$
$$=\boldsymbol{\omega}_i-p\mathbf{I}_i^{-1}\mathbf{r}_i\times\mathbf{n}_i=\boldsymbol{\omega}_i-p\mathbf{I}_i^{-1}\mathbf{t}_i,$$
$$\mathbf{v}_j'=\mathbf{v}_j+m_j^{-1}p\mathbf{n}_i,$$
$$\boldsymbol{\omega}_j'=\boldsymbol{\omega}_j+\mathbf{I}_j^{-1}\mathbf{r}_j\times p\mathbf{n}_i=$$
$$=\boldsymbol{\omega}_j+p\mathbf{I}_j^{-1}\mathbf{r}_j\times\mathbf{n}_i=\boldsymbol{\omega}_j+p\mathbf{I}_j^{-1}\mathbf{t}_j,$$
$$v_p'=v_p-m^{-1}p,$$

где m – приведенная масса поршня; m_i и m_j – массы тел; $\mathbf{I}_i$ и $\mathbf{I}_j$ – тензоры инерции тел.

Подставим полученные скорости в уравнение (8). Тогда после преобразований имеем уравнение

$$(m^{-1}+m_i^{-1}+m_j^{-1}+\mathbf{t}_i^{\mathrm{T}}\mathbf{I}_i^{-1}\mathbf{t}_i+\mathbf{t}_j^{\mathrm{T}}\mathbf{I}_j^{-1}\mathbf{t}_j)p=$$
$$=(\mathbf{v}_i-\mathbf{v}_j)\mathbf{n}_i+\boldsymbol{\omega}_i\mathbf{t}_i-\boldsymbol{\omega}_j\mathbf{t}_j+v_p. \qquad (9)$$

Решая его, найдем импульс p, применение которого позволит выполнить ограничение вида (8). Однако в методе последовательных импульсов также требуется обеспечить выполнение остальных ограничений. Поэтому ограничение (8) будет выполнено только с заданной точностью после проведения нескольких итераций. Сходимость метода последовательных импульсов обусловлена его эквивалентностью сходящемуся методу Гаусса–Зейделя для численного решения системы линейных уравнений, которая образуется при одновременном учете всех ограничений системы тел.

В процессе моделирования также будут нарушаться ограничения (6). Для обеспечения их выполнения применяется стабилизация на основе ранее разработанного авторами метода раздельных импульсов.

Результаты моделирования

Предложенные в статье методы и подходы для моделирования динамики гидроприводов были реализованы в системе виртуального окружения VirSim [11], разработанной в НИИСИ РАН. Программный комплекс написан на языке C++ с использованием графической библиотеки OpenGL и программно-аппаратной архитектуры CUDA, в него входят подсистемы управления, динамики и визуализации. В рассматриваемом комплексе логика управления виртуальными объектами задается с помощью функциональных схем, которые состоят из набора связанных друг с другом блоков. Согласно созданной функциональной схеме, подсистема управления осуществляет расчет управляющих сигналов, подаваемых на исполнительные устройства виртуальных объектов. Подсистема динамики обрабатывает эти сигналы и вычисляет новые координаты (положения и ориентации) объектов на основе математических моделей, описывающих их движение. Полученные координаты передаются в подсистему визуализации, которая выполняет рендеринг изображений виртуальной сцены. Масштаб реального времени достигается за счет использования аппаратно-программных средств для параллельных вычислений на GPU, включая шейдеры (вершинный, тесселяционный, геометрический и фрагментный).

Математическая модель гидропривода (1)–(5) и обработка ограничений вида (6) и (8) методом последовательных импульсов реализованы в виде программных модулей подсистемы динамики программного комплекса VirSim. Для интегрирования уравнений гидравлики (1), (2) и (5) была задействована полунеявная схема Эйлера. Так как модуль объемной упругости β является достаточно большим, уравнения (2) образуют жесткую систему дифференциальных уравнений. По этой причине интегрирование уравнений гидравлики реализовано с уменьшенным фиксированным шагом $\Delta t_1 = 0.1$ мс, в то время как интегрирование уравнений движения тел осуществлялось с переменным шагом Δt (от 1 до 10 мс), что необходимо для обеспечения реального времени моделирования.

Согласно полунеявной схеме Эйлера, интегрирование уравнений (1), (2) и (5) выполняется следующим образом:

$$\dot{x}_v(t+\Delta t_1) = \dot{x}_v(t) + \Delta t_1(K_v\omega_v^2 u(t) - 2D_v\omega_v\dot{x}_v(t) - \omega_v^2 x_v(t)),$$

$$x_v(t+\Delta t_1) = x_v(t) + \Delta t_1\dot{x}_v(t+\Delta t_1),$$

$$p_A(t+\Delta t_1) = p_A(t) + \frac{\beta\Delta t_1}{V_A + x_p(t)S_A}(Q_A(t+\Delta t_1) - v_p(t)S_A),$$

$$p_B(t+\Delta t_1) = p_B(t) + \frac{\beta\Delta t_1}{V_B - x_p(t)S_B}(Q_B(t+\Delta t_1) + v_p(t)S_B),$$

$$v_p(t+\Delta t_1) = v_p(t) + m^{-1}\Delta t_1(p_A(t+\Delta t_1)S_A - p_B(t+\Delta t_1)S_B - kv_p(t)),$$

где $Q_A(t+\Delta t_1)$ и $Q_B(t+\Delta t_1)$ вычисляются по формулам (3) и (4) соответственно.

В рассматриваемой схеме интегрирования на каждом шаге моделирования по входному сигналу $u(t)$ сначала вычисляется новое положение $x_v(t+\Delta t_1)$ клапана гидрораспределителя, затем давления $p_A(t+\Delta t_1)$ и $p_B(t+\Delta t_1)$, а в конце скорость движения поршня $v_p(t+\Delta t_1)$. Интегрирование продолжается до тех пор, пока не будут получены значения скоростей поршня для момента времени $t+\Delta t$. Эти скорости затем используются при обработке ограничений (6) и (8) с итерационным вычислением импульса p по формуле (9), а также для учета сухого трения и ограничений на относительное движение поршня. В качестве критерия окончания итераций было использовано условие $|p| \le \varepsilon$, где ε – заданное число.

Апробация разработанных в статье методов и подходов выполнялась на персональном компьютере с процессором Intel Core i7-6800K (3,4 ГГц, 6 ядер) и видеокартой NVIDIA GeForce GTX 1080 Ti (GPU содержит 3 584 ядра) на примере подъема груза массой 100 кг с помощью гидропривода. Параметры моделирования гидропривода приведены в таблице. Для апробации была создана функциональная схема, которая показана на рисунке 3. В этой схеме начиная с момента времени t = 1 000 мс по таймеру подается максимальный управляющий сигнал u = 1. Гидропривод в схеме реализован посредством блока, на вход которого поступают сигналы напряжения клапана гидрораспределителя и тормозной муфты, а на выходе выдается текущее положение поршня. На рисунке 4 приведены графики изменения давле-

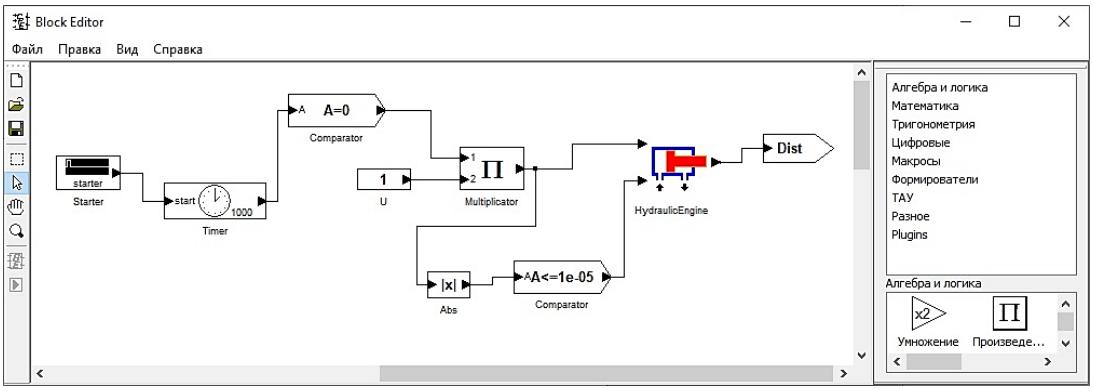
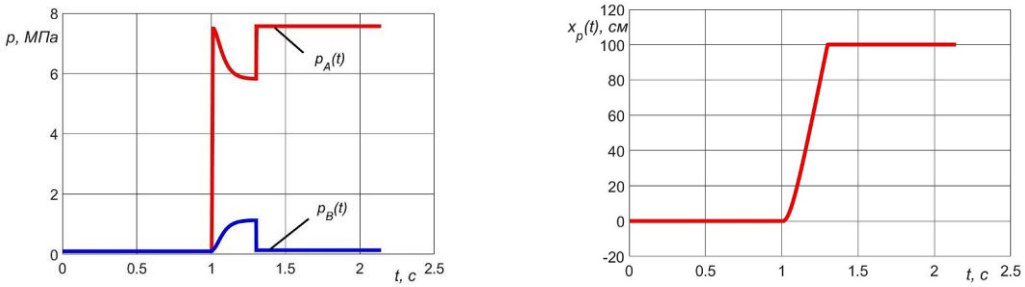


Рис. 3. Функциональная схема управления

Fig. 3. Functional control scheme



а) *б)*

Рис. 4. Зависимость от времени: а) давлений; б) положения поршня

Fig. 4. Dependence of pressures a) and piston position б) on time

Параметры моделирования

Simulation parameters

Параметр	Значение	Параметр	Значение	Параметр	Значение
m	10 кг	$x_p^{\min}$	0 см	d_v	0.5 см
ρ	800 кг/м^3	$x_p^{\max}$	100 см	c_1	0.05
β	1000 МПа	$S_A = S_B$	1.18*10^{-3} м^2	c_2	0.2
k	1000 кг/с	$V_A = V_B$	3*10^{-4} м^3	K_v	0.00833
$F_f^{\max}$	300 Н	p_T	0.1 МПа	ω_v	219 рад./с
α_d	0.61	p_S	7.6 МПа	D_v	0.9

ний в полостях гидроцилиндра и положения поршня от времени. Из рисунков видно, что при подаче напряжения давление в поршневой полости резко подскакивает, а при движении поршня немного убывает, в то время как в штоковой полости возрастает. В установившемся состоянии при достижении максимального положения поршня давление в поршневой полости выравнивается с давлением насоса, а давление в штоковой полости равно давлению гидробака. Полученные результаты показывают адекватность предложенных в статье методов и подходов для моделирования динамики гидропривода с требуемой точностью.

Предложенные в статье решения также были реализованы для моделирования динамики роботов, которые содержат гидроприводы. Примером служит виртуальная модель робота РТК-М (рис. 5) на шасси с четырьмя гусеницами. Этот робот оснащен манипулятором, кабелеукладчиком и опорами, управление которыми осуществляется с помощью гидроприводов. Основное предназначение робота – это выполнение аварийных демонтажных работ, во время которых опоры увеличивают устойчивость робота, не позволяя ему опрокинуться.

Рис. 5. Виртуальная модель робота РТК-М с опорами

Fig. 5. Virtual model of RTK-M robot with supports

Заключение

Разработаны методы и подходы для моделирования в реальном времени динамики гидроприводов в системах виртуального окружения. Предложенные решения основаны на математической модели гидропривода и методе последовательных импульсов. Получены ограничения, которые позволяют моделировать динамику систем тел с гидроприводами единым универсальным образом. Преимущество разработанных методов и подходов перед другими аналогичными решениями в том, что не требуются громоздкие матричные вычисления и решение больших систем линейных уравнений.

Апробация в программном комплексе системы виртуального окружения показала применимость разработанных методов и подходов для управления роботами, содержащими гидроприводы. Полученные результаты могут быть использованы при создании тренажеров, предназначенных для обучения операторов навыкам управления машинами различного типа, такими как краны, экскаваторы, бульдозеры, погрузчики и проч.

В перспективе разработанные методы и подходы могут быть применены для моделирования гидроприводов вращательного типа (гидромоторы и поворотные гидродвигатели), а также более сложных гидравлических систем. Кроме того, одним из возможных направлений исследований может быть реализация математической модели насоса с регулятором давления.

Список литературы

1. Ляшенко Ю.М., Ляшенко А.Ю., Ревякина Е.А. Применение компьютерных тренажеров в образовательном процессе машинистов одноковшовых экскаваторов как инструмент повышения уровня показателей функционального статуса // Современные прикладные исследования: мат. науч.-практич. конф. 2021. Т. 1. С. 236–242.
2. Wang W., Zhao J. Energy-efficient robust control for direct drive and energy recuperation hydraulic servo system. Complexity, 2020, pp. 1–19. doi: 10.1155/2020/6959273.
3. Жданов А.В. Математическая модель позиционного гидропривода // Вестн. СибАДИ. 2016. № 3(49). С. 87–93.
4. Михеева Н.И. Имитационное моделирование гидропривода в нелинейном приближении // Вестн. КРСУ. 2017. Т. 17. № 5. С. 83–86.
5. Debeleac C. Dynamic modeling and simulation of working regime of the hydraulic driven of auger bucket for loader using Matlab/SimHydraulics. Hidraulica, 2021, no. 4, pp. 7–16.
6. Полтева Е.А., Иванов Д.В., Сандлер И.Л., Антонова В.В. Моделирование информационно-измерительной системы гидравлического привода промышленного робота модели «Универсал 15» // Вестн. СамГУПС. 2020. № 2. С. 74–82.
7. Rahikainen J., Kiani-Oshtorjani M., Sopanen J., Jalali P., Mikkola A. Computationally efficient approach for simulation of multibody and hydraulic dynamics. Mechanism and Machine Theory, 2018, vol. 130, pp. 435–446. doi: 10.1016/j.mechmachtheory.2018.08.023.
8. Mitrev R., Janosevic D., Marinkovic D. Dynamical modelling of hydraulic excavator considered as a multibody system. Tech. Gazette, 2017, vol. 24, no. 2, pp. 327–338. doi: 10.17559/TV-20151215150306.
9. Rahikainen J., Gonzalez F., Naya M.A., Sopanen J., Mikkola A. On the cosimulation of multibody systems and hydraulic dynamics. Multibody System Dynamics, 2020, vol. 50, no. 2, pp. 143–167. doi: 10.1007/s11044-020-09727-z.
10. Kurinov I., Orzechowski G., Hamalainen P., Mikkola A. Automated excavator based on reinforcement learning and multibody system dynamics. IEEE Access, 2020, vol. 8, pp. 213998–214006. doi: 10.1109/ACCESS.2020.3040246.
11. Михайлюк М.В., Мальцев А.В., Тимохин П.Ю., Страшнов Е.В., Крючков Б.И., Усов В.М. Система виртуального окружения VirSim для имитационно-тренажерных комплексов подготовки космонавтов // Пилотируемые полеты в космос. 2020. № 4. С. 72–95. doi: 10.34131/MSF.20.4.72-95.
12. Manring N., Fales R. Hydraulic Control Systems. USA, NY, JohnWiley Publ., 2019, 389 p. doi: 10.1002/9781119418528.
13. Михайлюк М.В., Страшнов Е.В. Моделирование динамики системы связанных тел с учетом трения в шарнирах // Наука и Образование. МГТУ им. Н.Э. Баумана. 2016. № 1. С. 108–124. doi: 10.7463/0116.0830582.
14. Страшнов Е.В., Омельченко Д.В. Моделирование динамики антропоморфных шагающих роботов в системах виртуального окружения // INJOIT. 2020. Т. 8. № 10. С. 66–72.
15. Дронг В.И., Дубинин В.В., Ильин М.М. и др. Курс теоретической механики. М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. 580 с.

Software & Systems doi: 10.15827/0236-235X.142.582-589 2023, vol. 36, no. 4, pp. 582–589

Simulation of hydraulic actuator dynamics in virtual environment systems

Evgeny V. Strashnov
Irina N. Mironenko

For citation
Strashnov, E.V., Mironenko, I.N. (2023) ‘Simulation of hydraulic actuator dynamics in virtual environment systems’, *Software & Systems*, 36(4), pp. 582–589 (in Russ.). doi: 10.15827/0236-235X.142.582-589
Article info
Received: 19.06.2023 After revision: 07.07.2023 Accepted: 10.07.2023

Abstract. The development of computer graphics and computing machinery has led to active using of virtual environment systems in many areas of human activity. In particular, virtual (computer) training complexes for simulation and control of complex mechanical systems are widely used. Some of these systems are equipped with hydraulic actuators. This paper considers real-time simulation of hydraulic actuators dynamics in virtual environment systems. The authors propose solving this problem based on an approach with co-simulation dynamics of hydraulic actuators and articulated rigid bodies. To this end, the authors used a sequential impulses method that resulted in obtaining coordinates and velocities of piston and two linked bodies. The approach assumes that, first, piston velocity is computed by integrating hydraulic equations using a semi-implicit Euler scheme without taking load into account. Then, the obtained velocity is adjusted by an impulse that limits velocities of a piston and two linked bodies. The proposed approach is iterative and makes it possible to implement dynamics of articulated body systems with a hydraulic actuator in real time. The main result of the study is in flexibility of

the developed approach for simulating hydraulic actuator dynamics based on a sequential impulses method; it allows ensuring all constraints in a multibody system in a unified way. Unlike other similar solutions, this approach does not require cumbersome matrix calculations and solving large dimension linear equation systems. Approbation in the created VirSim software package showed adequacy and effectiveness of the solutions proposed in the paper. The practical significance of the work is that the proposed methods and approaches for simulating hydraulic actuators dynamics can be used in simulators for training operators of skills for controlling robots, machines of various types, mechanisms, equipment, etc.

Keywords: simulation of dynamics, hydraulic actuator, directional control valve, sequential impulses method, semi-implicit Euler scheme, virtual environment systems

Acknowledgements. The publication was carried out within the framework of the state assignment of the Federal Scientific Center NIISI RAS “Conducting fundamental scientific research (47 ГП)” on topic No. FNEF-2022-0012 "Virtual environment systems: Technologies, methods and algorithms for mathematical modeling and visualization”

References

1. Lyashenko, Yu.M., Lyashenko, A.Yu., Revyakina, E.A. (2021) ‘The use of computer simulators in the educational process of single-bucket excavator drivers as a tool for improving the level of functional status indicators’, *Proc. Sci.-Pract. Conf. Modern Applied Research*, 1, pp. 236–242 (in Russ.).

2. Wang, W., Zhao, J. (2020) ‘Energy-efficient robust control for direct drive and energy recuperation hydraulic servo system’, *Complexity*, pp. 1–19. doi: 10.1155/2020/6959273.

3. Zhdanov, A.V. (2016) ‘Mathematical model of position hydraulic drive’, *The Russian Automobile and Highway Industry J.*, 3(49), pp. 87–93 (in Russ.).

4. Miheeva, N.I. (2017) ‘Simulation modelling of the hydraulic drive in the nonlinear approximation’, *Herald of KRSU*, 17(5), pp. 83–86 (in Russ.).

5. Debeleac, C. (2021) ‘Dynamic modeling and simulation of working regime of the hydraulic driven of auger bucket for loader using Matlab/SimHydraulics’, *Hidraulica*, (4), pp. 7–16.

6. Polteva, E.A., Ivanov, D.V., Sandler, I.L., Antonova, V.V. (2020) ‘Modeling of the hydraulic drive of the industrial robot of the “UNIVERSAL 15” model’, *Vestn. SamGUPS*, (2), pp. 74–82 (in Russ.).

7. Rahikainen, J., Kiani-Oshtorjani, M., Sopanen, J., Jalali, P., Mikkola, A. (2018) ‘Computationally efficient approach for simulation of multibody and hydraulic dynamics’, *Mechanism and Machine Theory*, 130, pp. 435–446. doi: 10.1016/j.mechmachtheory.2018.08.023.

8. Mitrev, R., Janosevic, D., Marinkovic, D. (2017) ‘Dynamical modelling of hydraulic excavator considered as a multibody system’, *Tech. Gazette*, 24(2), pp. 327–338. doi: 10.17559/TV-20151215150306.

9. Rahikainen, J., Gonzalez, F., Naya, M.A., Sopanen, J., Mikkola, A. (2020) ‘On the cosimulation of multibody systems and hydraulic dynamics’, *Multibody System Dynamics*, 50(2), pp. 143-167. doi: 10.1007/s11044-020-09727-z.

10. Kurinov, I., Orzechowski, G., Hamalainen, P., Mikkola, A. (2020) ‘Automated excavator based on reinforcement learning and multibody system dynamics’, *IEEE Access*, 8, pp. 213998–214006. doi: 10.1109/ACCESS.2020.3040246.

11. Mikhailyuk, M.V., Maltsev, A.V., Timokhin, P.Yu., Strashnov, E.V., Kryuchkov, B.I., Usov, V.M. (2020) ‘The VirSim virtual environment system for the simulation complexes of cosmonaut training’, *Manned Spaceflight*, (4), pp. 72–95 (in Russ.). doi: 10.34131/MSF.20.4.72-95.

12. Manring, N., Fales, R. (2019) *Hydraulic Control Systems*. USA, NY, JohnWiley Publ., 389 p. doi: 10.1002/9781119418528.

13. Michaylyuk, M.V., Strashnov, E.V. (2016) ‘Simulating dynamics of the system of articulated rigid bodies with joint friction’, *Sci. and Education of the Bauman MSTU*, 1, pp. 108–124 (in Russ.). doi: 10.7463/0116.0830582.

14. Strashnov, E.V., Omelchenko, D.V. (2020) ‘Simulation of anthropomorphic walking robot dynamics in virtual environment systems’, *INJOIT*, 8(10), pp. 66–72 (in Russ.).

15. Drong, V.I., Dubinin, V.V., Il'in, M.M. et al. (2017) *Theoretical Mechanics Course*. Moscow, 580 p. (in Russ.).

Авторы
Страшнов Евгений Владимирович [1],
научный сотрудник, strashnov_evg@mail.ru
Мироненко Ирина Николаевна [1],
научный сотрудник, vstudio@niisi.ras.ru

[1] Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (ФГУ ФНЦ НИИСИ РАН),
г. Москва, 117218, Россия

Authors
Evgeny V. Strashnov [1], Research Associate,
strashnov_evg@mail.ru
Irina N. Mironenko [1], Research Associate,
vstudio@niisi.ras.ru

[1] Federal State Institution
"Scientific Research Institute
for System Analysis of the Russian
Academy of Sciences" (SRISA RAS),
Moscow, 117218, Russian Federation

УДК 004.056 doi: 10.15827/0236-235X.142.590-599 2023. Т. 36. № 4. С. 590–599

Идентификация фейк-новостей с помощью веб-ресурса на основе нейронных сетей

М.В. Тумбинская
Р.А. Галиев

Ссылка для цитирования
Тумбинская М.В., Галиев Р.А. Идентификация фейк-новостей с помощью веб-ресурса на основе нейронных сетей // Программные продукты и системы. 2023. Т. 36. № 4. С. 590–599. doi: 10.15827/0236-235X.142.590-599
Информация о статье
Поступила в редакцию: 01.06.2023 После доработки: 14.07.2023 Принята к публикации: 31.07.2023

Аннотация. Активное распространение информации средствами телекоммуникаций обусловило актуальность проблемы идентификация фейк-новостей на основе их заголовков. Целью данного исследования является повышение уровня достоверности информации пользователей в браузерной среде. Достижение ее возможно путем автоматизированного распознавания фейк-новостей на основе их заголовков в браузерной среде. В работе предложено использовать нейронные сети, которые позволяют проводить семантический анализ заголовков фейк-новостей и обрабатывать большие объемы данных. В статье рассмотрены популярные зарубежные и отечественные веб-ресурсы по идентификации фейк-новостей, их отличительные особенности. Проведены анализ и выбор нейросетевых моделей BERT-MLP, BERT-CNN, BERT-LSTM, которые были заложены в основу разработки веб-ресурса. Для обучения нейросетевых моделей использованы выборки русскоязычных заголовков новостей. Проведено экспериментальное исследование оценки качества и адекватности нейросетевых моделей на тестовых и валидационных выборках данных, в том числе на новостных заголовках различных тематик. Исходя из результатов оценки, разработанный веб-ресурс, в основу которого заложены модели BERT-CNN, BERT-GRU, BERT-LSTM, показал высокую результативность. Достоверность идентификации фейк-новостей составила 86,29 %. Практическая ценность работы заключается в том, что созданный веб-ресурс может быть использован в качестве инструмента для распознавания фейк-новостей специалистами по противодействию деструктивным политическим технологиям, а также специалистами по информационной безопасности. Для повышения точности классификации целесообразно расширить обучающую выборку данных, что позволит выявлять в данных более сложные зависимости.

Ключевые слова: фейк-новость, нейросетевая модель, BERT, CNN, GRU, LSTM, заголовок новости, веб-ресурс, нелегитимные новости

Введение. В современном мире информация является неотъемлемой частью жизни людей. Ее активное распространение происходит всеми возможными источниками: телевидение, радио, газеты, журналы, новостные сайты, социальные сети. Однако не вся информация достоверна. Распространяемые фейк-новости по своей структуре имеют большое сходство с официальными, однако их суть искажена, и они могут нанести серьезный ущерб как отдельным людям, вводя их в заблуждение, так и обществу в целом, создавая напряженность, влиять на общественное мнение, вызывать панику и хаос.

Так, весной 2020 года появились фейк-новости о распространении COVID-19 по сетям связи 5G. Это привело к последствиям в социальной сфере – часть вышек этих сетей в Великобритании были атакованы, вследствие чего у некоторых людей сложилось негативное мнение о данной технологии. Примером может служить и распространение фейк-новостей в процессе выборов в США в 2016 году. Под воздействием дезинформации в социальных сетях некоторые избиратели отдавали предпочтение одному из кандидатов. Таким образом, для общества и государства фейк-новости представляют потенциальную и реальную угрозу, поскольку в руках злоумышленника они являются высокоэффективным инструментом социального и психологического воздействия на человека, а также извлечения собственной выгоды.

Идентификация фейк-новостей достаточно изучена, но остается актуальной задачей.

Для борьбы с распространением фейк-новостей и снижения их воздействия создаются специальные веб-ресурсы, предназначенные для идентификации таких новостей. Они удобны и не требуют от пользователя каких-либо специальных знаний и навыков.

Предметом данного исследования являются информационные системы идентификации фейк-новостей. В качестве методов исследования используются нейросетевые технологии и машинное обучение. Ручной способ проверки новостей на основе их заголовков довольно трудоемкий, в связи с чем целесообразна разработка информационных систем идентификации фейк-новостей на основе нейросетевых технологий. В работе предложен веб-ресурс, в который заложены гибридные нейросетевые

модели BERT-MLP, BERT-CNN, BERT-LSTM, позволяющий в автоматизированном режиме идентифицировать фейк-новость по заголовку. Практическая значимость заключается в том, что разработанный веб-ресурс может быть использован в качестве инструмента для распознавания фейк-новостей специалистами по противодействию деструктивным политическим технологиям, а также специалистами по информационной безопасности. Для повышения точности классификации целесообразно расширить обучающую выборку данных, что позволит выявлять более сложные зависимости в данных.

Анализ веб-ресурсов по идентификации фейк-новостей

Для анализа существующих решений были выбраны десять популярных веб-ресурсов: ChatGPT, «Война с фейками», FactCheck, Politifact, Snopes, «Лапша Медиа», Oigetit Fake News Filter, Alt News, Full Fact, AFP Fact Check. Результаты сравнительного анализа позволяют сделать следующие выводы.

1) Проверку фейк-новостей на основании экспертной оценки осуществляют 80 % проанализированных веб-ресурсов. Это может привести к тому, что в зависимости от страны, в которой функционирует веб-ресурс, оценка может быть пристрастной, группа экспертов может не иметь доступа ко всей информации, необходимой для корректной оценки новости, оценка новости может приводить к временным задержкам в силу специфики ее выполнения экспертным методом.

2) Все проанализированные веб-ресурсы имеют возможность ввода требуемой для проверки заголовка новости, однако только 10 % рассмотренных веб-ресурсов позволяют пользователю получить достоверный результат оценки за короткое время.

3) Проверку русскоязычных новостей не поддерживают 70 % проанализированных веб-ресурсов. Это может вызвать затруднения у россиян при желании проверить новость на фейк.

4) Возможность определить уровень достоверности новости в процентах имеют только 20 % рассмотренных веб-ресурсов. Причем данная функция может дать более информативную оценку новости для пользователя по сравнению с классификационной оценкой.

5) Лишь 20 % проанализированных веб-ресурсов используют нейросетевые технологии для идентификации фейк-новостей. Применение данных технологий позволяет за короткий срок сформировать результат оценки введенного пользователем заголовка новости. Следует отметить, что использование комбинации работы нескольких нейронных сетей повышает точность и достоверность оценки фейк-новостей.

Анализ показал необходимость разработки нового веб-ресурса по идентификации заголовков фейк-новостей в браузерной среде, который обеспечивал бы возможность проверки русскоязычных фейк-новостей с отображением классификационной оценки отнесения заголовка новости к категории «истина» или «ложь» и с детализацией уровня достоверности новостного заголовка в процентах. Идентификация заголовков новостей должна осуществляться на основе интеграционной оценки работы нейронных сетей.

Анализ нейросетевых моделей для идентификации фейк-новостей

Поскольку идентификацию фейк-новостей можно интерпретировать как задачу бинарной классификации текстовой информации («истина» или «ложь»), для ее решения возможно использование алгоритмов машинного обучения с учителем средствами нейросетевых моделей [1]. Для формирования требований к выбору нейросетевых моделей данного типа рассмотрим исследования в предметной области. В работах [2–4] был проведен сравнительный анализ эффективности рекуррентных (RNN) и сверточных (CNN) нейронных сетей, а также нейронных сетей на основе архитектуры трансформеров (Transformer based) для решения задачи идентификации англоязычных фейк-новостей на тему COVID-19. Анализ работ [5–7] показал, что наилучшими нейросетевыми моделями являются модель BERT (точность оценки 97,71 %), а также гибридная модель BERT-CNN (точность оценки 97,52 %). В работе [6] предложена модель классификации текстовой информации на английском языке, комбинирующая результаты работы трех нейросетевых гибридных моделей. В первой модели использованы нейронная сеть с архитектурой трансформеров BERT и рекуррентная нейронная сеть с долгой краткосрочной памятью LSTM. Во второй модели также использован BERT, но уже с применением двух параллельных нейронных сетей LSTM (BiLSTM). Третья модель сформирована из модели BERT

и рекуррентной нейронной сети с управляемыми блоками GRU. Сделан вывод о том, что комбинированный подход имеет лучшую точность оценки (89,81 %) в сравнении с используемыми по отдельности моделями и другими методами машинного обучения. В [7] исследовано применение различных методов машинного обучения и нейронных сетей при обнаружении англоязычных заголовков фейк-новостей в социальных сетях. Показано, что наилучшими нейросетевыми моделями являются гибридная модель BERT-LSTM (точность 97,55 %) и модель BERT-CNN, именуемая в данной статье FakeBERT (98,90 %).

Таким образом, для эффективной идентификации фейк-новостей на основе их семантического анализа необходимо использовать следующие гибридные нейросетевые модели:

– BERT-CNN (основана на архитектуре трансформера BERT, а также сверточной нейронной сети CNN);

– BERT-GRU (основана на модели BERT, а также рекуррентной нейронной сети с управляемыми блоками GRU);

– BERT-LSTM (рекуррентная нейронная сеть с долгой краткосрочной памятью LSTM) [8, 9].

Кроме этого, для достижения лучших результатов необходимо использовать механизм комбинирования результатов работы представленных нейросетевых моделей.

Разработка веб-ресурса по идентификации фейк-новостей на основе нейронных сетей

Веб-ресурс был реализован в виде клиент-серверного веб-приложения. Разработка веб-ресурса была проведена в три этапа. На первом этапе разрабатывался веб-интерфейс. На втором – элементы серверной части, отвечающие за взаимодействие с клиентской частью, – маршруты и обработчики маршрутов Flask. На третьем этапе осуществлялись разработка и программная реализация модулей серверной части, отвечающих за обработку введенной пользователем информации и выдачу результатов классификации.

Взаимодействие клиентской и серверной частей построено таким образом, что после получения новостной ссылки клиентская часть отправляет запрос на серверную, в которой происходит ее дальнейшая обработка. В результате формируется классификационная оценка – отнесение обнаруженного заголовка новости к категории «истина» или «ложь» с детализацией уровня достоверности новостного заголовка в процентах. Результаты классификации отправляются на клиентскую часть в виде ответа и отображаются в веб-интерфейсе пользователя.

Серверная часть имеет последовательную архитектуру и включает в себя пять основных модулей.

1. *Модуль проверки корректности данных*. Используется для проверки введенной ссылки на соответствие общей структуре веб-ссылок. Кроме этого, модуль проверяет доступность источника ссылки путем отправки GET-запроса на получение содержимого страницы и удовлетворительного ответа от сервера источника. Это необходимо для общей проверки введенных пользователем данных, а также возможности дальнейшего извлечения заголовка новости. Если данные некорректны, модуль сообщает об этом и требует ввода повторно.

2. *Модуль проверки источника в реестре СМИ*. Необходим для проверки присутствия источника ссылки в списке зарегистрированных в РФ. Это позволяет пользователю получить дополнительную информацию о достоверности источника введенной ссылки. Проверка осуществляется путем извлечения доменного имени с введенной ссылки в виде текстовой строки и поиска данной строки в списке зарегистрированных СМИ. Список представляет собой файл с расширением .csv, который был предварительно сформирован путем сбора данных с официального реестра Роскомнадзора. В таблице присутствует информация об источниках, относящихся к одной из категорий и имеющих действующую лицензию на осуществление деятельности в области СМИ: сетевое издание, электронное периодическое издание, информационное агентство.

3. *Парсер заголовка новости, или модуль извлечения новостного заголовка*. Выполняет функцию извлечения заголовка новости с введенной ссылки на новость с веб-ресурса, социальной сети VK или мессенджера Telegram. Для извлечения новостного заголовка под каждый из указанных источников разработаны отдельные функции. Вследствие ограничения после извлечения заголовка новости данная строка проверяется на наличие слов, не являющихся русскими. Если такие слова обнаружены, модуль сообщает о их наличии и требует ввода данных повторно.

4. *Препроцессор, или модуль предварительной обработки новостного заголовка*. Выполняет операции преобразования текстового заголовка новости в формат, необходимый для

использования классификатора. К функциям препроцессора относятся:

– удаление пробелов, знаков препинания, а также спецсимволов;

– приведение всех слов заголовка новости к нижнему регистру;

– токенизация: разбиение заголовка новости на отдельные части (токены); добавление в начало и в конец последовательности специальных токенов: [CLS] для обозначения начала предложения (численное представление – 101) и [SEP] для обозначения конца последовательности (численное представление – 102); присвоение каждому токену специального числового идентификатора; формирование бинарной маски заголовка новости для уточнения длины заголовка и корректной работы классификатора.

Данный модуль возвращает последовательность числовых идентификаторов каждого токена и бинарную маску, которые далее передаются на вход классификатора.

5. *Классификатор, или модуль комбинирования результатов работы нейросетевых моделей*. Данный модуль предназначен для формирования классификационной оценки отнесения заголовка новости к категории «истина» или «ложь» с детализацией уровня достоверности новостного заголовка в процентах. В основе данного модуля – механизм комбинирования результатов работы гибридных нейросетевых моделей BERT-GRU, BERT-CNN и BERT-LSTM. Каждая из этих моделей обучена на специально подготовленной выборке данных, включающей как легитимные, так и нелегитимные заголовки новостей на русском языке.

В качестве входных данных классификатор получает предварительно обработанные данные из препроцессора, которые затем используются для решения задачи бинарной классификации (0 – «истина», 1 – «ложь»). Выходными выступают числовые вероятности бинарной классификации, лежащие в диапазоне от 0 до 1, над которыми затем выполняется операция усреднения. Если итоговая оценка находится в диапазоне от 0 до 0,5, заголовок относится к категории «истина», в противном случае – «ложь». Кроме этого, классификатор использует данную числовую оценку для детализации уровня достоверности новостного заголовка в процентах.

После того как серверная часть обработает запрос от клиентской части, она формирует ответ, который содержит данные о результатах классификации, и отправляет его на клиентскую часть.

Разработка клиентской части веб-ресурса. Пользователь имеет возможность проверить заголовок новости на фейк путем ввода в соответствующее текстовое поле ссылки (URL) на новость с веб-ресурса, социальной сети VK или мессенджера Telegram. Чтобы отправить введенную ссылку в виде HTTP-запроса POST на серверную часть и запустить алгоритм проверки заголовка новости на фейк, пользователю необходимо нажать на кнопку «Проверить».

Для отображения отрицательного ответа от сервера в процессе обработки введенного заголовка предусмотрено специальное поле, в котором в текстовом формате выводится одно из следующих сообщений: «Введена некорректная ссылка», «Веб-страница недоступна или не отвечает», «Обнаружены иностранные слова». Иначе в данном поле отображается: «Источник является официальным СМИ», то есть доменное имя источника ссылки присутствует в списке СМИ, зарегистрированных в Роскомнадзоре РФ, «Источник не является официальным СМИ» – в противном случае. После получения положительного ответа от серверной части предусмотрено текстовое поле для отображения результатов классификационной оценки отнесения обнаруженного заголовка новости к категории («истина» или «ложь»), с детализацей уровня достоверности заголовка новости в процентах. Кроме того, для детализации результатов работы каждой нейросетевой модели в отдельности рядом с кнопкой «Проверить» появляется кнопка «Подробнее». При нажатии на данную кнопку отображаются три текстовых поля, содержащих результаты работы нейросетевых моделей: BERT-CNN, BERT-GRU, BERT-LSTM (http://www.swsys.ru/uploaded/image/2023-4/1.jpg).

Разработка серверной части веб-ресурса. Для создания модуля взаимодействия клиентской и серверной частей разрабатываемого веб-ресурса был использован фреймворк Flask [10] на языке программирования Python. В процессе программной реализации серверной части были подключены и использованы следующие модули Python:

```
import re
import numpy as np
import pandas as pd
import requests
import vk_api
import tensorflow as tf
from urllib.parse import urlparse
from newspaper import Article
from bs4 import BeautifulSoup
```

```
   from flask import Flask, render_tem-
plate, request, jsonify
   from selenium import webdriver
   from selenium. webdriver.common.by
import By
   from transformers import TFBert-
Model, BertTokenizer
   from keras.models import load_model
   from keras.callbacks import Ear-
lyStopping
   from sklearn.model_selection import
train_test_split
```

Чтобы серверная часть смогла получать и обрабатывать запросы пользователя на получение содержимого веб-ресурса, а также введенную пользователем новостную ссылку, был инициализирован объект Flask с указанием маршрутов и обработчиков:

– '/': отображает главную страницу (веб-интерфейс), когда пользователь пытается получить доступ к веб-ресурсу (метод запроса GET); при использовании данного маршрута выполняется обработчик, который возвращает на клиентскую часть содержимое HTML-страницы;

– '/check_web_link': обрабатывает событие нажатия пользователем кнопки «Проверить» (метод запроса POST); связанный обработчик получает введенный пользователем текст, а затем последовательно выполняет запуск модулей проверки корректности данных и проверки источника в реестре СМИ; результаты выполнения функции отправляются на клиентскую часть в виде ответа в формате JSON;

– '/predict': обрабатывает событие нажатия пользователем кнопки «Проверить» (метод запроса POST), когда работа вышеизложенных модулей завершается успешно; обработчик получает введенный пользователем текст, а затем выполняет последовательный запуск модулей «Парсер» заголовка новости», «Препроцессор» и классификатора, результаты выполнения функции отправляются на клиентскую часть в виде ответа в формате JSON.

Сбор исходных данных для обучения нейросетевых моделей. Для обучения нейросетевых моделей, используемых в основе классификатора, были собраны как легитимные, так и нелегитимные заголовки новостей на русском языке. Сбор легитимных новостных заголовков осуществлялся с официальных новостных веб-ресурсов: сетевых изданий «Российская Газета», «РИА Новости», «Реальное время», IT-World.ru «РБК», информационного агентства «ТАСС», портала новостей «Новости Mail.ru», электронного периодического издания «БИЗНЕС Online».

Сбор нелегитимных новостных заголовков осуществлялся с различных веб-ресурсов и каналов мессенджера Telegram, в которых публикуются как сатирические, так и широко распространенные фейк-новости из Интернета. Перечень источников нелегитимных новостей: каналы мессенджера Telegram «Fake Control» и «Осторожно, ФЕЙК», веб-ресурсы «Война с фейками», «Лапша Медиа», «Панорама».

В основе функций сбора данных используются возможности подключаемых модулей: Selenium – для выполнения автоматизированного сбора данных, Pandas – для сохранения полученных данных в формат pandas.DataFrame. В случае сбора данных с общедоступных каналов мессенджера Telegram дополнительно используются возможности подключаемых модулей Requests для получения содержимого каналов в веб-версии мессенджера Telegram в формате .html и BeautifulSoup4 для обработки полученных данных и извлечения новостных заголовков.

Все собранные заголовки сохраняются в файле формата .csv. В данном файле содержится 40 000 новостных заголовков, в которых равное количество легитимных и нелегитимных заголовков новостей – по 20 000 каждого типа.

Общая структура нейросетевых моделей. В основе классификатора новостных заголовков лежат гибридные нейросетевые модели BERT-CNN, BERT-GRU, BERT-LSTM, которые выполняют классификационную оценку заголовков на основе двух категорий – «истина» и «ложь». Структура, а также значения гиперпараметров гибридных нейросетевых моделей были заимствованы в работах [1, 6].

Рассмотрим общую структуру используемых нейросетевых моделей.

Входными данными являются последовательность числовых идентификаторов новостного заголовка и бинарная маска новостного заголовка.

Каждая из гибридных нейросетевых моделей использует нейросетевой слой, основанный на архитектуре трансформеров BERT, для создания векторного представления каждого числового идентификатора в заголовке новости с учетом бинарной маски для уточнения длины заголовка. Для улучшения точности и сокращения времени обучения каждая из нейросетевых моделей использует заранее предобученную нейросетевую модель BERT, предобученную на русском корпусе слов – sberbank-ai/ruBert-base (https://huggingface.co/ai-forever/ruBert-base).

Полученные векторные представления затем передаются в нейросетевые слои в зависимости от конкретной гибридной модели (CNN, GRU или LSTM) для обнаружения характерных признаков каждой из категорий новостных заголовков. Таким образом, гибридные модели используют семантический анализ для выявления фейк-новостей.

Далее для обработки полученных из предыдущего слоя признаков используется полносвязный нейросетевой слой. Важно отметить, что первоначально в модели BERT-CNN этот слой отсутствовал, однако был добавлен вследствие улучшения точности распознавания обеих категорий новостных заголовков. В моделях BERT-GRU и BERT-LSTM [4] он присутствует изначально и остается неизменным.

Кроме этого, поскольку нейросетевые модели должны относить новостные заголовки к одной из двух категорий, ключевой задачей является выполнение бинарной классификации заголовков новостей (0 – «истина», 1 – «ложь»). Для этого в конце каждой модели используется полносвязный нейросетевой слой, содержащий один нейрон, выполняющий данную задачу.

В качестве выходных данных выступают числовые вероятности бинарной классификации, лежащие в диапазоне от 0 до 1.

Построение и обучение нейросетевых моделей. Для построения и обучения гибридных нейросетевых моделей BERT-CNN, BERT-GRU и BERT-LSTM используется следующие подключаемые модули: Transformers для использования нейросетевого слоя BERT, Tensorflow для построения общей архитектуры, обучения и сохранения нейросетевых моделей, Scikit-learn для разделения исходных данных на тренировочную и валидационную выборки.

Поскольку используемые нейросетевые модели могут обрабатывать данные только одинакового размера, в качестве максимальной длины новостного заголовка было выбрано значение 21 (рис. 1). Выбор осуществлялся с учетом необходимости минимизации длины заголовка для ускорения обучения нейросетевых моделей. Кроме этого, 97,35 % всех новостных заголовков имеют длину, не превышающую 21 слово.

Следует отметить, что в программной реализации нейросетевых моделей установлена максимальная длина последовательности, равная 23 токенам. Это связано с тем, что в процессе токенизации в начале и в конце обработанной последовательности добавляются специальные токены [CLS] (численное представление – 101) и [SEP] (численное представление – 102).

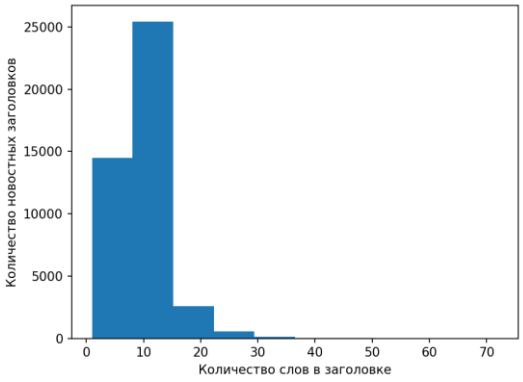


Рис. 1. Гистограмма распределения новостных заголовков в зависимости от количества слов

Fig. 1. News headline distribution histogram depending on word number

В качестве обучающей выборки случайным образом были выбраны 80 % исходных новостных заголовков. Остальные 20 % использовались как валидационная выборка для оценки точности нейросетевых моделей по окончании каждой эпохи обучения и настройки значений гиперпараметров нейросетевых моделей. Стоит отметить, что при обучении выбранных нейросетевых моделей валидационная выборка использовалась для нахождения оптимального количества эпох обучения. Благодаря этому обучение нейросетевых моделей осуществлялось в течение двух эпох, поскольку именно на третьей эпохе в данных моделях возникает явление «переобучения» [11].

Программная реализация классификатора. В основе работы классификатора лежит принцип комбинирования результатов работы трех обученных гибридных нейросетевых моделей – BERT-CNN, BERT-GRU, BERT-LSTM. Классификатор получает на вход данные от препроцессора, а затем последовательно выполняет запуск функций предсказания каждой нейросетевой модели в отдельности для получения бинарной вероятности отнесения заголовка новости к одной из двух категорий («истина» или «ложь»). После этого классификатор выполняет функцию усреднения для получения итогового числового результата предсказания, лежащего в диапазоне от 0 до 1.

Если итоговая оценка находится в диапазоне от 0 до 0,5, заголовок относится к категории «истина», в противном случае – «ложь». Кроме этого, классификатор использует данную число-

вую оценку для детализации уровня достоверности новостного заголовка (в процентах).

Таким образом, классификатор возвращает значение классификационной оценки с детализацией уровня достоверности новостного заголовка в процентах от каждой нейросетевой модели, включая результат их комбинирования.

Экспериментальное исследование оценки качества разработанного веб-ресурса

Разработанный веб-ресурс предназначен для выполнения классификационной оценки новостных заголовков («истина» или «ложь»). Главным компонентом веб-ресурса, выполняющим данную классификацию, является модуль комбинирования результатов работы гибридных нейросетевых моделей BERT-CNN, BERT-GRU, BERT-LSTM.

Вследствие этого для проведения оценки качества необходимо проверить, насколько точно и адекватно нейросетевые модели, лежащие в основе веб-ресурса, справляются с задачей классификационной оценки легитимных и нелегитимных заголовков новостей. Для этого воспользуемся метриками оценки качества нейросетевых моделей (в процентах), предложенными в работах [1, 6, 12].

Метрика общей точности (Accuracy) отражает долю всех правильно классифицированных новостных заголовков от их общего числа:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Метрика точности (Precision) отражает долю верно классифицированных нелегитимных новостных заголовков относительно общего числа заголовков новостей, отнесенных нейросетевой моделью к категории «ложь»:

$$Precision = \frac{TP}{TP + FP}.$$

Полнота (Recall) отражает долю верно классифицированных нелегитимных новостей относительно общего числа фактических нелегитимных новостей:

$$Recall = \frac{TP}{TP + FN}.$$

F1-мера (F1) является гармоническим средним между метриками точности и полноты и отражает компромисс между ними:

$$F = \frac{2(Precision \cdot Recall)}{Precision + Recall}.$$

В данных формулах используются следующие обозначения [12]: *TP* (true positive) – количество нелегитимных новостей, которые были правильно отнесены нейросетевой моделью к категории «ложь»; *TN* (true negative) – количество легитимных новостей, которые были правильно отнесены нейросетевой моделью к категории «истина»; *FP* (false positive) – количество легитимных новостей, которые были ошибочно отнесены нейросетевой моделью к категории «ложь»; *FN* (false negative) – количество нелегитимных новостей, которые были ошибочно отнесены нейросетевой моделью к категории «истина».

Результаты общей оценки нейросетевых моделей представлены в таблице. Оценка проводилась на валидационной выборке размером 8 000 новостных заголовков (по 4 000 на каждую категорию), а также на тестовой выборке, сформированной путем сбора 4 000 новостных заголовков (по 2 000 на каждую категорию). Сбор данных для тестовой выборки был сделан спустя две недели после сбора данных исходной (обучающей и валидационной) выборки.

Это необходимо для оценки возможности нейросетевых моделей осуществлять задачу идентификации заголовков, не использованных во время обучения и являющихся случайными для моделей. Стоит отметить, что в таблице также приводится оценка качества классификатора (RuFakeBERT), комбинирующего результаты работ приведенных гибридных моделей.

По итогам оценки разработанный модуль комбинирования результатов работы гибридных нейросетевых моделей BERT-CNN, BERT-GRU, BERT-LSTM имеет наилучшие показатели:

Результаты оценки нейросетевых моделей

Results of evaluating neural network models

Наименоменование модели	Валидационная выборка, %				Тестовая выборка, %			
	Acc	Pr	Rec	F1	Acc	Pr	Rec	F1
BERT-CNN	86,17	83,8	86,13	84,95	85,57	82,1	85,32	83,68
BERT-GRU	86,82	84,79	85,85	85,29	85,97	81,93	85,89	83,91
BERT-LSTM	86,48	83,39	86,51	84,92	86,54	83,81	85,44	84,62
RuFakeBERT	**87,21**	**84,83**	**86,87**	**85,67**	**86,7**	**83,87**	**85,93**	**84,89**

– на тестовой выборке результат общей точности классификатора составляет 87,21 %, F1-меры – 85,67 % (на основе метрик Precision – 84,83 %, Recall – 86,87 %);

– на валидационной выборке результат общей точности классификатора – 86,7 %, F1-меры – 84,89 % (на основе метрик Precision – 84,83 %, Recall – 86,87 %).

Для более детальной оценки качества классификатора был проведен анализ его способности точно идентифицировать каждую из категорий новостных заголовков («истина» или «ложь») отдельно друг от друга. Для этой оценки используются метрики [12]:

– отрицательного предсказанного значения (NPV), которое отражает долю верно классифицированных легитимных новостных заголовков относительно общего числа заголовков новостей, отнесенных моделью к категории «истина»:

$$NPV = \frac{TN}{TN + FN};$$

– положительного предсказанного значения (PPV), которое отражает долю верно классифицированных нелегитимных новостей относительно общего числа заголовков новостей, отнесенных моделью к категории «ложь»:

$$PPV = \frac{TP}{TP + FP}.$$

Для детализационной оценки качества классификатора как тестовая выборка легитимных и нелегитимных данных были выбраны новостные заголовки на различные темы: спорт, политика, общество, культура, медицина, образование, экономика, IT-технологии.

Детализационная оценка была проведена на тестовой выборке, состоящей из 4 000 новостных заголовков, по 500 заголовков каждой тематики. В каждой тематике было использовано равное количество легитимных и нелегитимных новостных заголовков. Результаты оценки представлены на диаграммах (рис. 2).

Таким образом, можно сделать вывод о том, что разработанный модуль более качественно и адекватно классифицирует легитимные новости на тему политики (89,6 %), а также нелегитимные новости на тему общества (92,4 %). Кроме этого, было выявлено, что самые низкие результаты оценки классификатор выдает при оценке легитимных новостных заголовков на тему культуры (82,6 %), а также нелегитимных спортивных новостей (83,5 %).

Для формирования итоговой оценки качества воспользуемся усредненными результатами нейросетевых моделей, а также классификатора по всем использованным метрикам: BERT-CNN – 85,09 %, BERT-GRU – 85,49 %, BERT-LSTM – 85,64 %, RuFakeBERT – 86,29 %.

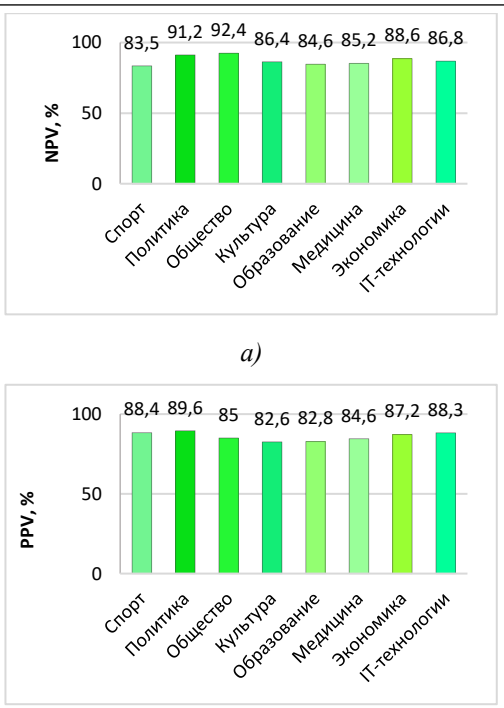


Рис. 2. Гистограмма оценки качества классификатора на выборке легитимных новостных заголовков: а) по метрике NPV; б) по метрике PPV

Fig. 2. Histogram of classifier quality assessment on a sample of legitimate news headlines: a) NPV metric; b) PPV metric

Как видно, разработанный классификатор (RuFakeBERT) показал улучшение результатов оценки качества по сравнению с нейросетевыми моделями BERT-CNN, BERT-GRU и BERT-LSTM на 1,2 %, 0,8 % и 0,65 % соответственно.

Для повышения точности классификации целесообразно расширить обучающую выборку данных. Это позволит нейросетевым моделям, используемым в модуле классификатора, выявлять более сложные зависимости в данных. Такой подход улучшит способность моделей к обобщению и повысит точность предсказаний для свежих новостных заголовков.

Заключение

В работе представлен обзор решений в области идентификации фейк-новостей, предложен веб-ресурс по идентификации заголовков

фейк-новостей на русском языке на основе интеграционной работы нейронных сетей с отображением классификационной оценки отнесения заголовка новости к категории «истина» или «ложь», с детализацией уровня достоверности новостного заголовка в процентах, проведено экспериментальное исследование. Новизной является программная реализация инструмента идентификации фейк-новостей.

Результаты исследований показали, что на тестовой выборке общая точность классификатора составила 87,21 %, F1-меры – 85,67 %, на валидационной выборке – соответственно 86,7 % и 84,89 %. Была проведена дополнительная оценка качества классификатора на каждой из категорий новостных заголовков («истина» или «ложь») отдельно друг от друга с использованием метрик NPV и PPV. Так, данный модуль лучше всего справляется с классификацией легитимных политических новостных заголовков (89,6 %), а также нелегитимных общественных новостей (92,4 %). Худшие результаты модель показала при классификации легитимных новостей о культуре (82,6 %) и нелегитимных спортивных новостей (83,5 %). Разработанный классификатор RuFakeBERT показал улучшение результатов оценки качества по всем использованным метрикам по сравнению с нейросетевыми моделями BERT-CNN, BERT-GRU и BERT-LSTM на 1,2 %, 0,8 % и 0,65 % соответственно. Итоговая оценка качества классификатора составила 86,29 %.

Экспериментальные исследования показали необходимость увеличения объема выборки для обучения нейросетевых моделей для более качественного анализа данных и достоверного процесса идентификации. В дальнейшем планируются разработка системы поддержки принятия решений для автоматического распознавания фейк-новостей и соответствующая доработка веб-ресурса.

Список литературы

1. Alghamdi J., Lin Y., Luo S. A comparative study of machine learning and deep learning techniques for fake news detection. Information, 2022, vol. 13, no. 12, art. 576. doi: 10.3390/info13120576.

2. Мухаматханов Р.М., Михайлов А.А., Баянов Б.И., Тумбинская М.В. Классификация DDos-атак на основе нейросетевой модели // Прикладная информатика. 2019. Т. 14. № 1. С. 96–103.

3. Wang H., Tian K., Zhengjiang Wu, Wang L. A short text classification method based on convolutional neural network and semantic extension. IJCIS, 2021, vol. 14, no. 1, pp. 367–375. doi: 10.2991/ijcis.d.201207.001.

4. Тумбинская М.В., Баянов Б.И., Рахимов Р.Ж., Кормильцев Н.В., Уваров А.Д. Анализ и прогнозирование вредоносного сетевого трафика в облачных сервисах // Бизнес-информатика. 2019. Т. 13. № 1. С. 71–81. doi: 10.17323/1998-0663.2019.1.71.81.

5. Тумбинская М.В. Анализ и прогнозирование сетевых атак типа SYN-FLOOD на web-ресурсы // Информатизация образования и науки. 2018. № 4. С. 61–68.

6. Tan K.L., Lee C.P., Lim K.M., Anbananthen K.S.M. Sentiment analysis with ensemble hybrid deep learning model. IEEE Access, 2022, vol. 10, pp. 103694–103704. doi: 10.1109/ACCESS.2022.3210182.

7. Kaliyar R.K., Goswami A., Narang P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools and Applications, 2021, vol. 80, no. 8, pp. 11765–11788. doi: 10.1007/s11042-020-10183-2.

8. Xing Y., Xiao C. A GRU model for aspect level sentiment analysis. J. Phys.: Conf. Ser., 2019, vol. 1302, art. 032042. doi: 10.1088/1742-6596/1302/3/032042.

9. Triyadi I., Prasetiyo B., Nikmah T.L. News text classification using Long-Term Short Memory LSTM algorithm. J. Soft Comput. Explor., 2023, vol. 4, no. 2, pp. 79–86. doi: 10.52465/joscex.v4i2.136.

10 Grinberg M. Flask Web Development: Developing Web Applications with Python. CA, Sebastopol, O'Reilly Media Publ., 2018, 292 p.

11. Ying X. An overview of overfitting and its solutions. J. Phys.: Conf. Ser., 2022, vol. 1168, art. 022022. doi: 10.1088/1742-6596/1168/2/022022.

12. De Diego I.M., Redondo A.R., Fernandez R.R. General performance score for classification problems. Applied Intelligence, 2022, vol. 52, no. 10, pp. 12049–12063. doi: 10.1007/s10489-021-03041-7.

Software & Systems doi: 10.15827/0236-235X.142.590-599 2023, vol. 36, no. 4, pp. 590–599

Neural network-based approach for identifying fake news

Marina V. Tumbinskaya
Renat A. Galiev

For citation
Tumbinskaya, M.V., Galiev, R.A. (2023) ‘Neural network-based approach for identifying fake news’, *Software & Systems*, 36(4), pp. 590–599 (in Russ.). doi: 10.15827/0236-235X.142.590-599

Article info
Received: 01.06.2023 After revision: 14.07.2023 Accepted: 31.07.2023

Abstract. Currently, telecommunication facilities actively disseminate news. Identification of fake news based on their headlines has become a relevant problem. The purpose of this study is to increase a user information reliability level in browser environment. Automated recognition of fake news based on their headlines in the browser environment can achieve this goal. The paper proposes using neural networks that allow analyzing fake news headlines semantically and processing large amounts of data. The article examines popular foreign and domestic web resources for identifying fake news, their distinctive features. The authors carried out an analysis and selected neural network models BERT-MLP, BERT-CNN, BERT-LSTM to be a basis for developing a web resource. They used samples of Russian-language news headlines to train neural network models. An experimental study was carried out to assess the quality and adequacy of neural network models on test and validation data samples including news headlines on various topics. The developed web resource based on the BERT-CNN, BERT-GRU, BERT-LSTM models showed good results. The reliability of identifying fake news was 86,29%. The practical value of the work is in the fact that the developed web resource can be used as a tool for recognizing fake news by specialists in counteracting destructive political technologies, as well as information security specialists. To improve classification accuracy, it is advisable to expand a training data set, which allows identifying more complex dependencies in data.
Keywords: fake news, neural network model, BERT, CNN, GRU, LSTM, news headline, illegitimate news

References

1. Alghamdi, J., Lin, Y., Luo, S. (2022) 'A comparative study of machine learning and deep learning techniques for fake news detection', *Information*, 13(12), art. 576. doi: 10.3390/info13120576.
2. Mukhamatkhanov, R., Mikhailov, A., Bayanov, B., Tumbinskaya, M. (2019) 'Classification of DDoS-attacks based on the neural network model', *J. of Applied Informatics*, 14(1), pp. 96–103 (in Russ.).
3. Wang, H., Tian, K., Zhengjiang, Wu, Wang, L. (2021) 'A short text classification method based on convolutional neural network and semantic extension', *IJCIS*, 14(1), pp. 367–375. doi: 10.2991/ijcis.d.201207.001.
4. Tumbinskaya, M.V., Bayanov, B.I., Rakhimov, R.Zh., Kormiltcev, N.V., Uvarov, A.D. (2019) 'Analysis and forecast of undesirable cloud services traffic', *Business Informatics*, 13(1), pp. 71–81 (in Russ.). doi: 10.17323/1998-0663. 2019.1.71.81.
5. Tumbinskaya, M.V. (2018) 'Analysis and forecasting of SYN-FLOOD network attacks to web-resources', *Informatization of Education and Sci.*, (4), pp. 61–68 (in Russ.).
6. Tan, K.L., Lee, C.P., Lim, K.M., Anbananthen, K. (2022) 'Sentiment analysis with ensemble hybrid deep learning model', *IEEE Access*, 10, pp. 103694–103704. doi: 10.1109/ACCESS.2022.3210182.
7. Kaliyar, R.K., Goswami, A., Narang, P. (2021) 'FakeBERT: Fake news detection in social media with a BERT-based deep learning approach', *Multimedia Tools and Applications*, 80(8), pp. 11765–11788. doi: 10.1007/s11042-020-10183-2.
8. Xing, Y., Xiao, C. (2019) 'A GRU model for aspect level sentiment analysis', *J. Phys.: Conf. Ser*., 1302, art. 032042. doi: 10.1088/1742-6596/1302/3/032042.
9. Triyadi, I., Prasetiyo, B., Nikmah, T.L. (2023) 'News text classification using Long-Term Short Memory (LSTM) algorithm', *J. Soft Comput. Explor.*, 4(2), pp. 79–86. doi: 10.52465/joscex.v4i2.136.
10. Grinberg, M. (2018) *Flask Web Development: Developing Web Applications with Python*. CA, Sebastopol: O'Reilly Media Publ., 292 p.
11. Ying, X. (2022) 'An overview of overfitting and its solutions', *J. Phys.: Conf. Ser*., 1168, art. 022022. doi: 10.1088/1742-6596/1168/2/022022.
12. De Diego, I.M., Redondo, A.R., Fernandez, R.R. (2022) 'General performance score for classification problems', *Applied Intelligence*, 52(10), pp. 12049–12063. doi: 10.1007/s10489-021-03041-7.

Авторы
Тумбинская Марина Владимировна [1], к.т.н., доцент, tumbinskaya@inbox.ru
Галиев Ренат Алмазович [1], студент, renat5390@yandex.ru

[1] Казанский национальный исследовательский технический университет им. А.Н. Туполева, г. Казань, 420111, Россия

Authors
Marina V. Tumbinskaya [1], Cand. of Sci. (Engineering), Associate Professor, tumbinskaya@inbox.ru
Renat A. Galiev [1], Student, renat5390@yandex.ru

[1] Kazan National Research Technical University named after A.N. Tupolev, Kazan, 420111, Russian Federation

УДК 004.89 doi: 10.15827/0236-235X.142.600-606 2023. Т. 36. № 4. С. 600–606

Интеграция методов обучения с подкреплением и нечеткой логики для интеллектуальных систем реального времени

А.П. Еремеев
М.Д. Сергеев
В.С. Петров

Ссылка для цитирования
Еремеев А.П., Сергеев М.Д., Петров В.С. Интеграция методов обучения с подкреплением и нечеткой логики для интеллектуальных систем реального времени // Программные продукты и системы. 2023. Т. 36. № 4. С. 600–606. doi: 10.15827/0236-235X.142.600-606

Информация о статье
Поступила в редакцию: 03.07.2023 После доработки: 13.07.2023 Принята к публикации: 14.07.2023

Аннотация. В данной работе рассмотрены возможности интеграции методов обучения с подкреплением и нечеткой логики в плане повышения эффективности алгоритмов обучения с подкреплением. Главное внимание уделяется применению таких интегрированных методов в интеллектуальных системах реального времени, особенно в системах поддержки принятия решений для мониторинга и управления сложными техническими объектами. Как основа используется метод обучения с подкреплением на базе темпоральных различий, состояние среды и сигнал вознаграждения формируются с применением нечеткой логики. Представлена программная реализация и приводятся данные компьютерного моделирования методов глубокого обучения с подкреплением на основе темпоральных различий, полученные при сравнительном анализе алгоритма на основе нечеткой логики и алгоритмов на основе нейронных сетей. Показано, что основными достоинствами алгоритмов обучения с подкреплением с применением нечеткой логики являются: эффективность обучения, выражающаяся в минимизации количества эпизодов, что особенно важно, когда доступность данных для обучения ограничена или обучение в реальном времени требует быстрой адаптации; устойчивость к шуму и выбросам в данных, что важно в реальных средах, где присутствуют шумы или изменяются данные; интерпретируемость – алгоритмы с нечеткой логикой предоставляют интерпретируемые правила и выводы на основе нечеткой логики; расширение области применения обучения с подкреплением на предметные/проблемные области и задачи с непрерывным пространством состояний. Данные исследования и разработки выполняются в рамках конструирования интеллектуальных систем поддержки принятия решений реального времени. Эти системы предназначены для помощи оперативно-диспетчерскому персоналу (лицам, принимающим решения) при мониторинге и управлении сложными техническими и организационными системами в условиях достаточно жестких временных ограничений и при наличии различного типа неопределенностей (неточности, нечеткости, противоречивости) в поступающей в систему информации, то есть так называемых зашумленных данных.

Ключевые слова: искусственный интеллект, обучение с подкреплением, нечеткая логика, интеллектуальная система, реальное время, поддержка принятия решений

Введение. Возможности создания интегрированной среды, объединяющей методы обучения с подкреплением (reinforcement learning, RL), гибкие алгоритмы поиска решения (anytime algorithms) и мультиагентный подход, а также применение БД NO-SQL уже были рассмотрены в [1, 2]. Выявлена перспективность *интеграции в интеллектуальные системы поддержки принятия решений реального времени* (ИСППР РВ), ориентированных на динамические предметные/проблемные области, *искусственных нейронных сетей* (ИНС) и RL-обучения на основе временных (темпоральных) различий (temporal differences, TD). Основные положения RL-обучения изложены в [3].

В данной работе показаны возможности применения нечеткой логики в методах обучения с подкреплением, представлены разработка соответствующих программных средств и результаты компьютерного моделирования на ряде типовых задач.

RL-обучение с учетом временных различий: эффективный подход к обучению с подкреплением

В работе [1] продемонстрировано, что одним из наиболее эффективных подходов к обучению с подкреплением в ИСППР РВ с учетом критерия «временные затраты–качество обучения» является RL-обучение на основе темпоральных различий (TD-обучение). Процесс обучения основан на опыте, полученном при взаимодействии агента с окружающей средой без необходимости предварительного знания о ней. Разработанные для многомерных временных рядов TD-методы обладают способностью обновлять расчетные оценки без ожидания окончательного результата, что делает их самонастраиваемыми. Особенно полезны они в динамических предметных областях и ИСППР РВ семиотического типа. Такие системы способны адаптироваться и подстраиваться к из-

менениям в управляемом объекте и окружающей среде [1, 4].

В наиболее простом TD-методе TD(0) функция ценности (оценка) рассчитывается по следующей формуле:

$$V_{s_t} \leftarrow V(s_t) + \alpha * [r_{t+1} + \gamma * V(s_{t+1}) - V(s_t)],$$

где Vs_t – оценка функции ценности нетерминального состояния s_t в момент времени t; r_{t+1} – полученное вознаграждение на текущем шаге; α – длина шага; γ – ценность терминального состояния.

Важно отметить, что TD-методы дают свои оценки, частично основываясь на предыдущих, что позволяет им самонастраиваться. Преимущества TD-методов в том, что они не требуют знания модели окружающей среды, включая вознаграждения и вероятностное распределение последующих состояний, а также могут оценивать выгоду уже на следующем временном шаге, не ожидая завершения всего эпизода. Даже в случае длительных эпизодов процесс обучения не замедляется.

Известно, что TD-алгоритм для любой заданной стратегии π сходится к функции V в среднем при использовании постоянного шага и в пределе сходится с вероятностью 1, если длина шага уменьшается в соответствии с условиями стохастической аппроксимации:

$$\sum_{k=1}^{\infty} \alpha_k(a) = \infty \text{ и } \sum_{k=1}^{\infty} \alpha_k^2(a) < \infty.$$

В работе [4] показано, что одним из важнейших достижений в области обучения с подкреплением является развитие управления с использованием TD-метода с разделенной оценкой ценности стратегий, известного как Q-обучение [5, 6]. Простейшая форма этого подхода, известная как одношаговое Q-обучение, определяется как

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma * \max_a Q(s_{t+1}, a) - Q(s_t, a_t)],$$

где $Q(s_t, a_t)$ – оценка функции ценности после перехода из нетерминального состояния s при действии a в момент времени t. Если состояние s_{t+1} является терминальным, значение $Q(s_{t+1}, a_{t+1})$ полагается равным нулю.

Схема алгоритма Q-обучения имеет следующий вид:

```
Инициализировать Q(s, a) произвольно
Повторять (для каждого эпизода):
  Инициализировать s
  Повторять (для каждого шага эпизода):
    Найти a по s, используя стратегию, полученную из Q (например, ε-жадную)
    Выполнить действие a, найти r, s′
    Q(s,a)¬Q(s,a)+α[r+
    +γ*maxQ(s',a')-Q(s,a)]
    s ← s′
  Пока s не станет завершающим состоянием
```

Недостатком данного подхода является необходимость дискретизировать все возможные пары «состояние–действие» и анализировать их многократно для нахождения оптимальных значений функции ценности. Чтобы решить эту проблему, предлагается использовать нечеткую логику для моделирования состояний и функции вознаграждения.

Обучение с подкреплением с применением нечеткой логики

Нечеткое Q-обучение (fuzzy Q-learning, FQL) – это расширение алгоритма Q-обучения, позволяющее преодолеть упомянутую выше проблему [7]. Кроме того, FQL позволяет инкапсулировать экспертные знания в систему, ускоряя процесс обучения. В FQL принятие решений представлено системой нечеткого логического вывода (fuzzy inference system, FIS), которая рассматривает большие дискретные или непрерывные состояния как входные данные. Идея алгоритма FQL заключается в использовании так называемой q-таблицы в качестве компактной версии Q-таблицы для представления обучающих данных. В FQL система логического вывода описана набором правил J, где $j \in J$ определяется как

IF(x^1 is L_j^1)...*AND* (x^n is L_j^n)... *AND* (x^N is L_j^N) *THEN* $a = o_j$ with $q(L_j, o_j)$,

где L_j^n – метка входной переменной x^n, x – вектор состояния, $x = [x^1, \ldots, x^n, \ldots, x^N]$, участвующий в j-правиле; $o_j = [o_j^1, \ldots, o_j^k, \ldots, o_j^K]$ – выходное множество действий для правила j, каждому из которых соответствует вектор $L_j = [L_j^1, \ldots, L_j^n, \ldots, L_j^N]$. Значение $Q(L_j, o_j^k)$ называется значением q-функции в состоянии L_j и действии o_j^k j-го правила.

На рисунке 1 приведена структурная схема алгоритма FQL [8]. В слое фаззификации каждая функция принадлежности (μ_L) отображает компоненту состояния в степень принадлежности к нечеткому множеству, соответствующему заданной метке. Пусть J_x обозначает набор всех правил в слое оценки правила, тогда принадлежность вектора состояния x, или степень истинности в терминах нечеткой логики (представленная α), по отношению к j-му пра-

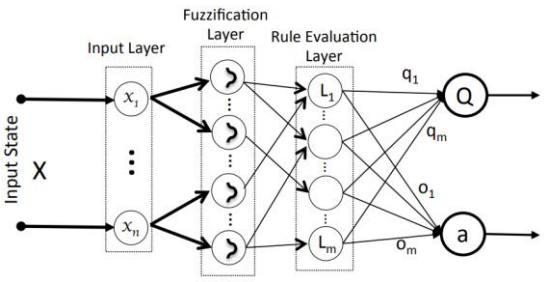


Рис. 1. Структурная схема алгоритма FQL

Fig. 1. FQL algorithm block diagram

вилу, $j \in J_x$, определяется как произведение функций, составляющих правило:

$$\alpha_j(x) = \prod_{n=1}^{N} \mu_{L_j^n}(x_n).$$

Алгоритм FQL имеет двухуровневый выбор действия. На первом уровне (уровень локальных действий) набор действий выбирается в соответствии со стратегией разведки/применения. Такая стратегия позволяет агенту исследовать неопробованные действия, чтобы получить больше опыта, и сочетать это с использованием уже известных успешных действий для обеспечения высокого долгосрочного вознаграждения. В реализации алгоритма используется ε-жадный метод в качестве стратегии для экспериментов. В этом методе на каждом временном шаге агент выбирает случайное действие с фиксированной вероятностью 1 – ε, где ε обычно берется близким к 1. С вероятностью ε действие выбирается жадно из изученных оптимальных действий по отношению к *q*-таблице с вероятностью

$$\begin{cases} \varepsilon, \forall j \in J_x : o_j^l = \arg\max_{k \in K} q(L_j; o_j^k), \\ 1-\varepsilon, \forall j \in J_x : o_j^l = random_{k \in K}(o_j^k), \end{cases}$$

где o_j^1 – выбранное локальное действие *j*-го правила. На втором уровне выбора действия (выбор предполагаемого действия) назначенное действие для входного вектора *x* выбирается из набора локальных действий следующим образом:

$$a = \max_{j \in J_x} a_j(x) o_j^l.$$

Чтобы подчеркнуть временную зависимость, в уравнение добавляется индекс времени. Аппроксимация значения качества состояния x_t вычисляется на основе уравнения

$$Q(x_t, a) = \sum_{j \in J_{x_t}} a_j(x_t) \times q_t(L_j, o_j^l).$$

После выполнения действия *a* система переходит в следующее состояние x_{t+1}, агент получает вознаграждение r_{t+1}. Для входного вектора x_{t+1} вычисляется значение Value-функции состояния:

$$V(x_{t+1}) = \sum_{j \in J_{x_{t+1}}} a_j(x_{t+1}) \times \max_k q_t(L_j, o_j^k).$$

Учитывая приведенные выше уравнения, вычисляется TD-ошибка:

$$\Delta Q = r_{t+1} + \gamma V(x_{t+1}) - Q(x_t, a),$$

где γ – дисконтирующий множитель.

Обновление *q*-функции для каждого активного правила $j \in J_x$ выглядит следующим образом:

$$q_{t+1}(L_j, o_j^l) = q_t(L_j, o_j^l) + \beta \alpha_j(x_t) \Delta Q.$$

Компьютерное моделирование

Сравнение эффективности RL-обучения на основе темпоральных различий с применением нечеткой логики проводилось с описанными в [2] методами RL-обучения. С применением ИНС DQN, DDQN и AC (Actor-Critic) были реализованы и испытаны соответствующие алгоритмы глубокого RL-обучения [2, 9, 10]. Для тестирования методов выбраны хорошо известные задачи, такие как перевернутый маятник (CartPole) и задача о горном автомобиле (Mountain Car).

CartPole – это модель среды (рис. 2), в которой можно управлять тележкой. По ее центру с помощью шарнира прикреплен шест (маятник). Тележка перемещается по горизонтальному рельсу. Силы трения и сопротивления отсутствуют. Шест в момент начала симуляции (игры) находится в вертикальном положении. Цель игры – предотвратить падение шеста. Это достигается за счет изменения скорости тележки в результате приложения к тележке горизонтальной силы. Возможные действия: толкнуть тележку влево или вправо. Игра заканчивается, если маятник отклонился от вертикальной оси более чем на 12°.

Среда задается состоянием, действием, наградой, начальным состоянием и флагом завершения эпизода.

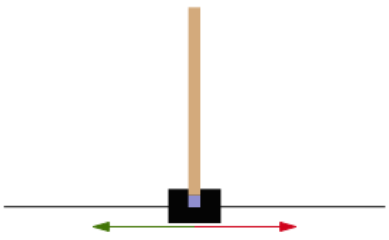


Рис. 2. Перевернутый маятник

Fig. 2. Inverted pendulum

Состояние описывается следующими величинами:

– позиция тележки – значение в диапазоне [-4.8, 4.8];

– скорость тележки;

– угол отклонения шеста от вертикали – значение в диапазоне [-24°, 24°], (-0.204, 0.204) в радианах;

– скорость изменения угла наклона шеста.

Действие может принимать значения 0 и 1:

0 – толкнуть тележку влево (приложить к ней горизонтальную силу, равную +1);

1 – толкнуть тележку вправо (приложить к ней горизонтальную силу, равную -1).

Награда на каждом шаге равна 1, при падении стержня – награда -1.

Начальное состояние задается как [0,0,0,0].

Случаи завершения эпизода:

– угол шеста вышел из диапазона [-12°, 12°];

– позиция тележки вышла из допустимого диапазона [-2.4, 2.4];

– длина эпизода >500.

Mountain Car – модель среды (рис. 3), в которой агент должен управлять автомобилем, чтобы преодолеть гору. Задача заключается в том, чтобы перевезти автомобиль на вершину горы, используя газ и рулевое управление. Возможным решением является последовательность действий: отойти от цели, немного подняться на противоположный склон, въезжать на склон с некоторой начальной скоростью.

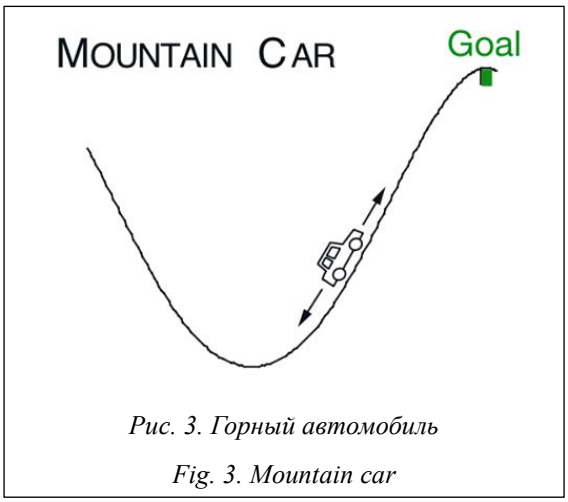


Рис. 3. Горный автомобиль

Fig. 3. Mountain car

Среда MountainCar описывается рядом компонентов.

Состояние среды определяется двумя величинами:

– позиция автомобиля: это значение указывает положение автомобиля на горизонтальной оси (диапазон значения – от -1.2 до 0.6);

– скорость автомобиля: это скорость движения автомобиля по горизонтальной оси (диапазон значения – от -0.07 до 0.07).

Агент может выбирать одно из трех действий:

– двигаться влево: автомобиль будет применять газ и двигаться влево;

– ничего не делать: автомобиль не изменяет скорость;

– двигаться вправо: автомобиль будет применять газ и двигаться вправо.

Награда: на каждом шаге агент получает награду -1. Цель агента – въехать на вершину горы, по достижении вершины горы агент получает награду 0. Достижение вершины горы считается успешным завершением задачи.

Начальное состояние задается следующим образом: позиция автомобиля – случайное значение из интервала [-0.6, -0.4], скорость автомобиля – 0.

Случаи завершения эпизода:

– достигнута вершина горы – позиция автомобиля >0.5;

– длина эпизода >200.

Был реализован алгоритм Q-обучения с нечеткой логикой (FQL). Для сравнительного анализа использовались DRL-методы DQN, DDQN и AC, реализованные в [8]. Приведем нечеткие правила для задач CartPole и MountainCar.

Нечеткие правила для задачи перевернутого маятника:

```
X = InputstateVariable
        (Trapeziums(-2,-1.2,-1.2,-0,775),
        Trapeziums(-1.2,-0.775,-0.775,-0.35),
        Trapeziums(-0.775,-0.35,-0.35,0.075),
        Trapeziums(-0.35,0.075,0.075,0.5),
        Trapeziums(0.75,0.5,0.5,0.5))
V = InputstateVariable
        (Trapeziums(-0.07,-0.07,-0.07,-0.035),
        Trapeziums(-0.07,-0.035,-0.035,0.),
        Trapeziums(-0.035,0.,0.,0.035)
        Trapeziums(0.,0.035,0.035,0.07),
        Trapeziums(0.035,0.035,0.035,0.07))
theta = InputstateVariable
        (Trapeziums(-0.07,-0.07,-0.07,-0.035),
        Trapeziums(-0.07,-0.035,-0.035,0.),
        Trapeziums(-0.035,0.,0.,0.035),
        Trapeziums(0.,0.035,0.035,0.07),
        Trapeziums(0.035,0.035,0.035,0.07))
theta_v = InputstateVariable
        (Trapeziums(-0.07,-0.07,-0.07,-0.035),
        Trapeziums(-0.07,-0.035,-0.035,0.),
        Trapeziums(-0.035,0.,0.,0.035),
        Trapeziums(0.,0.035,0.035,0.07),
        Trapeziums(0.035,0.035,0.035,0.07))
```

Нечеткие правила для задачи горного автомобиля:

```
P = InputStateVariable
                (Trapeziums(-1.3,-1.3-1.2,-0.9),
                Trapeziums(-0.9,-0.8,-0.6,-0.4),
                Trapeziums(-0.6,-0.4,-0.1,0.1),
                Trapeziums(-0.2,0,0.2,0.6),
                Trapeziums(0.4,0.6,0.8,1.3))
V = InputStateVariable
                (Trapeziums(-0.08,-0.08,-0.07,-0.04),
                Trapeziums(-0.05,-0.03,-0.01,0.1),
                Trapeziums(-0.02,0.,0.,0.02),
                Trapeziums(0.,0.02,0.03,0.05),
                Trapeziums(0.04,0.07,0.08,0.08))
```

Графики зависимости среднего вознаграждения, полученного агентом, от номера эпизода в задаче CartPole представлены на рисунке 4. Здесь и далее предполагается, что по оси абсцисс отложен номер эпизода, по оси ординат – количество набранных очков.

Из графика на рисунке 4 можно сделать вывод, что алгоритм с применением нечеткой логики FQL лучше справляется с задачей перевернутого маятника, чем DRL-методы DQN, DDQN и AC, которые также показывают достаточно высокую эффективность. Нечеткая логика позволила за довольно малое число эпизодов (127) достичь высокой награды в 497 очков (максимально возможная награда 500 очков).

На рисунке 5 представлены графики зависимости среднего вознаграждения от номера эпизода в задаче горного автомобиля. Для этой задачи алгоритм FQL за малое число эпизодов (116) достиг награды -155, в то время как алгоритму DDQN потребовалась 1 000 эпизодов для достижения награды -158, а алгоритм AC не справился с задачей. Однако в некоторых эпизодах FQL не смог справиться с задачей (например, при награде -200), что указывает на недостаточное описание системы нечеткими правилами. Изменение набора нечетких правил позволяет улучшить результаты. Реализация алгоритмов выполнена в среде Google Collab с использованием Python 3.10.

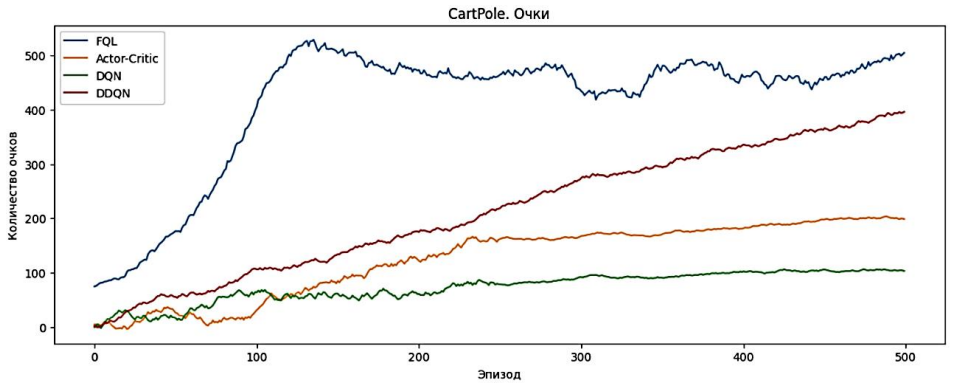


Рис. 4. Количество набранных очков в задаче перевернутого маятника

Fig. 4. Number of points scored in an inverted pendulum problem

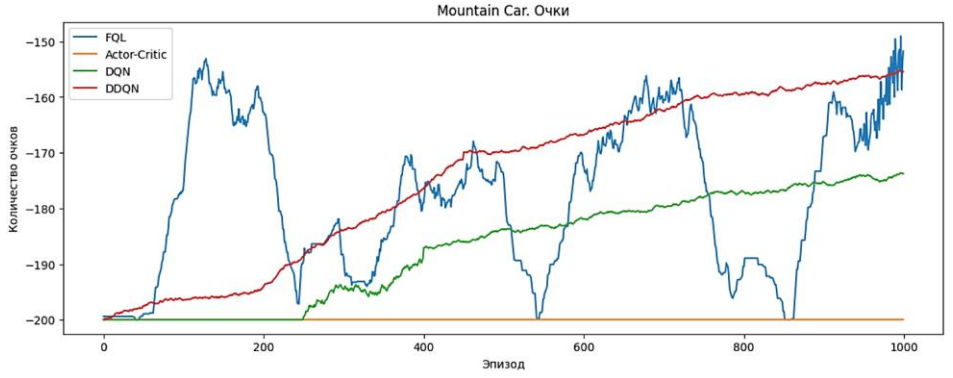


Рис. 5. Количество набранных очков в задаче горного автомобиля

Fig. 5. Number of points scored in a mountain car challenge

Заключение

В статье рассмотрен метод RL-обучения на основе TD-методов с применением нечеткой логики. Проведен сравнительный анализ алгоритма глубокого обучения с подкреплением с применением нечеткой логики FQL с алгоритмами на основе DRL-методов DQN, DDQN и AC. Проанализированы результаты работы методов обучения с подкреплением с применением нечеткой логики при компьютерных испытаниях реализованных на их основе алгоритмов обучения.

Можно отметить основные достоинства алгоритмов обучения с подкреплением с применением нечеткой логики.

Эффективность обучения: такие алгоритмы требуют меньшего количества эпизодов, поскольку обобщают знания из небольшого набора обучающих примеров. Это особенно важно, когда доступность данных для обучения ограничена или обучение в реальном времени требует быстрой адаптации.

Устойчивость к шуму: алгоритмы устойчивы к шуму и выбросам в данных, поскольку нечеткая логика учитывает их вариации и осуществляет обработку более гибко. Это важно в реальных средах, где существуют шумы или изменения в данных.

Интерпретируемость: алгоритмы с нечеткой логикой предоставляют интерпретируемые правила и выводы на основе нечеткой логики. Алгоритм позволяет понять, какие правила и механизмы лежат в основе принятия решений моделью.

Кроме того, расширяется область применения Q-обучения: нечеткая логика позволяет использовать его для задач с непрерывным пространством состояний.

В настоящее время рассматривается возможность интеграции методов RL-обучения (на основе Q- и TD-методов) с нейронными сетями в плане как использования RL-обучения для формирования обучающей выборки для нейронных сетей глубокого обучения, так и применения нейронных сетей при построении функций активации для Q-обучения.

Список литературы

1. Еремеев А.П., Кожухов А.А. Реализация методов обучения с подкреплением на основе темпоральных различий и мультиагентного подхода для интеллектуальных систем реального времени // Программные продукты и системы. 2017. Т. 30. № 1. С. 28–33. doi: 10.15827/0236-235X.117.028-033.

2. Еремеев А.П., Сергеев М.Д. Интеграция методов обучения с подкреплением и искусственных нейронных сетей // Международный научно-технический конгресс ИС & ИТ-2022: тр. конгресса. 2022. С. 10–21.

3. Саттон Р.С., Барто Э.Г. Обучение с подкреплением; [пер. с англ]. М.: ДМК Пресс, 2020. 552 с.

4. Осипов Г.С. Методы искусственного интеллекта. М.: Физматлит, 2011. 296 с.

5. Chen Z., Deng S., Chen X., Li C., Sanchez R.-V., Qin H. Deep neural networks-based rolling bearing fault diagnosis. Microelectronics Reliability, 2017, vol. 75, pp. 327–333. doi: 10.1016/j.microrel.2017.03.006.

6. Hsu C.-C., Lin C.-W. CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia, 2017, vol. 20, no. 2, pp. 421–429. doi: 10.1109/TMM.2017.2745702.

7. Glorennec P.Y., Jouffe L. Fuzzy q-learning. Proc. Int. Fuzzy Systems Conf., 1997, vol. 2, pp. 659–662. doi: 10.1109/FUZZY.1997.622790.

8. Masoumzadeh S.S., Hlavacs H., Tomás L. A self-adaptive performance-aware capacity controller in overbooked datacenters. Proc. ICCAC, 2016, pp. 183–195. doi: 10.1109/ICCAC.2016.8.

9. François-Lavet V., Henderson P., Islam R., Bellemare M.G., Pineau J. An introduction to deep reinforcement learning. Foundations and Trends in Machine Learning, 2018, vol. 11, no. 3-4, pp. 219–354. doi: 10.1561/2200000071.

10. Mnih V., Kavukcuoglu K., Silver D., Rusu A.A., Veness J., Bellemare M.G. et al. Human-level control through deep reinforcement learning. Nature, 2015, vol. 518, no. 7540, pp. 529–533. doi: 10.1038/nature14236.

Software & Systems doi: 10.15827/0236-235X.142.600-606 2023, vol. 36, no. 4, pp. 600–606

Integration of reinforcement learning methods and fuzzy logic for intelligent real-time systems

Aleksandr P. Eremeev
Maxim D. Sergeev
Vladimir S. Petrov

For citation
Eremeev, A.P., Sergeev, M.D., Petrov, V.S. (2023) 'Integration of reinforcement learning methods and fuzzy logic for intelligent real-time systems', *Software & Systems*, 36(4), pp. 600–606 (in Russ.). doi: 10.15827/0236-235X.142.600-606

Article info
Received: 03.07.2023 After revision: 13.07.2023 Accepted: 14.07.2023

Abstract. The paper considers the possibilities of integrating reinforcement learning methods and fuzzy logic in terms of improving the efficiency of reinforcement learning algorithms. The main focus is on applying such integrated methods in real-time intelligent systems, especially in decision support systems for monitoring and controlling complex technical objects. The basis is a reinforcement learning method based on temporal differences; an environmental state and the reward signal are formed using fuzzy logic. The paper presents software implementation and the data of computer simulation of deep learning methods with reinforcement based on temporal differences obtained from a comparative analysis of a fuzzy logic algorithm and neural network algorithms. The authors of the paper show that the main advantages of reinforcement learning algorithms using fuzzy logic are: training effectiveness in terms of minimizing the number of episodes, which is especially important regarding the availability of limited data for training or rapid adaptation of real-time training; resistance to noise and outliers in data, which is important in real environments with noise or data changes; interpretability - fuzzy logic algorithms provide interpretable rules and conclusions based on fuzzy logic; extension of a reinforcement learning scope to subject/problem areas and tasks with a continuous state space. These researches and developments are carried out in terms of designing intelligent real-time decision support systems. Such systems are designed to help operational dispatch personnel (decision makers) in monitoring and managing complex technical and organizational systems under fairly severe time constraints and various types of uncertainties (inaccuracies, fuzziness, inconsistencies) in the information entering a system, i.e. in the presence of so-called noisy data.
Keywords: artificial intelligence, reinforcement learning, fuzzy logic, intelligent system, real time, decision support

References

1. Eremeev, A.P., Kozhukhov, A.A. (2017) ‘Implementation of reinforcement learning methods based on temporal differences and a multi-agent approach for real-time intelligent systems’, *Software & Systems*, 30(1), pp. 28–33 (in Russ.). doi: 10.15827/0236-235X.117.028-033.

2. Eremeev, A.P., Sergeev, M.D. (2022) ‘Integration of training methods with reinforcement and artificial neural networks’, *Proc. Int. Congress "IS & IT-2022"*, pp. 10–21 (in Russ.).

3. Sutton, R.S., Barto, A.G. (2018) *Reinforcement Learning*. MA, Cambridge: The MIT Press, 552 p. (Russ. ed.: Moscow, 2020, 552 p.).

4. Osipov, G.S. (2015) *Methods of Artificial Intelligence*. Moscow, 296 p (in Russ.).

5. Chen, Z., Deng, S., Chen, X., Li, C., Sanchez, R.-V., Qin, H. (2017) ‘Deep neural networks-based rolling bearing fault diagnosis’, *Microelectronics Reliability*, 75, pp. 327–333. doi: 10.1016/j.microrel.2017.03.006.

6. Hsu, C.-C., Lin, C.-W. (2017) ‘CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data’, *IEEE Transactions on Multimedia*, 20 (2), pp. 421–429. doi: 10.1109/TMM.2017.2745702.

7. Glorennec, P.Y., Jouffe, L. (1997) ‘Fuzzy q-learning’, *Proc. Int. Fuzzy Systems Conf*., 2, pp. 659–662. doi: 10.1109/FUZZY.1997.622790.

8. Masoumzadeh, S.S., Hlavacs, H., Tomás, L. (2016) ‘A self-adaptive performance-aware capacity controller in overbooked datacenters’, *Proc. ICCAC*, pp. 183–195. doi: 10.1109/ICCAC.2016.8.

9. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J. (2018) ‘An introduction to deep reinforcement learning’, *Foundations and Trends in Machine Learning*, 11(3-4), pp. 219–354. doi: 10.1561/2200000071.

10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G. et al. (2015) ‘Human-level control through deep reinforcement learning’, *Nature*, 518 (7540), pp. 529–533. doi: 10.1038/nature14236.

Авторы
Еремеев Александр Павлович [1], д.т.н., профессор, eremeev@appmat.ru
Сергеев Максим Дмитриевич [1], аспирант, SergeevMD@mpei.ru
Петров Владимир Сергеевич [1], студент , PetrovVS@mpei.ru

[1] Национальный исследовательский университет «Московский энергетический институт», г. Москва, 111250, Россия

Authors
Aleksandr P. Eremeev [1], Dr.Sc. (Engineering), Professor, eremeev@appmat.ru
Maxim D. Sergeev [1], Postgraduate Student, SergeevMD@mpei.ru
Vladimir S. Petrov [1], Student, PetrovVS@mpei.ru

[1] National research University "MPEI", Moscow, 111250, Russian Federation

УДК 004.272.26 + 519.157.2 doi: 10.15827/0236-235X.142.607-614 2023. Т. 36. № 4. С. 607–614

Использование библиотеки MPI для параллельной реализации алгоритма полного перебора вариантов

Ай Мин Тайк
С.А. Лупин
Д.А. Федяшин

Ссылка для цитирования
Ай Мин Тайк, Лупин С.А., Федяшин Д.А. Использование библиотеки MPI для параллельной реализации алгоритма полного перебора вариантов // Программные продукты и системы. 2023. Т. 36. № 4. С. 607–614. doi: 10.15827/0236-235X.142.607-614
Информация о статье
Поступила в редакцию: 02.10.2023 После доработки: 16.10.2023 Принята к публикации: 18.10.2023

Аннотация. В работе представлены результаты исследования эффективности параллельного приложения для решения задачи квадратичного назначения. Приложение использует алгоритм полного перебора и интерфейс передачи сообщений (MPI) для организации взаимодействия процессов в распределенной вычислительной среде. Для генерации вариантов решений использован метод лексикографической перестановки, который хорошо распараллеливается и обеспечивает балансировку нагрузки узлов. Для расширения диапазона использования алгоритма перебора на оптимизационные задачи большой размерности пространство поиска решений разбивается на части, что позволяет существенно сократить число рассматриваемых вариантов. Эксперименты проводились в распределенной среде, содержащей 50 узлов с процессорами Intel ® Core™ i5 7-го поколения. Параллельное приложение при решении задачи большой размерности (n = 24) продемонстрировало достижимость ускорения вычислений в диапазоне от 99 до 95 % от максимально возможного, причем во всех случаях было найдено точное решение. Это подтверждает корректность методов, использованных для распределения нагрузки и снижения вычислительной сложности задач. Дальнейшая работа будет направлена на исследование возможности применения предлагаемых подходов при реализации алгоритма перебора и в виде гибридных приложений, что актуально для гетерогенных вычислительных сред. Поскольку задача квадратичного назначения относится к задачам дискретной оптимизации, полученные результаты дают основание считать, что предлагаемые решения будут эффективными и для других задач этого класса.

Ключевые слова: дискретная оптимизация, задача квадратичного назначения, алгоритм полного перебора, распределенные вычисления, MPI, HPC

Благодарности. Исследование проводилось в МИЭТ и МСЦ РАН в рамках государственного задания по теме FNEF-2022-0016

Введение. Люди постоянно сталкиваются с задачами комбинаторной оптимизации, даже не осознавая этого. Большинство систем управления основано на том, что оператор или автомат формирует оптимальное воздействие, выбирая его из некоторого конечного набора возможных решений. Задачи дискретной оптимизации находят широкое применение и в системах автоматизированного проектирования.

В данной статье рассмотрены особенности решения в параллельной среде одной из таких задач – задачи квадратичного назначения (QAP, Quadratic Assignment Problem), которая является характерным представителем этого класса. Она относится к NP-сложным, требующим значительных вычислительных ресурсов. Примерами проблем, формализуемых в виде задачи квадратичного назначения, являются задачи расстановки машин в обрабатывающих центрах, размещения библиотечных элементов в матричных БИС и микросхем на коммутационных платах [1].

Универсальным подходом к решению задач дискретной оптимизации, в частности, и QAP, является алгоритм полного перебора вариантов (BFA, Brute Force Algorithm). Алгоритм BFA методично исследует пространство поиска решений, оценивая все возможные варианты для нахождения точного решения задачи [2]. При этом для перебора вариантов часто используют методы перестановок.

Из-за высокой вычислительной сложности задачи комбинаторной оптимизации, основанные на перестановках, реализуются в распределенных вычислительных средах [3]. Перестановки позволяют проводить поиск наилучшего расположения элементов в задачах, где важны порядок и неповторяемость элементов вектора решения. Методы перестановок реализуются и в распределенных вычислениях. Для взаимодействия узлов вычислителя применяют MPI, стандартный коммуникационный протокол, обеспечивающий обмен информацией между процессорами в распределенной среде [4]. Параллельная реализация BFA позволяет увеличить размер решаемых задач и сократить время вычислений.

Алгоритмические подходы к решению QAP

Рассмотрим несколько известных алгоритмических подходов к параллельной реализации решателей для QAP.

Последовательные алгоритмы эффективны лишь для решения задач небольшой размерности. Для более крупных точные решения могут быть найдены только с использованием методов параллельных вычислений (OpenMP, MPI, Hybrid и CUDA).

В работе [5] представлена комбинаторная оптимизация NP-трудной задачи, приведенной к QAP, с помощью двух параллельных моделей (embarrassingly parallel algorithm и island parallel genetic algorithm) на GPU.

В статье [6] продемонстрировано решение задачи квадратичного назначения с использованием параллельного гибридного алгоритма. Результаты в среднем находятся в пределах 0,05 % от наилучших решений, приведенных в QAPLIB.

В работе [7] при решении QAP использованы графические ускорители. Качество решений, полученных с помощью реализации алгоритмов 2-opt и tabu search, находится в пределах 1,03 % и 0,15 % от известных значений.

Эксперименты дают представление о рабочих характеристиках ускоренных QAP-решателей. В исследовании [8] представлен алгоритм Cooperative Parallel Tabu Search (CPTS) для решения QAP. В CPTS процессоры обмениваются информацией на протяжении всего выполнения алгоритма в отличие от независимых стратегий параллельного поиска, которые агрегируют данные только в конце выполнения. Результаты исследований демонстрируют преимущества параллельных вычислений с точки зрения качества решения, вычислительных затрат и алгоритмической гибкости.

В работе [9] представлен параллельный алгоритм для решения QAP путем включения элементов имитационного отжига в подход локального поиска, способствующий достижению оптимальных результатов. Они сравниваются с результатами множества других алгоритмов, использующих различные экземпляры из набора данных QAPLIB.

Исследование [10] демонстрирует эффективность использования процедуры жадного случайного адаптивного поиска (GRASP) при решении QAP, что в нескольких случаях приводит к почти оптимальным результатам или к достижению оптимальности. Авторы разработали последовательную программу на языке Си, а впоследствии параллельную реализацию для эффективного снижения затрат процессорного времени.

Разработаны и решения задачи QAP с использованием перестановок для поиска экстремума целевой функции. В статье [11] представлен параллельный алгоритм для решения QAP с помощью многопоточного приложения. Для задач размерностью 12 и 14 с помощью перестановок проведен исчерпывающий поиск в пространстве решений. Предложенным методом эти перестановки равномерно распределяются между процессорными ядрами, что позволяет сократить время параллельного выполнения за счет использования шести потоков.

В данной работе исследования направлены на оценку возможности использования результатов, полученных для многопоточных приложений при решении задач комбинаторной оптимизации, на случай распределенных систем, применяющих MPI.

Простейший способ распараллеливания решения проблемы заключается в распределении набора перестановок по различным вычислительным узлам в распределенной вычислительной среде. При этом узлы независимо выполняют вычисления целевой функции для локальных перестановок. Такая стратегия распараллеливания побуждает каждый узел искать свое оптимальное локальное решение. По завершении вычислений на всех узлах выбирается наилучшее решение из всех локальных, которое и будет глобальным экстремумом.

Особенности реализации параллельных вычислений для QAP

Для решения задач QAP большого размера, требующих значительных вычислительных ресурсов, часто применяют методы параллельных вычислений. Такие вычисления могут помочь ускорить процесс решения за счет распределения рабочей нагрузки между несколькими процессорами или узлами, тем самым используя возможности современных высокопроизводительных вычислительных систем. Известны несколько подходов к организации параллельных вычислений, которые могут быть применены и для QAP.

Параллельный локальный поиск. Многие эвристические алгоритмы, такие как имитированный отжиг или генетический алгоритм, могут быть распараллелены путем одновременного запуска нескольких экземпляров алго-

ритма с различными исходными решениями или параметрами [12].

Параллельный метод ветвей и границ. Это широко используемый метод для нахождения точных решений [13]. Распараллеливание построения дерева поиска может привести к значительному ускорению. Узлы дерева поиска могут быть распределены между различными процессорами.

Решатели для целочисленного программирования. Многие коммерческие решатели для целочисленного программирования с открытым исходным кодом, такие как CPLEX и Gurobi, поддерживают параллельное выполнение [14]. Они могут быть использованы и для решения QAP в виде задач смешанного целочисленного программирования.

Алгоритм искусственной пчелиной колонии. Он может решать оптимизационные задачи, такие как QAP. В статье [15] представлена параллельная версия алгоритма для архитектур с разделяемой памятью.

Алгоритм перебора. Ориентированный на системы с общей памятью, он может быть распараллелен с помощью OpenMP, что особенно эффективно для многоядерных процессоров. В работе [16] проведено сравнение последовательной и параллельной реализаций данного алгоритма с использованием OpenMP для решения QAP. Показано, что производительность многопоточного приложения при запуске на четырехъядерном процессоре в 2,3 раза выше, чем последовательного.

Задачу QAP можно разделить на мелкие подзадачи, каждая из них может быть решена независимо на разных процессорах с использованием библиотек передачи сообщений, таких как MPI.

Параллельные гибридные алгоритмы. Они объединяют различные методы решения (эвристические и точные) в параллельной среде, чтобы использовать сильные стороны обоих методов.

Мощные параллельные графические процессоры (GPU) позволяют ускорить определенные вычисления, особенно те, которые связаны с матричными операциями, распространенными в алгоритмах QAP [17].

Важно отметить, что эффективность параллельных вычислений для QAP зависит от различных факторов, включая размер задачи, используемый алгоритм, доступное аппаратное обеспечение и эффективность методов распараллеливания. Параллельные вычисления могут обеспечить значительное ускорение и в то же время привести к возникновению проблем, связанных с балансировкой нагрузки, накладными расходами на связь и синхронизацией. При реализации параллельных алгоритмов для QAP следует учитывать множество факторов: возможность распараллеливания алгоритма, архитектуру доступных вычислительных ресурсов, балансировку нагрузки узлов для достижения наилучшей производительности. Далее представлены результаты исследования эффективности многопоточного приложения, реализующего BFA и использующего библиотеку с MPI, для решения QAP в распределенной вычислительной среде.

Особенности параллельной реализации BFA

Алгоритм полного перебора [18] осуществляет оценку всех возможных перестановок объектов, распределяя их в случае параллельной реализации между процессами с помощью библиотеки MPI.

Параллельная реализация BFA для QAP включает следующие процедуры.

Настройка параметров: задается размерность матрицы связности элементов R и матрицы расстояний D.

Настройка среды и инициализация MPI.

Генерация перестановок: распределяются перестановки между доступными MPI процессами и реализуется метод *следующей лексикографической перестановки* для генерации вариантов решения P.

Вычисление целевой функции: каждый процесс вычисляет целевую функцию для всех перестановок в назначенном ему подмножестве, фиксируя текущий рекорд [16].

Поиск и сравнение конечного результата: ведущий процесс получает со всех узлов локальные минимумы и выбирает из них глобально оптимальное решение.

Завершение MPI и вывод: процесс с rank = 0 выводит решение и затраченное на вычисления время.

В экспериментах использованы два варианта QAP-задачи – размерностью $n = 14$ и $n = 24$. Соответствующие матрицы R и D представлены на сайте (http://www.swsys.ru/uploaded/image/2023-4/5.jpg).

Пространство решения первой задачи было разделено на две части: $n_1 = 9$, $n_2 = 5$, а второй – на четыре: $n_1 = n_2 = n_3 = n_4 = 6$, руководствуясь описанным в [16] методом снижения вычислительной сложности QAP. Число анализируе-

мых вариантов в первом случае составит $P \approx 4{,}4 \cdot 10^7$, а во втором $P \approx 1{,}4 \cdot 10^{17}$.

В описываемом эксперименте использована 64-разрядная версия MPICH2 для реализации MPI. Этот стандарт с открытым исходным кодом (https://www.mpich.org) широко используется для программирования параллельных и распределенных вычислительных систем.

Приведенный далее фрагмент кода иллюстрирует подход к распараллеливанию BFA, применяемый при решении задачи QAP. Приложение реализует генерацию вариантов решения с помощью метода следующей лексикографической перестановки. Программа написана на языке C++ в среде Visual Studio:

```
   1: void reverseOrder(int*p,
int start,int end){
   2: while(start<end)
{swap(p[start],p[end]); start++;end--
;}}
   //генерация функции перестановки
   1:bool permutationfun(int*p, int n)
{int k=-1;
   2:for (i=n-2;i≥0;i--)
{if(p[i]<p[i+1]) {k=i; break;}}
   3: if (k==-1) return false;
int m=n-1;while (p[k]≥p[m]) m--;
   4: swap(p[k],p[m]);
reverseOrder(p,k+1,n-1); return true;}
   // вычислительная функция для каждой
перестановки
   1: int calfun(int*p,int**D,int**R,
int n,int m,int localmin) {calvalue=0;
   2: for i=0 to n do {for j=i+1 to n
do {calvalue+=R[i][j]*D[p[i]][p[j]];}}
   3: if(localmin≥calvalue)localmin=
=calvalue; calvalue=0;
   4: bool hasnext=true; while(hasnext)
{int k=-1;
   5: for (i=m-2; i≥0;i--)
{if(p[i]<p[i+1]) {k=i;break;}}
   6: if (k==-1) hasnext=false;
else {int m=n-1;while(p[k]≥p[m])m--;
   7: swap(p[k],p[m]);
reverseOrder(p,k+1,m-1);
   8: for k=0 to n do {for h=k+1 to n do
{calvalue+=R[k][h]*D[p[k]][p[h]];}}
   9: if(localmin≥calvalue)
localmin=calvalue; return localmin;}
   // основная функция
   1: int main(int argc,char* argv[])
{MPI_Init(&argc,&argv);
   2: MPI_Comm_rank(MPI_COMM_WORLD,
&rank);
   3: MPI_Comm_size(MPI_COMM_WORLD,
&size);
   4: int rank,size,
localmin=5000,calvalue=0;count=0;
double startt,endt;
   5: int ptwo[]={},pthree[]={},
pfour[]={};
   6: int p[]={};
   7: int D[n][n],R[n][n];assign cost
matrix D and distance matrix R;
   8: startt=MPI_Wtime();do{if
(count%size==rank){sort(pthree,
pthree+m);
   9: do{sort(pfour,pfour+m);
do{sort(p,p+m);for(int k=0;k<m;k++){
   10: p[m+k]=ptwo[k];p[(m*2)+k]=
=pthree[k];p[(m*3)+k]=pfour[k];}
   11: localmin=calfun(p,D,R,
localmin);}while(permutationfun
(pfour,m));
   12: }while(permutationfun
(pthree,m));}count++;
while(permutationfun(ptwo,m));
   13: endt= MPI_Wtime();
intglobalmin=0;
   14: MPI_Reduce(&localmin,&globalmin,
1,MPI_INT,MPI_MIN,0,MPI_COMM_WORLD);
   15: if(rank==0)
{cout≪globalmin≪"\t"≪endt-startt;}
MPI_Finalize();return 0;}
```

При решении QAP с использованием параллельного алгоритма BFA библиотека MPI обеспечивает эффективное распределение перестановок и определение глобального оптимума полученных процессами локальных решений.

При параллельном решении QAP нагрузка каждого процесса определяется через его ранг, что позволяет эффективно распределять перестановки и балансировать нагрузку узлов распределенной среды. Функция MPI_Reduce используется для сбора результатов и нахождения глобального оптимума, а MPI_Wtime – для измерения времени выполнения различных частей алгоритма.

Представленный фрагмент программного кода реализует принцип разделения пространства поиска на ограниченные части, описанный в [16]. Для $n = 24$ таких частей четыре, что позволяет разделить нагрузку между процессами, опираясь только на число перестановок в первой части. В рассматриваемом случае оно составляет 720 и определяет максимальное число параллельных процессов. Очевидно, что максимальную эффективность использования вычислительных ресурсов среды можно получить, если это число будет нацело делиться на число запущенных процессов. Для задачи размерностью $n = 14$ пространство поиска можно разделить двумя способами: используя или $n_1 = 9$, $n_2 = 5$, или $n_1 = 5$, $n_2 = 9$.

Генерация вариантов решения с помощью метода следующей лексикографической перестановки гарантирует их неповторяемость для всех процессов.

Такой подход позволяет не только с минимальными накладными расходами организовать параллельные вычисления, но и повысить эффективность распределенной реализации BFA.

Необходимо отметить, что используемый метод распределения нагрузки гарантирует ее равномерность только в том случае, если сложность вычисления критериальной функции не зависит от параметров вектора решения. Это условие выполняется, если n-мерное пространство поиска изотропно по отношению к целевой функции. Для решаемой задачи квадратичного назначения это условие выполняется, если не используются специальные методы. В проводимых вычислениях вопросы оптимизации процесса вычисления целевой функции не рассматривались.

В ходе проведенных экспериментов корректность выбранного метода распределения нагрузки нашла практическое подтверждение. Время работы всех узлов независимо от исследуемого участка пространства поиска решений не отличалось. Аналогичные результаты были получены и для многопоточной реализации BFA [16].

Результаты экспериментов

Для оценки эффективности предлагаемых подходов к параллельной реализации BFA были использованы две QAP-задачи размерностью $n = 14$ и $n = 24$. Условия задач представлены в таблицах (http://www.swsys.ru/uploaded/image/2023-4/5.jpg).

Минимальное значение целевой функции F равно 46 для $n = 14$ и 301 для $n = 24$. Это точные решения, которые были получены для неограниченного пространства поиска.

В таблицах 1 и 2 представлены варианты найденных решений задачи. Для всех решений значение функционала совпадает с известным точным решением, что подтверждает корректность метода разделения пространства поиска. Курсивом выделены номера занимаемых элементами мест.

В таблицах 3 и 4 представлены результаты исследования масштабируемости и эффективности параллельного приложения. Вычислительные эксперименты проводились в компьютерном классе, оснащенном персональными компьютерами с процессором Intel® Core™ i5 7-го поколения, у которых количество физических ядер равно 4, а тактовая частота 3,00 ГГц. В таблицах использованы обозначения: t_{calc} – время вычисления, сек.; N_{proc} – число работающих процессов (использовались только физические ядра); *Acc* – полученное ускорение; *Eff* – эффективность параллельной реализации или коэффициент использования вычислительных ресурсов: $Eff = (t_{calc}(1)/t_{calc}(N_{proc}))\cdot(1/N_{proc}) = Acc\cdot(1/N_{proc})$.

Таблица 1

Решения для $n = 14$

Table 1

Solutions for $n = 14$

Расположение элементов в векторе решения *P*														***F***
1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	**46**
1	4	7	2	5	8	3	6	9	10	11	12	13	14	
1	2	3	4	5	6	7	8	9	10	11	12	14	13	
1	4	7	2	5	8	3	6	9	10	11	12	14	13	
1	2	3	4	5	6	7	8	9	12	11	10	13	14	
1	4	7	2	5	8	3	6	9	12	11	10	13	14	
1	2	3	4	5	6	7	8	9	12	11	10	14	13	
1	4	7	2	5	8	3	6	9	12	11	10	14	13	

Таблица 2

Решение для $n = 24$

Table 2

Solutions for $n = 24$

Расположение элементов в векторе решения *P*												***F***
1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	
5	2	1	4	3	6	9	8	11	10	7	12	**301**
13	***14***	***15***	***16***	***17***	***18***	***19***	***20***	***21***	***22***	***23***	***24***	
17	14	15	16	13	18	19	24	20	22	23	21	

Таблица 3

Время решения задачи для $n = 14$

Table 3

Problem solution time for $n = 14$

N_{proc}	1	2	3	4	5	6	7	8
t_{calc}	11,48	6,01	4,11	3,08	2,68	2,25	2,36	1,95
Acc	-	1,91	2,79	3,72	4,28	5,10	4,86	5,89
Eff	-	0,96	0,93	0,93	0,86	0,85	0,7	0,74

Таблица 4

Время решения задачи для $n = 24$

Table 4

Problem solution time for $n = 24$

N_{proc}	1	45	90	180
t_{calc}	240213,62	5402,33	2713,76	1407,66
Acc	-	44,46	88,52	170,64
Eff	-	0,99	0,98	0,95

Для чистоты экспериментов при решении задачи с $n = 14$ использовано только по одному ядру на каждом узле.

Следует отметить, что для задачи с $n = 24$ полученные ускорение и коэффициент использования вычислительных ресурсов выше, чем в случае с $n = 14$. Это хорошо согласуется с результатами, полученными и в других работах [19], поскольку доля накладных расходов на синхронизацию процессов будет ниже при большем времени работы узлов. Количество процессов, задействованных в вычислениях для $n = 24$, выбиралось так, чтобы число перестановок в распараллеливаемой части (720) нацело делилось на него.

Заключение

Исследование параллельной реализации алгоритма перебора вариантов при решении QAP-задач позволяет сделать следующие выводы.

Библиотека MPI обеспечивает масштабируемость приложения: высокая эффективность сохраняется при росте числа задействованных вычислительных узлов распределенной среды.

Генерация вариантов решения с помощью метода следующей лексикографической перестановки хорошо распараллеливается и обеспечивает балансировку нагрузки узлов.

Разбиение пространства поиска решений на части позволяет расширить диапазон использования BFA и для оптимизационных задач большой размерности.

В дальнейшем планируется исследовать возможности использования предлагаемых подходов к реализации BFA и в виде гибридных приложений, что, на взгляд авторов, актуально для гетерогенных вычислительных сред. Гибридный параллелизм, использующий комбинацию MPI и OpenMP, позволит более эффективно балансировать нагрузку узлов, содержащих разное количество ядер.

Список литературы

1. Duman E., Or I. The quadratic assignment problem in the context of the printed circuit board assembly process. Comput. & Operations Research, 2007, vol. 34, no. 1, pp. 163–179. doi: 10.1016/j.cor.2005.05.004.

2. Тайк А.М., Лупин С.А., Кхаинг М.Т. Методы повышения эффективности алгоритма полного перебора на примере решения задачи о неограниченном ранце // INJOIT. 2023. Т. 11. № 5. С. 41–46.

3. Pérez-Rodríguez R. An estimation of distribution algorithm for combinatorial optimization problems. IJIO, 2022, vol. 3, no. 1, pp. 47–67. doi: 10.12928/ijio.v3i1.5862.

4. Bhagat L., Arora S., Shrawankar U. Practical Guide to Distributed Systems in MPI. Pothi.com Publ., 2017, 97 p.

5. Gomez E.C., Chaves R.P., Hurtado O.G. Combinatorial optimization NP-Hard problem solved by using the quadratic assignment problem (QAP) solution through a parallel genetic algorithm on GPU. Vision Electronica, 2017, vol. 11, no. 2, pp. 146–151.

6. Tosun U. On the performance of parallel hybrid algorithms for the solution of the quadratic assignment problem. Eng. Applications of Artificial Intelligence, 2015, vol. 39, pp. 267–278. doi: 10.1016/j.engappai.2014.12.011.

7. Sonuc E., Sen B., Bayir S. A cooperative GPU-based parallel multistart simulated annealing algorithm for quadratic assignment problem. JESTECH, 2018, vol. 21, no. 5, pp. 843–849. doi: 10.1016/j.jestch.2018.08.002.

8. Abdelkafi O., Derbel B., Liefooghe A. A parallel tabu search for the large-scale quadratic assignment problem. Proc. IEEE CEC, 2019, pp. 3070–3077. doi: 10.1109/CEC.2019.8790152.

9. Ozcetin E., Ozturk G. A parallel iterated local search algorithm on GPUs for quadratic assignment problem. IJET, 2018, vol. 4, no. 2, pp. 123–127.

10. Loranca B.B., Analco M.E., Velázquez R.G., López A.S. et al. Quadratic assignment problem: A solution approach with parallel GRASP. IJCOPI, 2017, vol. 8, no. 3, pp. 33–38.

11. Cardenas E.G., Poveda R.Ch., Garcia O.H. A solution for the quadratic assignment problem (QAP) through a parallel genetic algorithm based grid on GPU. Appl. Math. Sci., 2017, vol. 11, no. 57, pp. 2843–2854. doi: 10.12988/ams.2017.710319.

12. Codognet Ph., Munera D., Diaz D., Abreu S. Parallel Local Search. In: Handbook of Parallel Constraint Reasoning. Hamadi Y., Sais L. (eds), 2018, pp. 381–417. doi: 10.1007/978-3-319-63516-3_10.

13. Fujii K., Ito N., Kim S., Kojima M., Shinano Y., Toh K.-Ch. Solving Challenging Large Scale QAPs, ZIB Report, 2021, art. 21-02, pp. 1–15.

14. Ralphs T., Shinano Y., Berthold T., Koch T. Parallel solvers for mixed integer linear optimization. In: Handbook of Parallel Constraint Reasoning, 2018, pp. 283–336. doi: 10.1007/978-3-319-63516-3_8.

15. Li S., Li W., Zhang H., Wang Z. Research and implementation of parallel artificial bee colony algorithm based on ternary optical computer. Automatika, 2019, vol. 60, no. 4, pp. 422–431. doi: 10.1080/00051144.2019.1639118.

16. Тайк А.М., Лупин С.А., Балабаев С.А. Особенности применения алгоритма полного перебора для решения задачи квадратичного назначения // INJOIT. 2023. Т. 11. № 7. С. 60–68.

17. Kumar M., Mitra P. Solving quadratic assignment problem using iterated local search on GPU spatial memory, Proc. Int. Conf. ACIA, 2023, vol. 2705, no. 1, pp. 1–9. doi: 10.1063/5.0133414.

18. Thike A.M., Lupin S. Parallel application for wireless network topology optimisation. IJEEDC, 2019, vol. 7, no. 4, pp. 14–20.

19. Рыбаков А.А., Мещеряков А.О. Векторизация трехмерного метода погруженных границ для повышения эффективности расчетов на микропроцессорах Intel // Программные продукты и системы. 2023. Т. 36. № 1. С. 130–143. doi: 10.15827/0236-235X.141.130-143.

Software & Systems doi: 10.15827/0236-235X.142.607-614 2023, vol. 36, no. 4, pp. 607–614

Using MPI library for parallel implementation of a brute-force algorithm

Aye Min Thike
Sergey A. Lupin
Dmitry A. Fedyashin

For citation
Thike, A.M., Lupin, S.A., Fedyashin, D.A. (2023) 'Using MPI library for parallel implementation of a brute-force algorithm', *Software & Systems*, 36(4), pp. 607–614 (in Russ.). doi: 10.15827/0236-235X.142.607-614
Article info
Received: 02.10.2023 After revision: 16.10.2023 Accepted: 18.10.2023

Abstract. The paper presents results of studying the effectiveness of a parallel application for solving a quadratic assignment problem. The application uses a brute-force algorithm and a Message Passing Interface (MPI) to ensure interprocess communication in a distributed computing environment. Solution options are generated by a lexicographic permutation method, which is well parallelized and ensures load balancing of nodes. The range of brute-force algorithm use to large scale optimization problems is expanded by dividing a search space into parts, which significantly reduces the number of options under consideration. The experiments were conducted in a distributed environment containing 50 nodes with the 7th generation Intel ® Core ™ i5 processors. A parallel application for solving a large-scale problem (n = 24) demonstrated the possibility accelerating calculations in the range from 99 to 95 % from the maximum possible one; an exact solution was found in all cases. This confirms the correctness of methods used for load distribution and for computational complexity reduction of tasks. Further work will be aimed at exploring the possibility of using the proposed approaches to implementing a brute-force algorithm and as hybrid applications, which is relevant for heterogeneous computing environments. Since a quadratic assignment problem is a discrete optimization problem, the obtained results suggest that the proposed approaches are effective for other problems of this class.

Keywords: discrete optimization, quadratic assignment problem, brute-force algorithm, distributed computing, MPI, HPC
Acknowledgements. The study was conducted at MIET and JSCC RAS within the framework of the state assignment on the topic FNEF-2022-0016

References

1. Duman, E., Or, I. (2007) 'The quadratic assignment problem in the context of the printed circuit board assembly process', *Comput. & Operations Research*, 34(1), pp. 163–179. doi: 10.1016/j.cor.2005.05.004.

2. Thike, A.M., Lupin, S.A., Khaing, M.T. (2023) 'Methods for improving the efficiency of Brute-force algorithm by the example of solving an Unbounded Knapsack Problem', *INJOIT*, 11(5), pp. 41–46 (in Russ.).

3. Pérez-Rodríguez, R. (2022) 'An estimation of distribution algorithm for combinatorial optimization problems', *IJIO*, 3(1), pp. 47–67. doi: 10.12928/ijio.v3i1.5862.

4. Bhagat, L., Arora, S., Shrawankar, U. (2017) *Practical Guide to Distributed Systems in MPI*. Pothi.com Publ., 97 p.

5. Gomez, E.C., Chaves, R.P., Hurtado, O.G. (2017) 'Combinatorial optimization NP-Hard problem solved by using the quadratic assignment problem (QAP) solution through a parallel genetic algorithm on GPU', *Vision Electronica*, 11(2), pp. 146–151.

6. Tosun, U. (2015) 'On the performance of parallel hybrid algorithms for the solution of the quadratic assignment problem', *Eng. Applications of Artificial Intelligence*, 39, pp. 267–278. doi: 10.1016/j.engappai.2014.12.011.

7. Sonuc, E., Sen, B., Bayir, S. (2018) 'A cooperative GPU-based parallel multistart simulated annealing algorithm for quadratic assignment problem', *JESTECH*, 21(5), pp. 843–849. doi: 10.1016/j.jestch.2018.08.002.

8. Abdelkafi, O., Derbel, B., Liefooghe, A. (2019) 'A parallel tabu search for the large-scale quadratic assignment problem', *Proc. IEEE CEC*, pp. 3070–3077. doi: 10.1109/CEC.2019.8790152.

9. Ozcetin, E., Ozturk, G. (2018) 'A parallel iterated local search algorithm on GPUs for quadratic assignment problem', *IJET*, 4(2), pp. 123–127.

10. Loranca, B.B., Analco, M.E., Velázquez, R.G., López, A.S. et al. (2017) 'Quadratic assignment problem: A solution approach with parallel GRASP', *IJCOPI*, 8(3), pp. 33–38.

11. Cardenas, E.G., Poveda, R.Ch., Garcia, O.H. (2017) 'A solution for the quadratic assignment problem (QAP) through a parallel genetic algorithm based grid on GPU', *Appl. Math. Sci*., 11(57), pp. 2843–2854. doi: 10.12988/ams.2017.710319.

12. Codognet, Ph., Munera, D., Diaz, D., Abreu, S. (2018) 'Parallel local search', in Hamadi, Y., Sais, L. (eds) *Handbook of Parallel Constraint Reasoning*, pp. 381–417. doi: 10.1007/978-3-319-63516-3_10.

13. Fujii, K., Ito, N., Kim, S., Kojima, M., Shinano, Y., Toh, K.-Ch. (2021) 'Solving challenging large scale QAPs', *ZIB Report*, art. 21-02, pp. 1–15.

14. Ralphs, T., Shinano, Y., Berthold, T., Koch, T. (2018) 'Parallel solvers for mixed integer linear optimization', in *Handbook of Parallel Constraint Reasoning*, pp. 283–336. doi: 10.1007/978-3-319-63516-3_8.

15. Li, S., Li, W., Zhang, H., Wang, Z. (2019) 'Research and implementation of parallel artificial bee colony algorithm based on ternary optical computer', *Automatika*, 60(4), pp. 422–431. doi: 10.1080/00051144.2019.1639118.

16. Thike, A.M., Lupin, S.A., Balabaev, S.A. (2023) 'The features of using a brute force algorithm for solving a quadratic assignment problem', *INJOIT*, 11(7), pp. 60–68 (in Russ.).

17. Kumar, M., Mitra, P. (2023) 'Solving quadratic assignment problem using iterated local search on GPU spatial memory', *Proc. Int. Conf. ACIA*, 2705(1), pp. 1–9. doi: 10.1063/5.0133414.

18. Thike, A.M., Lupin, S. (2019) 'Parallel application for wireless network topology optimisation', *IJEEDC*, 7(4), pp. 14–20.

19. Rybakov, A.A., Meshcheryakov, A.O. (2023) 'Vectorization of the three-dimensional immersed boundary method for improving the efficiency of calculations on Intel microprocessors', *Software & Systems*, 36(1), pp. 130–143 (in Russ.). doi: 10.15827/0236-235X.141.130-143.

Авторы

Ай Мин Тайк [1], к.т.н., докторант, ayeminthike52@gmail.com
Лупин Сергей Андреевич [1, 2], к.т.н., профессор, lupin@miee.ru
Федяшин Дмитрий Андреевич [1], начальник ВЦ, fedyashinda@gmail.com

[1] Национальный исследовательский университет «МИЭТ», г. Москва, 124498, Россия
[2] Межведомственный суперкомпьютерный центр РАН, г. Москва, 119334, Россия

Authors

Aye Min Thike [1], Cand. of Sci. (Engineering) Doctoral Student, ayeminthike52@gmail.com
Sergey A. Lupin [1, 2], Cand. of Sci. (Engineering), Professor, lupin@miee.ru
Dmitry A. Fedyashin [1], Head of Computer Center, fedyashinda@gmail.com

[1] National Research University of Electronic Technology, MIET, Moscow, 124498, Russian Federation
[2] Joint Supercomputer Center of RAS, Moscow, 119334, Russian Federation

УДК 004.056 doi: 10.15827/0236-235X.142.615-631 2023. Т. 36. № 4. С. 615–631

Обеспечение информационной безопасности научного суперкомпьютерного центра

А.В. Баранов
П.М. Корепанов
Е.Е. Кузнецов

Ссылка для цитирования
Баранов А.В., Корепанов П.М., Кузнецов Е.Е. Обеспечение информационной безопасности научного суперкомпьютерного центра // Программные продукты и системы. 2023. Т. 36. № 4. С. 615–631. doi: 10.15827/0236-235X.142.615-631
Информация о статье
Поступила в редакцию: 25.08.2023 После доработки: 28.09.2023 Принята к публикации: 28.09.2023

Аннотация. В большинстве научных суперкомпьютерных центров (СКЦ) коллективного пользования обрабатывается открытая информация. Для ее защиты, как правило, применяются штатные технологии информационной безопасности, встроенные в используемые операционные системы, системы хранения данных, сетевые устройства. Наблюдается рост как числа угроз безопасности информации, так и проводимых в отношении СКЦ компьютерных атак и состоявшихся инцидентов, что несет для центров репутационные и финансовые риски. В статье рассмотрены особенности обработки информации в СКЦ, существенно ограничивающие применение известных мер и средств защиты информации. К таким особенностям отнесены свобода пользователя СКЦ в выборе инструментальных средств и прикладных программных пакетов для решения своих исследовательских задач, необходимость обеспечения максимальной скорости расчетов на предоставленных пользователям суперкомпьютерных ресурсах, ограниченность применения защищенных операционных систем и средств обновления системного программного обеспечения. Обоснована актуальность разработки комплексного системного подхода к защите информации, при котором достаточный уровень информационной безопасности СКЦ обеспечивается без существенных ограничений спектра и снижения качества предоставляемых пользователям услуг по высокопроизводительным вычислениям. Рассмотрены актуальные угрозы безопасности информации СКЦ, приведена классификация обрабатываемых данных, определен перечень актуальных мер защиты информации. С учетом исследованных особенностей защиты информации в СКЦ представлен вариант построения системы информационной безопасности центра, основанный на разделении информационно-вычислительной инфраструктуры центра на зоны безопасности и применении средств контроля сетевого периметра и анализа событий безопасности.

Ключевые слова: информационная безопасность, защита информации, суперкомпьютерный центр, средства контроля периметра сети, NTA, SIEM
Благодарности. Работа выполнена в МСЦ РАН в рамках государственного задания по теме FNEF-2022-0016

Введение. Главной задачей *суперкомпьютерных центров* (СКЦ), функционирующих в сфере науки и образования, является предоставление услуг по высокопроизводительным вычислениям многочисленным пользователям – исследователям и студентам. В основном обрабатываемая в СКЦ коллективного пользования информация является открытой, и для ее защиты, как правило, применяются штатные технологии информационной безопасности, встроенные в используемые операционные системы, *системы хранения данных* (СХД), сетевые устройства. Последствиями компьютерных инцидентов в СКЦ могут быть искажение и потеря информации, что может привести к задержке в проведении научных исследований, но, как правило, не влечет за собой прямые убытки. Другими словами, основные риски являются не финансовыми, а репутационными, связанными с утратой доверия со стороны пользователей. В последние годы наблюдается рост как числа угроз безопасности информации, так и проводимых в отношении СКЦ компьютерных атак и состоявшихся инцидентов, и это позволяет говорить об актуальности комплексного системного подхода к обеспечению информационной безопасности СКЦ. Следует отметить, что тема информационной безопасности СКЦ все чаще поднимается в мировом научном сообществе, по данному направлению проводятся различные конференции, такие как IEEE/ACM S-HPC [1].

Как известно, любые защитные меры в отношении информационной инфраструктуры ограничивают возможности ее применения. Например, повсеместно используемый полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей (CAPTCHA) [2] увеличивает время входа в информационную систему, и нередко пользователи испытывают трудности при его прохождении. Предоставляемые научным СКЦ услуги по высокопроизводительным вычислениям предполагают свободу пользователя в выборе инструментальных средств и прикладных программных пакетов для решения своих исследо-

вательских задач, однако применение некоторых мер по защите информации может ограничить эту свободу. Кроме того, для пользователя СКЦ важным является достижение максимальной скорости расчетов, то есть работа суперкомпьютерных ресурсов с производительностью, по возможности приближенной к пиковой. Современные программные средства защиты могут потреблять значительную долю процессорного времени, их использование на вычислительных узлах суперкомпьютера может отрицательно сказаться на производительности вычислений. В этих условиях актуальной научно-технической задачей является разработка такого комплекса мер защиты информации в СКЦ, который, обеспечивая достаточный уровень информационной безопасности, не ограничивает (или несущественно ограничивает) спектр предоставляемых пользователям услуг по высокопроизводительным вычислениям и не снижает их качество. Настоящая статья посвящена поиску возможных решений этой задачи.

Обзор актуальных исследований

Основными направлениями исследований в области высокопроизводительных вычислений являются архитектура суперкомпьютерных систем, процессоров и коммуникационных сред, эффективные параллельные алгоритмы, системное и инструментальное ПО суперкомпьютеров, а также методы и средства эффективного использования их вычислительных ресурсов. Информационная безопасность СКЦ пока находится вне их фокуса, и публикации по данной тематике немногочисленны. Тем не менее, в области защиты информации в СКЦ можно выделить несколько направлений исследований.

Часть публикаций посвящена вопросам стандартизации подходов к обеспечению информационной безопасности высокопроизводительных вычислительных систем. Так, в работе [3] рассмотрены наиболее распространенные угрозы безопасности СКЦ, приведены классификации возможных угроз по источникам их возникновения и уязвимостей компонентов СКЦ, обусловливающих эти угрозы. Предложены методы противодействия угрозам и модель защиты информации в виде трехдольного графа. Система безопасности, представляющая собой множество средств защиты информации, выступает в качестве некоего барьера между защищаемыми объектами (компонентами СКЦ) и угрозами. Считается, что система безопасна, если достигается полное перекрытие, предусматривающее не менее одного барьера между угрозами и объектами защищаемой системы. В работе [4] представлена эталонная модель системы высокопроизводительных вычислений, разделенной на четыре функциональные зоны: вычислений, данных, управления и доступа. Для каждой зоны определены их возможные конфигурации и выделены основные виды угроз. Предложенные эталонная модель, способы и рекомендации по обеспечению безопасности информации в СКЦ были использованы в ходе настоящего исследования.

В исследовании [5] автор выделяет принципиальные отличия защиты информации в суперкомпьютерах от традиционных систем обработки данных и определяет две ключевые проблемы, решение которых будет освещаться и в данной работе. Первая проблема заключается в необходимости предоставлять непосредственный доступ к вычислительным ресурсам большому количеству пользователей, которым на выделенных ресурсах разрешено выполнять произвольный программный код, вторая – в том, что многие распространенные решения по защите информации расходуют значительную вычислительную мощность и оказываются по этой причине неэффективными в области высокопроизводительных вычислений. Основное внимание автор уделяет угрозам, связанным с целостностью и доступностью информации, в числе которых изменение кода или данных, неправомерное использование вычислительных ресурсов, отказ в обслуживании суперкомпьютерной системы или сети.

Другая область исследований связана с алгоритмами шифрования данных. Например, в работе [6] на базе суперкомпьютера Sunway TaihuLight реализована модель защиты данных с использованием параллельных алгоритмов AES и SHA3, позволяющая достичь высокой эффективности шифрования/дешифрования данных и гарантировать их целостность. Модель защиты данных состоит из двух частей: шифрования и дешифрования. В работе предложены стратегии оптимизации параллельного алгоритма AES, учитывающие особенности вычислительной архитектуры и иерархии памяти. Авторам в результате оптимизации алгоритма удалось увеличить пропускную способность модели защиты данных почти в два раза.

Организации доступа пользователей к суперкомпьютерным ресурсам посвящена публикация [7]. Авторы рассматривают используе-

мые инструменты и процесс включения портала MIT SuperCloud Portal для федеративной аутентификации с федерацией InCommon и инфраструктурой открытых ключей правительства США. В результате внедрения системы пользователи получили возможность применения надежных существующих систем многофакторной аутентификации, развернутых в их основных учреждениях. Описан также процесс самостоятельной регистрации и проверки ключей ssh.

Важной областью исследований являются методы и средства анализа системных журналов и поиска аномалий в работе суперкомпьютера. С этой точки зрения информационная безопасность суперкомпьютерных систем рассматривается в [8]. Анализ системных журналов увязывается с известными атаками и распространенными методами и средствами защиты, в частности, методами обнаружения вторжений. Авторы делают вывод о недостаточном использовании системных журналов. В работе [9] представлены подробный обзор и оценка шести методов обнаружения аномалий, проведено сравнение их точности и эффективности на двух наборах реальных данных, а также предложены программные инструменты с открытым исходным кодом для реализации рассмотренных методов. В [10] авторы предлагают свою систему обнаружения аномалий/вторжений на основе анализа журналов. Обнаружение аномалий в суперкомпьютерных системах рассматривается как последовательный процесс принятия решений с применением методов обучения с подкреплением. Публикация [11] посвящена программной платформе обнаружения и диагностики аномалий в системных журналах в режиме онлайн с использованием длинной краткосрочной памяти (LSTM) для моделирования системного журнала как последовательности естественного языка.

Помимо анализа системных журналов, поиск связанных с безопасностью аномалий в работе суперкомпьютерной системы может осуществляться при помощи мониторинга. Так, в [12] рассмотрен подход к автоматизированному обнаружению аномалий в HPC-системах с использованием машинного обучения. Авторы применили особый тип нейронных сетей, называемый автокодировщиком (autoencoder). Сети обучены нормальному поведению каждого вычислительного узла на основе его исторических телеметрических данных о «хорошем» поведении. Автокодировщики обучаются и тестируются на серийном суперкомпьютере и развертываются как расширение встроенных устройств мониторинга. В работе [13] продемонстрировано, как данные системы мониторинга Nagios могут быть использованы для решения задачи обнаружения и предсказания аномалий в высокопроизводительных вычислительных системах, причем без применения супервизорных подходов.

Некоторые из публикаций посвящены защите инженерной инфраструктуры СКЦ. В работе [14] показано, как злоумышленник может атаковать СКЦ, нарушив работу систем управления состоянием окружающей среды в помещениях, где размещены вычислительные узлы. На реальных данных продемонстрировано использование злоумышленником систем управления, обеспечивающих подачу воды в подсистему охлаждения, в качестве точек входа для косвенного воздействия на вычислительные ресурсы СКЦ. В исследовании [15] затронута тема влияния повышения потребляемой мощности на сбои в работе вычислительных систем, такие как отключение питания при перегрузке электросети, что, в свою очередь, приводит к потере результатов вычислений при выключении узлов и увеличении времени ожидания в очереди из-за их остановки. Также авторы предлагают согласованный подход к сбору, обработке и хранению статистики энергопотребления для пользовательских приложений, которые могут выполняться на разных суперкомпьютерах одного центра.

Важнейшей областью обеспечения информационной безопасности СКЦ является защита его сетевой инфраструктуры. В [16] предложен вариант межсетевого экрана, который позволяет установить набор правил, регулирующих соединения во внутренней сети суперкомпьютера с помощью Linux netfilter. В работе [17] рассмотрены два метода «грубой силы» обнаружения атак, применяемых в Корейском институте науки и технологий. Один метод заключается в анализе системных журналов на предмет выявления событий неудачной аутентификации. Считается, что атака обнаружена, если количество неудачных попыток входа превышает заданное пороговое значение. Другой метод состоит в анализе событий межсетевого экрана, заключающемся в отсеивании определенных подключений правилами экрана. Такие события группируются по адресам источника и назначения отсеянного подключения, и, если количество событий в какой-либо группе превышает пороговое значение, считается, что атака обнаружена.

Анализ публикаций последних лет дает основания утверждать, что обеспечение информационной безопасности СКЦ коллективного пользования является актуальной научно-технической задачей, решение которой находится на начальной стадии. Необходимы исследования и разработки в области безопасных архитектур суперкомпьютерных систем и центров, обеспечивающих в то же время эффективное и удобное использование вычислительных ресурсов для проведения высокопроизводительных расчетов.

Угрозы безопасности информации в научном СКЦ

Угрозы несанкционированного доступа к информации возникают практически везде, где применяются средства вычислительной техники. Научные СКЦ не являются исключением – инциденты безопасности возникают в таких средах, по крайней мере, с 1980-х годов [18, 19] и регулярно происходят в настоящее время (см., например, https://csirt.egi.eu/attacks-on-multiple-hpc-sites/). Так же, как и любой другой компьютер, суперкомпьютеры подключаются к сети и работают под управлением стандартных операционных систем, как правило, на базе Linux, и, соответственно, подвергаются многим традиционным атакам, таким как подбор паролей и сканирование портов. В работе [4] дана следующая классификация угроз безопасности информации в СКЦ.

- Угрозы, которые могут привести к утечке данных или нарушению их целостности. Опасность этих угроз связана с большим числом пользователей СКЦ и, соответственно, с относительно высокой вероятностью нарушения (злонамеренного или нет) кем-то из пользователей правил информационной безопасности.
- Злоупотребление вычислительными ресурсами: использование суперкомпьютерных мощностей вычислительных комплексов для выполнения действий, нарушающих законодательство, например, для майнинга криптовалюты.
- Вредоносное ПО. Суперкомпьютеры могут стать целью атак, направленных на установку вредоносного ПО, которое обусловливает компрометацию учетных данных пользователей, нарушение работоспособности системы или утечку конфиденциальной информации.
- Сетевые угрозы [17, 20]. Суперкомпьютеры, как правило, подключены к сети Интернет, что делает их уязвимыми для различных сетевых атак, таких как отказ в обслуживании (DoS), перехват сетевого трафика (сниффинг), атаки на протоколы прикладного уровня, такие как Secure Shell, эксплуатация уязвимостей сетевого оборудования, сканирование периметра сети и другие.
- Угрозы инженерной инфраструктуре СКЦ: несанкционированный физический доступ к оборудованию, повреждение компонентов [9], пожары, наводнения и другие чрезвычайные ситуации.
- Уязвимости приложений и операционных систем, например, в параллельных файловых системах [21]. Наличие таких уязвимостей может позволить злоумышленникам получить несанкционированный доступ к суперкомпьютерным ресурсам и данным пользователей, а также осуществить иные типы атак. Возможность таких атак обусловлена вероятными уязвимостями в коде ПО, ошибками конфигурации или недостаточными мерами безопасности.
- Социальная инженерия. Угрозы, основанные на социальной инженерии, направлены на манипулирование сотрудниками или пользователями СКЦ с целью получения несанкционированного доступа к системе или конфиденциальной информации. Такие угрозы могут включать фишинг [22], а также вовлечение сотрудников в устный или письменный диалог с целью получения от них информации, необходимой для осуществления той или иной атаки.
- Внутренние угрозы, которые могут представлять серьезную угрозу для безопасности суперкомпьютера. В роли злонамеренных или невольных нарушителей могут выступать авторизованные пользователи суперкомпьютерных систем, системные администраторы и администраторы информационной безопасности, лица, привлекаемые для установки, настройки, испытаний, пусконаладочных и иных видов работ, лица, обеспечивающие функционирование СКЦ: охрана, уборщики и др. Перечисленные потенциальные внутренние нарушители могут осуществить в отношении суперкомпьютерных систем несанкционированный доступ или иные злонамеренные либо ошибочные действия [23].

Типовая схема обработки данных в научном СКЦ

Прежде чем переходить к вопросам обеспечения информационной безопасности в научном СКЦ, необходимо рассмотреть типовую схему организации такого центра (рис. 1), а также категории обрабатываемой в нем инфор-

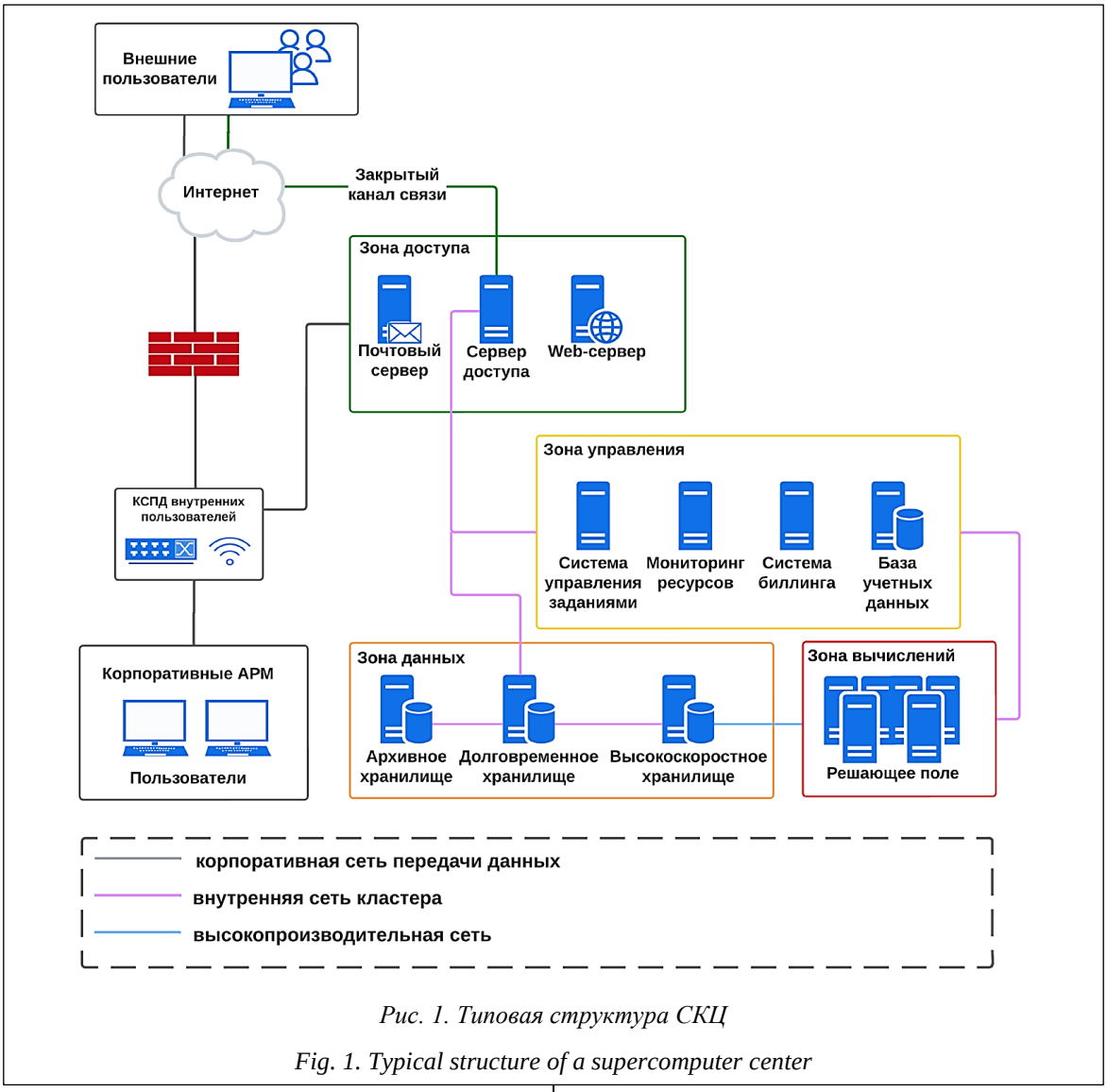


Рис. 1. Типовая структура СКЦ

Fig. 1. Typical structure of a supercomputer center

мации. Воспользовавшись результатами работы [4], разделим компоненты СКЦ на зоны безопасности.

Ядром любого СКЦ является зона высокопроизводительных вычислений, в которой размещаются одна или нескольких высокопроизводительных вычислительных установок, собственно, и называемых суперкомпьютерами. Каждая установка состоит из пула вычислительных (суперкомпьютерных) узлов, представляющих собой серверы с одним или несколькими многоядерными процессорами. Достаточно часто узлы комплектуются ускорителями вычислений, как правило, на базе графических процессоров. Для возможности информационного взаимодействия узлы суперкомпьютера объединяются так называемым интерконнектом – высокоскоростной низколатентной сетью, такой как Infiniband или Intel OmniPath. Дополнительно к интерконнекту суперкомпьютерные узлы оснащаются транспортной сетью для подключения к СХД и сетью управления. Обратим внимание на важный с точки зрения безопасности факт: каждый узел управляется собственной операционной системой и имеет уникальные сетевые адрес и имя.

Главной задачей суперкомпьютерной системы является обеспечение выполнения параллельных программ. Под параллельной программой подразумевается программа, содержащая отдельные части, обычно называемые ветвями, которые могут выполняться одновременно, причем каждая ветвь на отдельном вычислительном устройстве (процессоре, ядре, вычислительном узле). На практике ветви такой программы представляют собой процессы операционной системы, выполняющиеся на одном или нескольких узлах суперкомпьютера.

Взаимодействие процессов параллельной программы осуществляется через интерконнект при помощи специализированных коммуникационных библиотек, таких как MPI, SHMEM и др. Драйверы интерконнекта, коммуникационные библиотеки, подсистемы управления заданиями и ресурсами, мониторинга и управления учетными записями пользователей формируют специализированный программный стек, полный состав которого на примере Межведомственного суперкомпьютерного центра РАН рассмотрен в [24, 25]. Являясь важнейшим стержневым компонентом, программный стек формируется индивидуально для каждого суперкомпьютера, причем процесс его формирования, конфигурирования и настройки, как показывает практика, требует значительных времени и трудозатрат высококвалифицированного персонала СКЦ. На программный стек суперкомпьютера опираются все запускаемые в зоне вычислений параллельные программы, которые могут быть как разработаны самими пользователями, так и являться компонентами специализированных прикладных программных пакетов для высокопроизводительных вычислений.

Прямой доступ к зоне вычислений и, соответственно, к суперкомпьютерным узлам для пользователей закрыт. Терминальные устройства пользователей (рабочие станции, ноутбуки, планшеты, телефоны и т.п.) расположены вне границ СКЦ, и пользователи с этих устройств соединяются со специально выделенным сервером доступа, размещенным в одноименной зоне. Зона доступа включает узлы, подключенные к внешним сетям, таким как корпоративная сеть СКЦ или Интернет. Эта зона предоставляет средства для авторизации пользователей, разграничения их доступа к информационным и вычислительным ресурсам, обеспечения подключений пользователей и системных администраторов к суперкомпьютерам. Сервер доступа служит для подготовки пользователями параллельных программ, исходных данных, анализа результатов расчетов. Чтобы подготовленная параллельная программа могла быть выполнена на решающем поле, пользователь должен оформить так называемое задание – информационный объект, включающий саму программу, требования к ресурсам (какие узлы суперкомпьютера, в каком количестве и на какое время должны быть предоставлены для выполнения программы) и исходные данные. Подготовленное задание направляется в специальную программную систему управления заданиями [26], отвечающую за ведение очереди заданий и распределение узлов суперкомпьютера между различными параллельными программами разных пользователей.

Система управления заданиями функционирует на выделенном сервере управления, размещенном в одноименной зоне. Зона управления включает в себя множество серверов, через которые системные администраторы могут конфигурировать, настраивать, тестировать и контролировать компоненты зоны вычислений, в том числе формировать или модифицировать программный стек суперкомпьютера. Здесь же размещаются серверы и подсистемы мониторинга суперкомпьютерных ресурсов и учета их потребления пользователями. Непосредственный доступ в зону управления разрешен только администраторам, причем администратор входит сначала в зону доступа и только затем в зону управления. Для пользователей суперкомпьютеров зона управления доступна только через интерфейс системы управления заданиями. Зона управления является наиболее критичной с точки зрения информационной безопасности, поскольку содержит множество точек отказа.

Пользовательские данные размещаются в одной или нескольких СХД, которые в соответствии с [4] образуют зону данных. Особенностью СХД СКЦ является то, что каждый раздел СХД монтируется на всех серверах, включая вычислительные узлы, под одним и тем же именем. Это делает данные одинаково доступными пользователю или администратору из зон доступа, управления и вычислений. Для обеспечения безопасности информации важно понимать иерархию СХД современных СКЦ, которую составляют следующие виды хранилищ данных:

– высокоскоростное хранилище так называемых горячих данных;

– долговременное надежное хранилище оперативных данных;

– долговременное архивное (медленное) хранилище так называемых холодных данных.

Под горячими данными понимаются данные, непосредственно используемые параллельными программами при их выполнении в зоне вычислений, в том числе промежуточные результаты расчетов. Главное назначение хранилища таких данных – обеспечение как можно более быстрого доступа к большим объемам информации, сохраненным в результате выполнения параллельной программы. Пользователи активно используют эти хранилища для

скоростной аккумуляции результатов нескольких подряд выполнившихся заданий, а также для передачи результатов одного задания в качестве входных данных следующего за ним задания. С точки зрения обеспечения безопасности важно, что такие хранилища, как правило, являются гиперконвергентными, то есть используют в качестве запоминающих устройств скоростные локальные диски вычислительных узлов суперкомпьютера. В единое хранилище эти диски объединяются при помощи параллельных файловых систем, таких как GPFS или Lustre. Время хранения горячих данных ограничивается системным администратором, а их длительная сохранность не гарантируется. Пользователям рекомендуется регулярно сохранять критичные для них данные в долговременное оперативное хранилище, представляющее собой, как правило, классическую сетевую СХД. Именно в оперативном хранилище выполняются действия по подготовке параллельных программ, исходных данных для них, здесь же сохраняются результаты расчетов, используемые пользователем для анализа. Архивные хранилища не подразумевают оперативного доступа и служат для сохранения резервных копий оперативных данных, а также для длительного хранения редко используемых так называемых холодных данных.

Кроме перечисленных зон доступа, управления, вычислений и данных, в СКЦ может быть выделена корпоративная сеть внутренних пользователей, включающая рабочие места сотрудников СКЦ и серверы обеспечения их деятельности, такие как сервер веб-сайта СКЦ, почтовый сервер, серверы обеспечивающих подразделений (бухгалтерия, охрана, тендерная группа и пр.).

Рассмотрим подлежащие защите данные, обрабатываемые в СКЦ. С точки зрения возрастания требований безопасности эти данные можно разделить на следующие категории:

– программные коды, исходные данные и результаты расчетов;

– стек системного и инструментального ПО суперкомпьютера (программный стек);

– учетные данные пользователей;

– учетные данные системных администраторов суперкомпьютера.

Наименее критичной к безопасности, хотя и наиболее объемной категорией, являются пользовательские программные коды, исходные данные и результаты высокопроизводительных расчетов, размещаемые в зоне данных. Обычно пользовательские данные не содержат информацию ограниченного доступа (например, персональные данные, служебные документы и т.п.), а интерес со стороны потенциальных нарушителей к необработанным результатам расчетов и пользовательскому программному коду сравнительно низкий из-за высокой сложности, неструктурированности данных и отсутствия документации. Обычно для защиты этой категории данных, если нет дополнительных соглашений с отдельными пользователями, применяются штатные средства разграничения доступа, имеющиеся в составе операционных систем суперкомпьютерных узлов, серверов СХД, зон доступа и управления. Хранилище оперативных данных обеспечивает надежное долговременное хранение данных этой категории, но СКЦ, как правило, при этом не дает никаких юридических гарантий обеспечения их сохранности. От аккуратности, дисциплины и бдительности пользователя во многом зависит защита его данных от несанкционированного доступа: пользователь должен обеспечивать конфиденциальность своей аутентификационной информации (значения паролей, закрытых ключей) и регулярное резервное копирование своих данных в независимое от СКЦ пространство хранения.

Упоминавшийся стек системного и инструментального ПО (программный стек) суперкомпьютера чувствителен к угрозам модификации и уничтожения данных. Злонамеренное или ошибочное внесение некорректных изменений в программный стек может привести к неработоспособности суперкомпьютера и простою его ресурсов, поскольку восстановление разрушенного или уничтоженного стека, как показывает практика, занимает длительное время и требует значительных трудозатрат со стороны персонала СКЦ. Рабочая копия программного стека размещается в зоне данных, резервные копии могут создаваться на специально выделенном сервере зоны управления.

Следующей по критичности категорией являются учетные данные пользователей СКЦ. Компрометация (утечка паролей, закрытых ключей и т.п.) пользовательской учетной записи может повлечь несанкционированный доступ к суперкомпьютеру от имени и с правами пользователя. Под угрозой оказываются данные скомпрометированного пользователя, а также высокопроизводительные вычислительные ресурсы, которые от имени пользователя могут быть применены для несанкционированных расчетов, а также для противозаконных действий.

Наиболее защищаемая категория данных – это учетные данные системных администраторов, осуществляющих непосредственное управление суперкомпьютером. Компрометация любой учетной записи этой категории может нанести максимальный ущерб, вплоть до приведения суперкомпьютера в нерабочее состояние (например, путем разрушения программного стека), а также хищения или повреждения всего объема пользовательских данных. Следует отметить, что учетные данные пользователей и системных администраторов могут быть сравнительно просто отделены от остальных категорий и размещены в защищенном сегменте зоны управления.

Особенности обеспечения информационной безопасности СКЦ

Высокопроизводительные вычислительные системы, размещаемые в научных СКЦ, представляют собой отдельный класс научного оборудования, применяемого для исследований преимущественно в режиме коллективного пользования. Предъявляемые требования информационной безопасности не должны препятствовать решению основной задачи СКЦ – организации совместного использования суперкомпьютерных ресурсов множеством пользователей в целях проведения актуальных научных исследований. Кроме этого, развитие методов и средств обеспечения информационной безопасности высокопроизводительных научных вычислений следует рассматривать как отдельную специализированную область науки. Перед тем как выявить проблемные места и отличительные особенности в обеспечении информационной безопасности СКЦ, рассмотрим традиционно применяемые в СКЦ методы и средства защиты информации.

- Аутентификация и авторизация пользователей и сотрудников СКЦ, а также системных администраторов суперкомпьютерных систем. Это базовый метод обеспечения информационной безопасности, повсеместно применяемый в СКЦ. Широко распространены парольная защита, а также применение ключей и сертификатов для подключения к суперкомпьютерным системам. Использование многофакторной аутентификации, а также таких методов, как применение биометрических данных или аппаратных ключей, хотя и повышает уровень безопасности, пока не имеет широкого распространения по причинам, которые будут рассмотрены далее.
- Контроль доступа к информационным и вычислительным ресурсам СКЦ, а также к данным пользователей. Реализация механизмов контроля доступа позволяет определять для каждого пользователя или групп пользователей набор разрешенных операций доступа (чтение, изменение, создание, удаление, выполнение и пр.) к программам и данным. Обеспечивающие контроль методы могут включать списки контроля доступа, ролевую модель доступа или политики безопасности на уровне операционной системы или приложений.
- Шифрование данных, являющееся эффективным способом защиты конфиденциальности информации. В подавляющем большинстве СКЦ, как минимум, шифруется трафик от терминала пользователя до сервера доступа суперкомпьютера. Кроме этого, могут шифроваться данные пользователей в СХД центра, а также содержимое баз данных и передаваемая по сети информация, в том числе передаваемая внутри коммуникационной подсистемы, объединяющей суперкомпьютерные узлы. Отмечается, что применение алгоритмов шифрования и правильное управление ключами являются главными аспектами эффективной защиты данных [6].
- Мониторинг и обнаружение событий и инцидентов безопасности. Они позволяют своевременно выявлять аномальное поведение, вторжения или другие нарушения безопасности. Механизмы мониторинга настраиваются для сбора и анализа данных из различных источников, включая по возможности журналы событий, данные о сетевых соединениях и иные регистрируемые источники события. Обнаружение атак позволяет оперативно реагировать на угрозы безопасности и предпринимать соответствующие действия для предотвращения инцидентов или минимизации их последствий [27].
- Физическая безопасность СКЦ. Обеспечение физической безопасности играет важную роль в защите от несанкционированного доступа к суперкомпьютерам путем проникновения посторонних лиц в машинный зал, а также от физических повреждений оборудования такими лицами. Среди защитных мер применяются контролируемый доступ в помещения, видеонаблюдение, биометрическая идентификация, а также защита от пожара, протечек и других стихийных бедствий.

Следует отметить, что большинство перечисленных методов и средств защиты информации в СКЦ имеют ограниченное применение. Рассмотрим объективные причины, пре-

пятствующие полноценной реализации мер обеспечения информационной безопасности в научно-исследовательских СКЦ.

- Противоречие между требованиями безопасности информации и удобством работы пользователей СКЦ. Проблема заключается в том, что меры по обеспечению безопасности почти всегда приводят к ограничению возможностей пользователей. Перефразируя известную максиму, можно сказать, что самый защищенный суперкомпьютер – это выключенный суперкомпьютер. Защита информации в СКЦ – это всегда баланс между безопасностью и удобством пользователей, которое определяется прежде всего широтой спектра предоставляемых ему услуг и возможностей по производству научных расчетов. Допустим, пользователям предоставляется возможность передачи прав доступа к своим программам и данным, и в какой-то момент некоторый пользователь временно передает определенные права другому пользователю. Если по истечении заданного времени права не будут отозваны (например, по причине забывчивости пользователя), в суперкомпьютерной системе образуется уязвимость, о которой не знают системные администраторы. Допустим, в качестве меры защиты в этом случае будет принят механизм запросов к системному администратору на передачу прав доступа третьим лицам. У системного администратора будет зафиксировано, кто, кому, когда и какие права передал, но, очевидно, в подобной системе совместная работа пользователей будет осуществляться со значительными трудностями и временными задержками. Аналогичные соображения можно привести по поводу двухфакторной аутентификации пользователей, которая заметно затрудняет вход последних в суперкомпьютерную систему, особенно в случае ошибок пользователя при наборе паролей и кодов. Таким образом, фактор удобства работы пользователя может осложнить защиту высокопроизводительной системы коллективного пользования, система может стать более уязвимой.

- Невозможность выполнения доверенного программного кода. Одной из распространенных мер защиты информации является обеспечение выполнения доверенного программного кода на доверенном вычислительном устройстве. Применение такой меры в научном СКЦ невозможно, поскольку проводимые пользователями исследования прямо предполагают применение новых программных пакетов, которые не могут быть доверенными по определению. Пользователями СКЦ широко применяются ПО с открытым исходным кодом и самостоятельно разработанные программы. Как известно, ПО с открытым исходным кодом уязвимо к угрозам цепочки поставок [27]. Самостоятельно разработанное ПО может содержать ошибки, которые в некоторых случаях могут привести к появлению уязвимостей и, соответственно, к актуализации угроз безопасности информации (конфиденциальности, целостности или доступности данных). В научных вычислительных экспериментах широко применяется специализированное ПО, которое может либо не иметь должных доверия и поддержки, либо содержать в себе новые классы уязвимостей. Другими словами, пользователю СКЦ предоставлена базовая возможность выполнения на вычислительных узлах суперкомпьютера произвольного программного кода, качественную проверку которого невозможно осуществить. Особо следует отметить невозможность изоляции недоверенного кода в рамках некоторой виртуальной машины, поскольку накладные расходы на виртуализацию существенны и значительно снижают производительность расчетов [28]. Одним из актуальных направлений в этой области является исследование безопасности новых технологий, предлагаемых для включения в программный стек суперкомпьютерной системы.

- Ограниченность вычислительных ресурсов для средств защиты информации. Как известно, применение современных средств защиты информации, таких как средства шифрования или глубокого анализа сетевого трафика, способно заметно замедлить работу вычислительной системы. В то же время главной функцией суперкомпьютерных систем является обеспечение максимальной производительности расчетов. Можно сказать, что безопасность информации в СКЦ ценна только в той степени, в какой она не замедляет работу суперкомпьютеров и не препятствует высокопроизводительным вычислениям, и в этой связи необходимо минимизировать негативное влияние на производительность мер защиты информации. Кроме этого, некоторые особенности построения суперкомпьютерных систем делают некоторые средства защиты неприменимыми в принципе. Например, информационные обмены между процессами параллельной программы, выполняющимися на разных вычислительных узлах, осуществляются через интерконнект с применением механизмов прямого доступа одного узла к оперативной памяти другого. Сетевой трафик интерконнекта даже теоретически

невозможно обработать при помощи средств анализа трафика, рассчитанных на работу с сетевыми протоколами стека TCP/IP.

- Ограниченность применимости защищенных операционных систем, таких как Astra Linux SE, которые на текущий момент не поддерживают программный стек суперкомпьютера. Как правило, для этих систем отсутствуют драйверы интерконнекта и эффективные реализации коммуникационных библиотек. Отсутствие поддержки программного стека значительно ограничивает применение защищенных операционных систем в качестве операционных систем вычислительных узлов суперкомпьютера. При этом подобные системы могут использоваться в качестве операционных систем серверов и рабочих станций в зонах доступа, управления или данных, за исключением сервера доступа. Поскольку на сервере доступа пользователи осуществляют трансляцию и сборку своих параллельных программ, на этом сервере в полном объеме должен быть предоставлен программный стек суперкомпьютера. Обычной практикой является идентичность системного и инструментального ПО сервера доступа и вычислительных узлов суперкомпьютера.
- Ограниченность возможности регулярных обновлений системного ПО. Как известно, такие обновления являются неотъемлемыми элементами защиты информации и позволяют устранять известные уязвимости и исправлять ошибки, которые могут быть использованы злоумышленниками для несанкционированного доступа к ресурсам или проведения компьютерных атак. Однако, как уже упоминалось, важнейшим элементом любого СКЦ является программный стек суперкомпьютера. Любое обновление системного ПО влечет за собой неизбежное внесение изменений в программный стек и может привести к его неработоспособности или снижению производительности. Каждое изменение в программном стеке тщательно тестируется специалистами СКЦ, прежде чем внедряется в обслуживающие пользователей суперкомпьютерные системы. Далеко не все обновления системного ПО, даже критические с точки зрения безопасности информации, могут быть применены без ущерба для качества функционирования СКЦ.

Таким образом, защита высокопроизводительных вычислительных систем требует применения комплексного системного подхода, включающего различные методы и механизмы. Решения для обеспечения информационной безопасности должны быть не только направлены на борьбу с потенциальными нарушителями, в том числе и в контексте цепочки поставок оборудования, цепочки поставок ПО и инсайдерских угроз, но и включать в себя сбои, связанные с недостатками и ошибками в аппаратном обеспечении, базовом ПО, операционной системе, библиотеках, компиляторах, а также ошибки проектирования и реализации инфраструктуры высокопроизводительных вычислений. Сюда же следует отнести ошибки пользователей, ошибки в исследовательских программных кодах и рабочих процессах, а также естественные сбои, такие как отказ аппаратных компонентов. Наконец, решения должны быть просто удобными для применения пользователями-исследователями и, в конечном счете, обеспечивать высокие производительность и качество вычислений. В некоторых случаях решением может быть простое отключение критически важных систем от Интернета или, возможно, от любой другой сети [29]. Однако для большинства СКЦ такой подход неприменим.

Средства защиты сетевого периметра и анализа событий безопасности

Рассмотрим концепцию SOC Visibility Triad, предложенную агентством Gartner в 2019 г. [30]. Суть ее заключается в сборе данных из нескольких источников (рис. 2) для обнаружения угроз безопасности, а именно:

– сбор и анализ системных журналов, генерируемых компонентами вычислительной инфраструктуры: терминальными устройствами пользователей, почтовым сервисом, службами аутентификации и авторизации и т.д.;

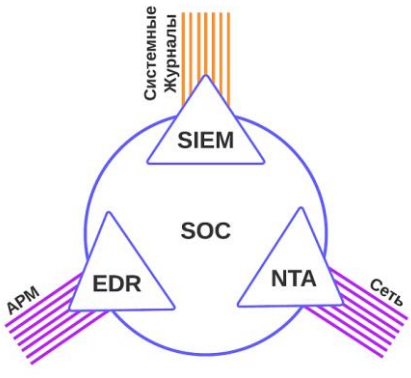


Рис. 2. Источники информации для обнаружения угроз безопасности

Fig. 2. Information sources for detecting security threats

– обнаружение аномалий на терминальных устройствах (автоматизированных рабочих местах) пользователей в корпоративной сети;

– сетецентричное обнаружение потенциальных атак и реагирование на инциденты.

Сетецентричный подход предполагает анализ трафика сетевых устройств с помощью ПО и оборудования для контроля сетевого периметра и трафика внутри сети.

Показанная на рисунке 2 комбинация источников позволяет не только увидеть больше различных событий безопасности, но и сократить время их обнаружения и, следовательно, время реагирования на потенциальные инциденты. Кроме этого, за счет корреляции сетевых аномалий и событий безопасности становится возможным выявление неизвестных атак, для которых у систем контроля сетевого периметра нет сигнатур. При этом, говоря о сетевом периметре, стоит помнить, что стык между корпоративной сетью передачи данных и зонами доступа, управления, вычислений и данных – это тоже граница, часть сетевого периметра, равно как и стыки между самими этими зонами и внутри них, например, стык между сетью управления и интерконнектом внутри зоны вычислений.

Рассматривая возможные решения по обеспечению информационной безопасности, следует учитывать описанные выше особенности научного СКЦ. Например, можно осуществлять контроль границы между транспортной сетью и сетью управления, но не трафика внутри интерконнекта. Применяя концепцию SOC Visibility Triad, выделим из множества средств защиты информации две большие группы: средства контроля сетевого периметра и средства сбора и анализа событий безопасности.

Контроль периметра сети [31, 32] может рассматриваться в качестве одного из основных методов защиты СКЦ. Контроль сетевого периметра включает в себя мониторинг и ограничение доступа к сети центра, а также обнаружение и предотвращение попыток несанкционированного доступа или атак на информационно-вычислительную инфраструктуру СКЦ. Рассмотрим существующие виды средств контроля периметра [32].

- Системы NGFW и UTM. NGFW (Next-Generation Firewall) – это межсетевой экран нового поколения, который комбинирует функции традиционного межсетевого экрана с возможностями контроля приложений и защиты от вторжений. UTM (Unified Threat Management) – интегрированная система управления угрозами, выполняет те же функции, что и системы NGFW, но в отличие от них работает в один поток вместо нескольких, что часто рассматривается как недостаток. При этом стоимость систем UTM ниже по сравнению с NGFW.
- Системы NTA и NDR. NTA (Network Traffic Analysis) – система анализа сетевого трафика, которая позволяет обнаруживать аномальное поведение или подозрительную активность в сети. NDR (Network Detection and Response) представляет собой систему обнаружения и реагирования на сетевые угрозы, которая позволяет выявлять вторжения и другие аномалии в сети, а также принимать меры по их предотвращению.
- Системы IDS и IPS. IDS (Intrusion Detection System) и IPS (Intrusion Prevention System) – соответственно, системы обнаружения и предотвращения вторжений, которые анализируют сетевой трафик и реагируют на подозрительную активность или попытки вторжения. Ключевое отличие систем заключается в том, что IPS может разрывать подозрительное соединение, в то время как IDS только анализирует трафик и сообщает о подозрительной активности. Следует отметить, что эти системы нередко являются частью более сложного и комплексного ПО, но при ограниченном бюджете могут применяться отдельно.

Несомненным преимуществом средств контроля сетевого периметра для СКЦ является то, что они размещаются вне зон суперкомпьютерных систем и таким образом не оказывают негативного влияния ни на производительность вычислений, ни на удобство и порядок работы пользователей СКЦ. В статье [32] приведена таблица рассмотренных средств контроля периметра, позволяющая сравнить функциональные возможности разных видов и классов этих средств. Авторы рекомендуют обратить внимание на NTA-системы, ярким примером которых является PT NAD (Positive Technologies Network Attack Discovery, см. https://www.ptsecurity.com/ru-ru/research/analytics/network-traffic-analysis-2022/). Система PT NAD предоставляет функциональность обнаружения и анализа сетевой активности без вмешательства в работу сети. Она способна выявлять подозрительные пакеты данных, несанкционированные подключения и другие аномалии в сетевом трафике. Система основана на использовании алгоритмов машинного обучения и повышенной гибкости в настройке, что позволяет ей адаптироваться к изменяющимся угрозам и обеспечивать безопасность

СКЦ. Она анализирует не только внешний, но и внутренний трафик и детектирует перемещения злоумышленников внутри сети, попытки эксплуатации уязвимостей, атаки на конечных пользователей в домене и на внутренние сервисы. Обзор других российских NTA-решений можно найти в https://securitymedia.org/analytics/obzor-rossijskih-nta-ndr-sistem.html. Следует отметить, что одной из проблем использования NTA-систем является большое количество генерируемой информации о различных процессах в сети.

Сетевой периметр – это первое, с чем сталкивается внешний нарушитель, поэтому необходимо контролировать входящий и исходящий трафики. Однако только этого недостаточно – необходима организация нескольких уровней защиты сети. Так, Агентство национальной безопасности США рекомендует следующие меры [33]:

– применение пограничного маршрутизатора для подключения к внешней сети, например, к интернет-провайдеру;

– организация нескольких уровней NGFW по всей сети для ограничения входящего трафика, ограничения исходящего трафика и проверки всей внутренней активности между разрозненными регионами сети, причем рекомендуется на каждом уровне применять системы NGFW разных производителей для защиты от атак, использующих одну и ту же уязвимость;

– создание демилитаризованной зоны (DMZ) – изолированной зоны между корпоративной и внешней сетями, в которой размещаются общедоступные системы и сервисы (вебсайт, почта, LDAP и т.п.);

– мониторинг сети при помощи систем обнаружения вторжений (IDS) или NTA-систем;

– выделение нескольких серверов журналирования для выявления корреляции событий безопасности на разных средствах защиты сети;

– наличие резервных средств защиты сети в зонах с наибольшим объемом трафика для обеспечения оперативной балансировки нагрузки с целью увеличения пропускной способности сети и уменьшения задержек.

Таким образом, защищая периметр сети СКЦ и анализируя трафик внутри корпоративной сети, а также защищая публичные сервисы, расположенные в зоне доступа (почтовый сервер, веб-сервер), и выделяя их в отдельную демилитаризованную зону, возможно обеспечить безопасность, а также автоматизировать реагирование на инциденты, что очень важно при большом объеме трафика в сети.

Реализация рассмотренного комплекса мер приводит к тому, что в сети появляется большое количество оборудования и различных систем безопасности, которые постоянно генерируют поток событий. В связи с этим возникает проблема анализа этого потока для выявления атак или признаков компрометации системы. Для этой цели применяются системы управления событиями информационной безопасности (системы SIEM – Security Information and Event Management [34]), способные обнаруживать и реагировать на угрозы, а также выявлять и анализировать события безопасности в режиме реального времени.

Система SIEM предназначена для анализа информации, поступающей из различных источников, таких как системы контроля периметра, антивирусные системы, сетевое оборудование, журналы событий и другие источники. Как только SIEM обнаруживает отклонение, он сразу генерирует событие для администратора безопасности. Следует отметить, что SIEM-системы российских производителей являются одними из лидеров рынка и представлены такими продуктами, как KUMA (Kaspersky Unified Monitoring and Analysis Platform), MaxPatrol SIEM, RuSIEM, KOMRAD Enterprise SIEM. Все эти системы сертифицированы ФСТЭК России. Вместе с коммерческими решениями на рынке также присутствуют свободно распространяемые решения SIEM: ELK Stack (Elasticsearch, Logstash, Kibana), OSSIM (Open Source Security Information Management), Graylog. Сравнительные обзоры представленных на российском рынке SIEM-систем можно найти в работах [35–37].

Для обеспечения информационной безопасности научного СКЦ важно, что SIEM-системы, как и средства контроля сетевого периметра, не потребляют суперкомпьютерные ресурсы и не нарушают порядок работы пользователей СКЦ. Применение их в СКЦ в качестве средств защиты информации более чем целесообразно и оправданно.

Вариант построения системы информационной безопасности научного СКЦ

Проведенный анализ позволяет сформулировать основные принципы построения архитектуры системы информационной безопасности научного СКЦ.

- Для зоны вычислений (вычислительного кластера) не могут быть применимы многие современные средства защиты информации. Тре-

буются эффективное использование штатных механизмов операционных систем вычислительных узлов, проверка безопасности исходного кода программного стека суперкомпьютера, строгое сегментирование сетевого взаимодействия и разделяемых пользовательских вычислительных ресурсов, максимально возможные меры физической защиты.

- Для зоны хранения данных, помимо указанных принципов, допускаются средства защиты при условии их незначительного влияния на скорость передачи данных между СХД и вычислительным кластером. Однако применение гиперконвергентных хранилищ горячих данных, построенных на основе вычислительных узлов суперкомпьютера, осложняет разграничение зон вычислений и данных. В этом случае возможны либо объединение этих зон в единую зону вычислений и данных, либо перенос хранилища горячих данных в зону вычислений.
- Для всех остальных зон (доступа, управления, корпоративной сети передачи данных) возможно применение современных средств защиты, традиционно используемых для защиты типовой корпоративной сетевой инфраструктуры, в том числе с географически распределенными подразделениями. При этом следует помнить о том, что на сервере доступа пользователям должен быть предоставлен программный стек суперкомпьютера для возможности подготовки параллельных программ.

С учетом особенностей обработки информации в СКЦ рассмотрим вариант построения системы информационной безопасности с применением средств контроля сетевого периметра и SIEM. Предлагаемый вариант представлен на рисунке 3.

На уровне сети защита периметра строится с использованием следующих мер:

– публичные сервисы, расположенные в зоне доступа, выделяются в отдельную демилитаризованную зону (сегмент публичных сервисов);

– осуществляется межсетевое экранирование с контролем состояния соединений между общедоступным, корпоративным сегментами и сегментом публичных сервисов;

– сенсоры NTA получают информацию о трафике на периметре сети и в сегментах *корпоративной сети передачи данных* (КСПД) со SPAN-портов коммутаторов; полученный трафик проверяется на наличие признаков нарушителя в сети с помощью сигнатурного и поведенческого анализа;

– осуществляется фильтрация трафика по категориям и отдельным ресурсам с помощью протокола ICAP (см. https://datatracker.ietf.org/doc/html/rfc3507);

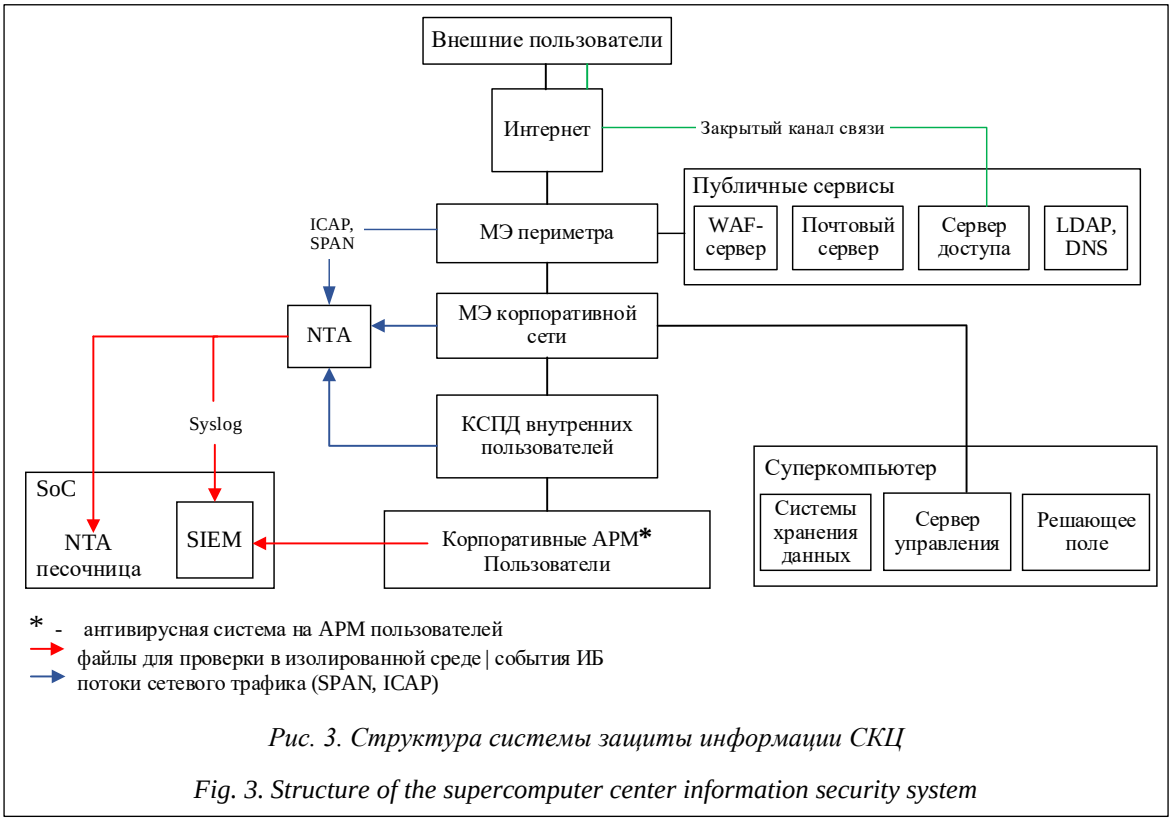


Рис. 3. Структура системы защиты информации СКЦ

Fig. 3. Structure of the supercomputer center information security system

– копируется и затем расшифровывается SSL/TLS-трафик (SSL mirroring);

– обнаруженные в трафике файлы и ссылки отправляются на проверку в локальную песочницу, а также производителю NTA для проверки файлов и ссылок;

– в песочнице файлы и ссылки проверяются с помощью сигнатурного и поведенческого анализа, а также выполняется их запуск в изолированной среде с детальной инспекцией и журналированием действий;

– полученную информацию NTA отправляет в SIEM в виде событий.

На уровне компонентов информационно-вычислительной инфраструктуры СКЦ (серверы, АРМ пользователей и сотрудников и т.п.) устанавливаются антивирусные системы, которые собирают события безопасности, информацию о процессах, работе с файлами, сетевом трафике. При выявлении подозрительной активности принимаются необходимые меры для изоляции соответствующего сервера или АРМ и сбора информации для последующего анализа. Собранная информация отправляется в SIEM, которая объединяет все события и предоставляет интерфейс для мониторинга событий безопасности и реагирования на инциденты. Для повышения общего уровня защищенности инфраструктуры к решениям NTA и SIEM добавляется сетевой экран уровня приложений (Web Application Firewall – WAF). Он обеспечивают защиту веб-серверов при их публикации в сети общего пользования. Кроме этого, реализуются функционал обратного прокси-сервера, расширенные варианты аутентификации, расшифровка SSL, специализированная защита от атак на веб-приложения.

Заключение

Проведенный анализ актуальных исследований в области обеспечения информационной безопасности научного СКЦ показал, что научная разработка вопросов защиты информации в СКЦ находится на начальной стадии, публикации по этой тематике немногочисленны. Обработка информации в СКЦ имеет ряд характерных особенностей, ограничивающих применение известных методов и средств защиты информации. К основным особенностям относятся отсутствие ограничений пользователей на выполняемый ими программный код, невозможность его проверки, недопустимость снижения производительности суперкомпьютерных ресурсов из-за применения средств защиты, ограниченные возможности по применению защищенных операционных систем и обновлению системного ПО вычислительных узлов суперкомпьютера. Для обеспечения защиты информации в СКЦ предлагается разделение информационно-вычислительной инфраструктуры СКЦ на зоны с выделением зон вычислений и данных, для защиты которых предлагается применять средства контроля сетевого периметра и анализа событий безопасности. Предложенный вариант системы информационной безопасности СКЦ позволяет реализовать технологию защиты информации в СКЦ, учитывающую выявленные особенности.

Список литературы

1. Message from the S-HPC 22 workshop chairs. Proc. IEEE/ACM First Int. Workshop on Cyber S-HPC, 2022, pp. iv–iv. doi: 10.1109/S-HPC56715.2022.00004.

2. Xu X., Liu L., Li B. A survey of CAPTCHA technologies to distinguish between human and computer. Neurocomputing, 2020, vol. 408, pp. 292–307. doi: 10.1016/j.neucom.2019.08.109.

3. Абрамов Н.С., Фраленко В.П. Угрозы безопасности вычислительных комплексов: классификация, источники возникновения и методы противодействия // Программные системы: теория и приложения. 2015. Т. 6. № 2. С. 63–83. doi: 10.25209/2079-3316-2015-6-2-63-83.

4. Guo Y., Chandramouli R., Wofford L., Gregg R. et al. High-Performance Computing (HPC) security: Architecture, threat analysis, and security posture. NIST SP, 2023, no. 800-223 ipd. doi: 10.6028/NIST.SP.800-223.ipd.

5. Peisert S. Security in high-performance computing environments. Communications of the ACM, 2017, vol. 60, no. 9, pp. 72–80. doi: 10.1145/3096742.

6. Chen Y., Li K., Fei X., Quan Z., Li K. Implementation and optimization of a data protecting model on the Sunway TaihuLight supercomputer with heterogeneous many-core processors. Concurrency and Computation: Pract. and Experience, 2019, vol. 31, no. 21, art. e4758. doi: 10.1002/cpe.4758.

7. Prout A., Klein A., Michaleas P. et al. Securing HPC using Federated authentication. Proc. IEEE HPEC, 2019, pp. 1–7. doi: 10.1109/HPEC.2019.8916255.

8. Luo Z., Qu Z., Nguyen T.T., Zeng H., Lu Z. Security of HPC systems: From a log-analyzing perspective. ICST Transactions on Security and Safety, 2019, vol. 6, no. 21, art. 163134. doi: 10.4108/eai.19-8-2019.163134.

9. He S., Zhu J., He P., Lyu M.R. Experience report: System log analysis for anomaly detection. Proc. IEEE 27th ISSRE, 2016, pp. 207–218. doi: 10.1109/ISSRE.2016.21.

10. Luo Z., Hou T., Nguyen T.T., Zeng H., Lu Z. Log analytics in HPC: A data-driven reinforcement learning framework. Proc. IEEE INFOCOM, 2020, pp. 550–555. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162664.

11. Du M., Li F., Zheng G., Srikumar V. DeepLog: Anomaly detection and diagnosis from system logs through deep learning. Proc. CCS'17, 2017, pp. 1285–1298. doi: 10.1145/3133956.3134015.

12. Borghesi A., Libri A., Benini L., Bartolini A. Online anomaly detection in HPC systems. Proc. IEEE AICAS, 2019, pp. 229–233. doi: 10.1109/AICAS.2019.8771527.

13. Borghesi A., Molan M., Milano M., Bartolini A. Anomaly detection and anticipation in high performance computing systems. Proc. IEEE Transactions on Parallel and Distributed Systems, 2022, vol. 33, no. 4, pp. 739–750. doi: 10.1109/TPDS.2021.3082802.

14. Chung K., Formicola V., Kalbarczyk Z.T., Iyer R.K., Withers A., Slagell A.J. Attacking supercomputers through targeted alteration of environmental control: A data driven case study. Proc. IEEE Conf. on CNS, 2016, pp. 406–410. doi: 10.1109/CNS.2016.7860528.

15. Kiselev E., Baranov A., Telegin P., Kuznetsov E. System for collecting statistics on power consumption of supercomputer applications. In: LNCS. Proc. RuSCDays, 2022, vol. 13708, pp. 548–561. doi: 10.1007/978-3-031-22941-1_40.

16. Prout A., Arcand W., Bestor D., Bergeron B. et al. Enhancing HPC security with a user-based firewall. Proc. IEEE HPEC, 2016, pp. 1–4. doi: 10.1109/HPEC.2016.7761641.

17. Lee J.-K., Kim S.-J., Hong T. Brute-force attacks analysis against SSH in HPC multi-user service environment. INDJST, 2016, vol. 9, no. 24, pp. 1–4. doi: 10.17485/ijst/2016/v9i24/96070.

18. Stoll C. Stalking the wily hacker. Communications of the ACM, 1988, vol. 31, no. 5, pp. 484–497. doi: 10.1145/42411.42412.

19. Bishop M. UNIX security in a supercomputing environment. Proc. Supercomputing'89, 1989, pp. 693–698. doi: 10.1145/76263.76341.

20. Addai P., Freas R., Tesfa E.M., Sellers M., Mohd T.K. Prevention and detection of network attacks: A comprehensive study. In: LNBIP. Proc. ICDSST, 2023, vol. 474, pp. 56–66. doi: 10.1007/978-3-031-32534-2_5.

21. Wilhelm F. General Pr0ken Filesystem – Hacking IBM's GPFS. ERNW INSINUATOR, 2015. URL: https://insinuator.net/2015/04/general-pr0ken-filesystem-hacking-ibms-gpfs/ (дата обращения: 04.09.2023).

22. Alkhalil Z., Hewage Ch., Nawaf L., Khan I. Phishing attacks: A recent comprehensive study and a new anatomy. Front. Comput. Sci., 2021, vol. 3. doi: 10.3389/fcomp.2021.563060.

23. Liu L., De Vel O., Han Q.-L., Zhang J., Xiang Y. Detecting and preventing cyber insider threats: A survey. IEEE Communications Surveys & Tutorials, 2018, vol. 20, no. 2, pp. 1397–1417. doi: 10.1109/COMST.2018.2800740.

24. Savin G.I., Shabanov B.M., Telegin P.N., Baranov A.V. Joint supercomputer center of the Russian Academy of Sciences: Present and future. Lobachevskii J. Math., 2019, vol. 40, no. 11, pp. 1853–1862. doi: 10.1134/S1995080219110271.

25. Baranov A.V., Savin G.I., Shabanov B.M., Shitik A.S., Svadkovskiy I.A., Telegin P.N. Methods of jobs containerization for supercomputer workload managers. Lobachevskii J. Math., 2019, vol. 40, no. 5, pp. 525–534. doi: 10.1134/S1995080219050020.

26. Reuther A., Byun Ch., Arcand W. et al. Scalable system scheduling for HPC and big data. JPDC, 2018, vol. 111, pp. 76–92. doi: 10.1016/j.jpdc.2017.06.009.

27. Boyens J., Smith A., Bartol N., Winkler K., Hplbrook A., Fallon M. Cybersecurity supply chain risk management practices for systems and organizations. NIST SP, 2022, no. 800-161r1. doi: 10.6028/NIST.SP.800-161r1.

28. Aladyshev O., Baranov A., Ionin R., Kiselev E., Shabanov B. Variants of deployment the high performance computing in clouds. Proc. IEEE EIConRus, 2018, pp. 1453–1457. doi: 10.1109/EIConRus.2018.8317371.

29. Martinez J., Connor C., Hollander B., Paschke K. HPC infrastructure security – ensuring storage and network safety, integrity, efficiency, and performance. Proc. Virtual Conf. Tapia, 2020. doi: 10.2172/1657098.

30. Barros A., Chuvakin A., Belak A. Applying network-centric approaches for threat detection and response. Gartner Research, 2019. URL: https://www.gartner.com/en/documents/3904768 (дата обращения: 04.09.2023).

31. Uctu G., Alkan M., Dogru I.A., Dorterler M. Perimeter network security solutions: A survey. Proc. ISMSIT, 2019, pp. 1–6. doi: 10.1109/ismsit.2019.8932821.

32. Ким Д., Рыжков В. NTA, IDS, UTM, NGFW – в чем разница? // Positive Technologies. 2021. URL: https://www.securitylab.ru/analytics/517592.php?ysclid=lia4byjqfe792657504 (дата обращения: 04.09.2023).

33. Network Infrastructure Security Guide. National Security Agency, Cybersecurity Tech. Report, 2022. URL: https://media.defense.gov/2022/Jun/15/2003018261/-1/-1/0/CTR_NSA_NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20220615.PDF (дата обращения: 04.09.2023).

34. Vielberth M. Security Information and Event Management (SIEM). In: Encyclopedia of Cryptography, Security and Privacy, 2021. doi: 10.1007/978-3-642-27739-9_1681-1.

35. Лазарева Н.Б., Кармадонов М.С. Анализ программных продуктов SIEM-систем с целью выбора решения для защиты ИБ предприятия // Far East Math: мат. науч.-практич. конф. 2022. С. 127–136.

36. Стукалин А.А., Назарова О.Б. Системы мониторинга информационных инцидентов (SIEM-системы): обзор и сравнительный анализ // Управленческие механизмы противодействия идеологии экстремизма и терроризма: мат. науч.-практич. конф. 2018. С. 116–121.

37. Вишнякова А.Н., Рябцев А.А., Кондрашова Е.В., Шинаков К.Е. Обзор основных SIEM-систем, представленных на российском рынке // Информационная безопасность и защита персональных данных. Проблемы и пути их решения: сб. науч.-практич. конф. 2023. С. 62–65.

Software & Systems doi: 10.15827/0236-235X.142.615-631 2023, vol. 36, no. 4, pp. 615–631

Information security of a supercomputer center

Anton V. Baranov
Pavel M. Korepanov
Egor E. Kuznetsov

For citation
Baranov, A.V., Korepanov, P.M., Kuznetsov, E.E. (2023) 'Information security of a supercomputer center', *Software & Systems*, 36(4), pp. 615–631 (in Russ.). doi: 10.15827/0236-235X.142.615-631
Article info
Received: 25.08.2023 After revision: 28.09.2023 Accepted: 28.09.2023

Abstract. Most scientific supercomputer centers (SC) process open access information. Typically, they use standard information security technologies built into operating systems, data storage, and network devices. In recent years we can see an increase in both the number of information security threats and computer attacks and incidents against SCs, which causes reputational and financial risks for SCs. The paper discusses information processing features in SC, which significantly limit known information security measures and tools. Such features include SC user's freedom to choose tools and application software packages to solve their research problems; a need to ensure maximum computing performance of supercomputer resources provided to users; a limited use of secure operating systems and software update tools. The paper justifies the relevance of developing a complex system approach to SC information security. An adequate level of SC information security must be ensured without significant restrictions of user abilities and reducing the quality of high-performance computing services provided to users. The paper discusses current threats to SC information security, provides a classification of data processed in SC, and defines a list of current information security measures. The proposed optional SC information security system structure takes into account the reviewed features of SC information security. The proposed approach is based on dividing SC computing infrastructure into security zones and on using network perimeter control tools, as well as access and information security event analysis.
Keywords: supercomputer center, information security, network perimeter and access control, security information and event management
Acknowledgements. The work was within the state task of JSCC RAS on topic FNEF-2022-0016

References

1. 'Message from the S-HPC 22 workshop chairs', (2022) *Proc. IEEE/ACM First Int. Workshop on Cyber S-HPC*, pp. iv-iv. doi: 10.1109/S-HPC56715.2022.00004.
2. Xu, X., Liu, L., Li, B. (2020) 'A survey of CAPTCHA technologies to distinguish between human and computer', *Neurocomputing*, 408, pp. 292–307. doi: 10.1016/j.neucom.2019.08.109.
3. Abramov, N., Fralenko, V. (2015) 'Computer systems security threats: Classification, sources of origin and counteraction methods', *Program Systems: Theory and Applications*, 6(2), pp. 63–83 (in Russ.). doi: 10.25209/2079-3316-2015-6-2-63-83.
4. Guo, Y., Chandramouli, R., Wofford, L., Gregg, R. et al. (2023) 'High-Performance Computing (HPC) security: Architecture, threat analysis, and security posture', *NIST SP*, (800-223 ipd). doi: 10.6028/NIST.SP.800-223.ipd.
5. Peisert, S. (2017) 'Security in high-performance computing environments', *Communications of the ACM*, 60(9), pp. 72–80. doi: 10.1145/3096742.
6. Chen, Y., Li, K., Fei, X., Quan, Z., Li, K. (2019) 'Implementation and optimization of a data protecting model on the Sunway TaihuLight supercomputer with heterogeneous many-core processors', *Concurrency and Computation: Pract. and Experience*, 31(21), art. e4758. doi: 10.1002/cpe.4758.
7. Prout, A., Klein, A., Michaleas, P. et al. (2019) 'Securing HPC using Federated authentication', *Proc. IEEE HPEC*, pp. 1–7. doi: 10.1109/HPEC.2019.8916255.
8. Luo, Z., Qu, Z., Nguyen, T.T., Zeng, H., Lu, Z. (2019) 'Security of HPC systems: From a log-analyzing perspective', *ICST Transactions on Security and Safety*, 6(21), art. 163134. doi: 10.4108/eai.19-8-2019.163134.
9. He, S., Zhu, J., He, P., Lyu, M.R. (2016) 'Experience report: System log analysis for anomaly detection', *Proc. IEEE 27th ISSRE*, pp. 207–218. doi: 10.1109/ISSRE.2016.21.
10. Luo, Z., Hou, T., Nguyen, T.T., Zeng, H., Lu, Z. (2020) 'Log analytics in HPC: A data-driven reinforcement learning framework', *Proc. IEEE INFOCOM*, pp. 550–555. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162664.
11. Du, M., Li, F., Zheng, G., Srikumar, V. (2017) 'Deeplog: Anomaly detection and diagnosis from system logs through deep learning', *Proc. CCS'17*, pp. 1285–1298. doi: 10.1145/3133956.3134015.
12. Borghesi, A., Libri, A., Benini, L., Bartolini, A. (2019) 'Online anomaly detection in HPC systems', *Proc. IEEE AICAS*, pp. 229–233. doi: 10.1109/AICAS.2019.8771527.
13. Borghesi, A., Molan, M., Milano, M., Bartolini, A. (2022) 'Anomaly detection and anticipation in high performance computing systems', *Proc. IEEE Transactions on Parallel and Distributed Systems*, 33(4), pp. 739–750. doi: 10.1109/TPDS.2021.3082802.
14. Chung, K., Formicola, V., Kalbarczyk, Z.T., Iyer, R.K., Withers, A., Slagell, A.J. (2016) 'Attacking supercomputers through targeted alteration of environmental control: A data driven case study', *Proc. IEEE Conf. on CNS*, pp. 406–410. doi: 10.1109/CNS.2016.7860528.
15. Kiselev, E., Baranov, A., Telegin, P., Kuznetsov, E. (2022) 'System for collecting statistics on power consumption of supercomputer applications', in *LNCS. Proc. RuSCDays*, 13708, pp. 548–561. doi: 10.1007/978-3-031-22941-1_40.

16. Prout, A., Arcand, W., Bestor, D., Bergeron, B. et al. (2016) 'Enhancing HPC security with a user-based firewall', *Proc. IEEE HPEC*, pp. 1–4. doi: 10.1109/HPEC.2016.7761641.

17. Lee, J., Kim, S., Hong, T. (2016) 'Brute-force attacks analysis against SSH in HPC multi-user service environment', *INDJST*, 9(24), pp. 1-4. doi: 10.17485/ijst/2016/v9i24/96070.

18. Stoll, C. (1988) 'Stalking the wily hacker', *Communications of the ACM*, 31(5), pp. 484–497. doi: 10.1145/42411.42412.

19. Bishop, M. (1989) 'UNIX security in a supercomputing environment', *Proc. Supercomputing'89*, pp. 693–698. doi: 10.1145/76263.76341.

20. Addai, P., Freas, R., Tesfa, E.M., Sellers, M., Mohd, T.K. (2023) 'Prevention and detection of network attacks: A comprehensive study', in *LNBIP. Proc. ICDSST*, 474, pp. 56–66. doi: 10.1007/978-3-031-32534-2_5.

21. Wilhelm, F. (2015) 'General Pr0ken Filesystem – Hacking IBM's GPFS', *ERNW INSINUATOR*, available at: https://insinuator.net/2015/04/general-pr0ken-filesystem-hacking-ibms-gpfs/ (accessed September 04, 2023).

22. Alkhalil, Z., Hewage, Ch., Nawaf, L., Khan, I. (2021) 'Phishing attacks: A recent comprehensive study and a new anatomy', *Front. Comput. Sci*., 3. doi: 10.3389/fcomp.2021.563060.

23. Liu, L., De Vel, O., Han, Q.-L., Zhang, J., Xiang, Y. (2018) 'Detecting and preventing cyber insider threats: A survey', *IEEE Communications Surveys & Tutorials*, 20(2), pp. 1397–1417. doi: 10.1109/COMST.2018.2800740.

24. Savin, G.I., Shabanov, B.M., Telegin, P.N., Baranov, A.V. (2019) 'Joint supercomputer center of the Russian Academy of Sciences: Present and future', *Lobachevskii J. Math*., 40(11), pp. 1853–1862. doi: 10.1134/S1995080219110271.

25. Baranov, A.V., Savin, G.I., Shabanov, B.M., Shitik, A.S., Svadkovskiy, I.A., Telegin, P.N. (2019) 'Methods of jobs containerization for supercomputer workload managers', *Lobachevskii J. Math*., 40(5), pp. 525–534. doi: 10.1134/S1995080219050020.

26. Reuther, A., Byun, Ch., Arcand, W. (2018) 'Scalable system scheduling for HPC and big data', *JPDC*, 111, pp. 76–92. doi: 10.1016/j.jpdc.2017.06.009.

27. Boyens, J., Smith, A., Bartol, N., Winkler, K., Hplbrook, A., Fallon, M. (2022) 'Cybersecurity supply chain risk management practices for systems and organizations', *NIST SP*, (800-161r1). doi: 10.6028/NIST.SP.800-161r1.

28. Aladyshev, O., Baranov, A., Ionin, R., Kiselev, E., Shabanov, B. (2018) 'Variants of deployment the high performance computing in clouds', *Proc. IEEE EIConRus*, pp. 1453–1457. doi: 10.1109/EIConRus.2018.8317371.

29. Martinez, J., Connor, C., Hollander, B., Paschke, K. (2020) 'HPC infrastructure security – ensuring storage and network safety, integrity, efficiency, and performance', *Proc. Virtual Conf. Tapia*, 2020. doi: 10.2172/1657098.

30. Barros, A., Chuvakin, A., Belak, A. (2019) 'Applying network-centric approaches for threat detection and response', *Gartner Research*, available at: https://www.gartner.com/en/documents/3904768 (accessed September 04, 2023).

31. Uctu, G., Alkan, M., Dogru, I.A., Dorterler, M. (2019) 'Perimeter network security solutions: A survey', *Proc. ISMSIT*, pp. 1–6. doi: 10.1109/ismsit.2019.8932821.

32. Kim D., Ryzhkov V. (2021) 'Difference between NTA, IDS, UTM, NGFW', available at: https://www.securitylab.ru/analytics/517592.php?ysclid=lia4byjqfe792657504 (accessed September 04, 2023) (in Russ.).

33. (2022) 'Network Infrastructure Security Guide. National Security Agency', *Cybersecurity Tech. Report*, available at: https://media.defense.gov/2022/Jun/15/2003018261/-1/-1/0/CTR_NSA_NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20220615.PDF (accessed September 04, 2023).

34. Vielberth, M. (2021) 'Security Information and Event Management (SIEM)', in *Encyclopedia of Cryptography, Security and Privacy*. doi: 10.1007/978-3-642-27739-9_1681-1.

35. Lazareva, N.B., Karmadonov, M.S. (2022) 'Analysis of software products of SIEM systems in order to choose a solution for the protection of enterprise information security', *Proc. Far East Math*, pp. 127–136 (in Russ.).

36. Stukalin, A.A., Nazarova, O.B. (2018) 'SIEM systems: review and comparative analysis', *Management Mechanisms Against Extremism and Terrorism Ideology: Proc. Sci. and Pract. Conf.*, pp. 116–121 (in Russ.).

37. Vishnyakova, A.N., Ryabtsev, A.A., Kondrashova, E.V., Shinakov, K.E. (2023) Review of the main SIEM systems in Russian market', *Information Security and Personal Data Protection. Problems and Solutions: Proc. Sci. and Pract. Conf.*, pp. 62–65 (in Russ.).

Авторы

Баранов Антон Викторович [1], к.т.н., доцент, ведущий научный сотрудник, antbar@mail.ru, abaranov@jscc.ru

Корепанов Павел Михайлович [1], начальник сектора информационной безопасности, kpm@jscc.ru

Кузнецов Егор Евгеньевич [1], младший научный сотрудник, egor57k@ro.ru

[1] Межведомственный суперкомпьютерный центр РАН, г. Москва, 119334, Россия

Authors

Anton V. Baranov [1], Cand. of Sci. (Engineering), Associate Professor, Leading Researcher, antbar@mail.ru, abaranov@jscc.ru

Pavel M. Korepanov [1], Head of Sector Information security, kpm@jscc.ru

Egor E. Kuznetsov [1], Junior Researcher, egor57k@ro.ru

[1] Joint Supercomputer Center of RAS, Moscow, 119334, Russian Federation

УДК 004.89 doi: 10.15827/0236-235X.142.632-643 2023. Т. 36. № 4. С. 632–643

Разработка моделей производственных и бизнес-процессов сетевых предприятий на основе многоагентных систем

Ю.Ф. Тельнов
В.А. Казаков
А.В. Данилов
А.А. Брызгалов

Ссылка для цитирования
Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Брызгалов А.А. Разработка моделей производственных и бизнес-процессов сетевых предприятий на основе многоагентных систем // Программные продукты и системы. 2023. Т. 36. № 4. С. 632–643. doi: 10.15827/0236-235X.142.632-643
Информация о статье
Поступила в редакцию: 31.08.2023 После доработки: 14.09.2023 Принята к публикации: 15.09.2023

Аннотация. В статье представлены концептуальная и поведенческие модели производственных и бизнес-процессов сетевых предприятий на основе применения многоагентных технологий, реализующих холархические и гетерархические принципы построения киберфизических производственных систем, которые обеспечивают построение и функционирование гибких и адаптивных систем управления производством. В качестве метода исследования предлагается использовать усовершенствованный метод декомпозиции многоагентных систем на автономные компоненты в соответствии с ролевым назначением агентов продуктов, агентов ресурсов и агентов платформы в производственных и бизнес-процессах. Метод разработки единой концептуальной модели позволяет интегрировать производственные и бизнес-процессы в системах управления производством сетевого предприятия на разных уровнях. Для обеспечения качества и надежности функционирования систем управления производством сетевых предприятий предлагается использовать метод формирования соглашений об уровне обслуживания. Основными результатами исследования являются построенная концептуальная модель архитектуры многоагентной системы управления производством сетевого предприятия, отражающая взаимодействие программных агентов – административных оболочек активов в соответствии с их ролями, и разработанные поведенческие модели процессов сетевых предприятий для организации бизнес-экосистемы, конфигурирования и контроля исполнения производственной цепочки с использованием языковых средств UML. В модели организации и контроля исполнения производственной цепочки введено использование соглашений об уровне обслуживания. Особенностями построенных моделей производственных и бизнес-процессов являются усиление их интеграции на различных уровнях управления производством, а также введение механизма управления качеством через аппарат создания и использования соглашений об уровне обслуживания, что позволяет повысить качество и надежность гибкого конфигурирования и исполнения производственных цепочек сетевого предприятия.

Ключевые слова: модели производственных и бизнес-процессов, индустрия четвертого поколения, бизнес-экосистема, цифровая платформа, многоагентная система, киберфизическая производственная система, система управления производством, административная оболочка актива

Благодарности. Исследование выполнено за счет гранта Российского научного фонда № 22-11-00282

Введение. Создание сетевых предприятий с помощью многоагентной технологии, реализующей принципы индустрии четвертого поколения, позволяет усилить интеграцию производственных и бизнес-процессов, их адаптивность и гибкость, а также надежность функционирования производственных систем [1–3]. Последние достижения в области развития микросервисных архитектур, применяемых в киберфизических производственных системах, способствуют повышению эффективности построения многоагентных цифровых платформ, поддерживающих функционирование распределенных бизнес-экосистем и формируемых на их основе сетевых предприятий [4, 5].

В работе [6] сформулированы требования к программной реализации цифровых платформ в соответствии с концепцией индустрии четвертого поколения, обеспечивающих создание сетевых предприятий, в частности, определены требования к созданию общеплатформенных сервисов и функциональных сервисов цифровых двойников, реализуемых в виде административных оболочек активов (AS). Вместе с тем для эффективного построения многоагентных систем в рамках реализации концепции индустрии четвертого поколения требуется достаточно детальная разработка моделей производственных и бизнес-процессов на различных стадиях жизненного цикла выпуска продукции или оказания услуг. В настоящей работе предлагаются концептуальная и поведенческие модели процесса организации бизнес-экосистемы на цифровой платформе и процессов оперативного конфигурирования и контроля исполнения производственной цепочки создания про-

дукции, что позволит повысить функциональную пригодность многоагентных технологий для построения гибких децентрализованных систем управления производственными процессами.

В организации взаимодействия интеллектуальных агентов, соответствующих различным участникам производственной цепочки, большое значение придается организации надежности доступа ко всей системе и обеспечения необходимого уровня качества исполнения производственных и бизнес-процессов на уровне как типов продуктов, так и конкретных заказов производства продукции и услуг. С этой точки зрения известно развитие инструментария автоматизированного создания и использования соглашений об уровне обслуживания (SLA, Service Level Agreement) на основе динамически формируемой информации административных оболочек активов [7, 8]. В работе [7] предлагается система iNegotiate, которая является посредником в переговорах между участниками потенциального SLA.

По мнению авторов [8], использование цифровой платформы для потребителя услуг помогает обеспечить доверие к данным и сэкономить время за счет автоматизации контроля параметров и мониторинга SLA. В этом отношении цифровая платформа системы управления производственными процессами должна обеспечивать решение задач как организации производственных цепочек, так и диагностики отклонений от нормативного выполнения соглашений об уровне обслуживания. В статье предлагается отражение в модели производственных и бизнес-процессов инструментария создания и использования соглашений об уровне обслуживания на стадиях оперативного планирования и контроля исполнения.

В качестве методов концептуального и поведенческого моделирования в работе предлагаются построение производственных и бизнес-процессов в соответствии с ролевой специализацией программных агентов, а также обеспечение надежности оперативного конфигурирования и контроля исполнения производственной цепочки за счет заключения соглашений об уровне обслуживания и контроля его исполнения, что позволит повысить функциональную пригодность многоагентных технологий для построения гибких децентрализованных систем управления производственными процессами. В данной статье представляются модели процессов, связанных с позаказным производством единичной продукции на сетевом предприятии.

Метод моделирования производственных и бизнес-процессов в многоагентной системе управления производством

Моделирование производственных и бизнес-процессов для последующей реализации в многоагентной системе управления производством выполняется с помощью языковых средств UML, позволяющих строить статические и динамические модели ПО. Разработка моделей производственных и бизнес-процессов с помощью языковых средств UML предполагает, в первую очередь, структурирование процессов в виде множества взаимосвязанных компонентов производственной системы, отображаемых в административных оболочках, а во вторую, представление этого множества в виде концептуальной модели классов, соответствующих компонентам, а также поведенческих моделей в форме диаграмм последовательностей.

Для поддержания работы отдельных программных агентов, их создания, проверки надежности, поддержки реализации связей между агентами требуется общая цифровая платформа, организованная в соответствии с требованиями облачной технологии [6], в задачи которой входит поддержание как всей бизнес-экосистемы в виде множества агентов участвующих предприятий, так и конкретных динамически создаваемых сетевых предприятий. В терминологии FIPA многоагентная платформа должна поддерживать агента платформы (AMS, Agent Management System – систему управления агентами, регистрирующую агентов на платформе), реестр активных агентов продуктов и ресурсов (активных сервисов агентов), составляющих конкретное сетевое предприятие (DF, Directory Facilitator), а также сервис передачи сообщений между агентами (MTS, Message Transport Service). Агент платформы берет на себя также функции управления предложениями ресурсов (шаблонами соглашений об уровне обслуживания, SLA-T) и собственно соглашениями об уровне обслуживания (SLA) между участниками платформы.

Ключевой вопрос в обосновании метода моделирования кибернетической части производственной системы – определение ролевой специализации программных агентов. К основным типам относят прежде всего агентов продуктов и агентов ресурсов [1, 9, 10]. Данное разделение хорошо соотносится с классификацией активов по спецификациям архитектуры RAMI [11], обслуживаемых агентами, – административными оболочками активов. По этой классификации

с точки зрения уровня вложенности элементов производственной деятельности холархии [12] выделяют продукт и его компоненты, обслуживаемые AS-агентами продуктов, а также средства производства: контроллеры, оборудование, рабочие центры, предприятие, связанный мир (сетевое предприятие – авторский термин), обслуживаемые AS-агентами ресурсов. В работе [10] рассматривается бинарное взаимодействие между агентом продукта, который отражает требования к выполнению необходимых работ и их реализации, и агентом ресурса на уровне предприятия, которое эти работы выполняет.

AS-агенты ресурсов создаются в статике и на верхнем уровне предприятий заносятся в реестр агентов платформы в момент подключения к бизнес-экосистеме платформы. AS-агенты продуктов создаются в процессе реализации проекта и отражают динамические информационные модели продукта на всех стадиях жизненного цикла и уровнях управления.

Рассмотрим схему взаимодействия агентов продуктов, агентов ресурсов и агента платформы более детально (рис. 1).

Так, на уровне головного предприятия агент продукта осуществляет декомпозицию продукта на составные части и для каждой части подбирает агента ресурса, который будет осуществлять управление созданием этой составной части. Агент ресурса, получив управление от агента продукта, создает своего агента продукта (соответствующей составной части), который фиксирует требования и результаты создания. Если для получения результата – компонента требуется декомпозиция работ, новый агент продукта рекурсивно повторяет определенную выше процедуру.

Таким образом формируется фрактальная структура производственного процесса, в которой агент продукта для компонента организует и координирует производственную цепочку агентов ресурсов, управляющих производственными операциями. Такая фрактальная структура распространяется и на сетевые предприятия, когда головное предприятие и соответствующий агент продукта играют координирующую роль, а входящие в гетерархию предприятия-участники, выпускающие составные узлы и детали, имеют соответствующих агентов ресурсов. Агенты продуктов получают состав узлов и деталей по онтологии.

Предлагаемый метод декомпозиции многоагентных систем в соответствии с ролевым назначением программных агентов усовершенствован расширением функций агента платформы за счет добавления функций администрирования предложений агентов ресурсов и исполнения соглашений об уровне обслуживания между участниками бизнес-экосистемы, а также добавлением соответствующих функций формирования и обработки соглашений агентами ресурсов и агентами продуктов, что позволяет повысить гибкость, качество и надежность выполнения производственных и бизнес-процессов сетевых предприятий.

Основные результаты построения моделей производственных и бизнес-процессов сетевого предприятия

Концептуальная модель. Представленная на рисунке 2 модель основана на спецификации из [13] и отражает основные компоненты,

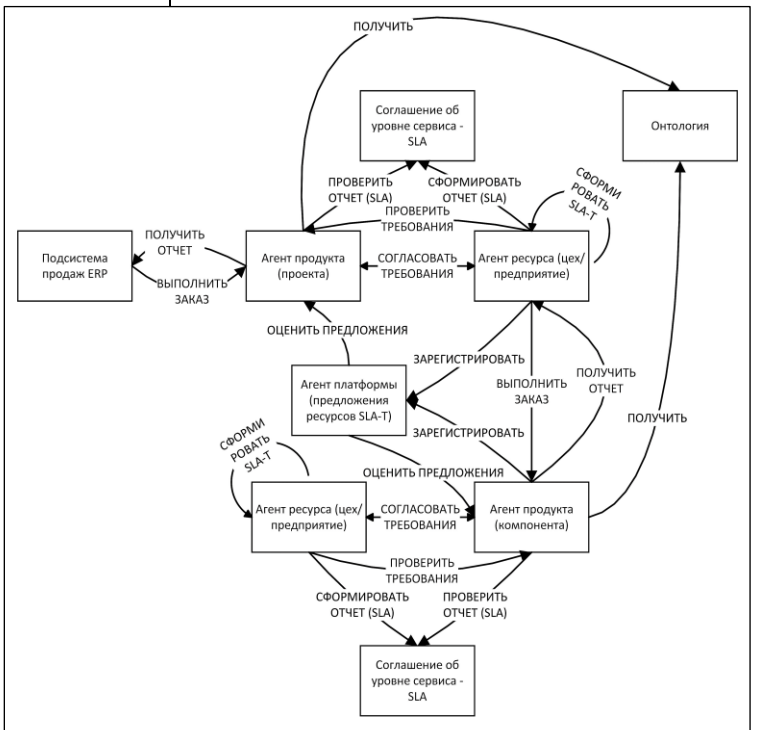


Рис. 1. Фрагмент фрактальной многоагентной системы выполнения заказов на изготовление продукции

Fig. 1. Fragment of a fractal multi-agent system for filling production orders

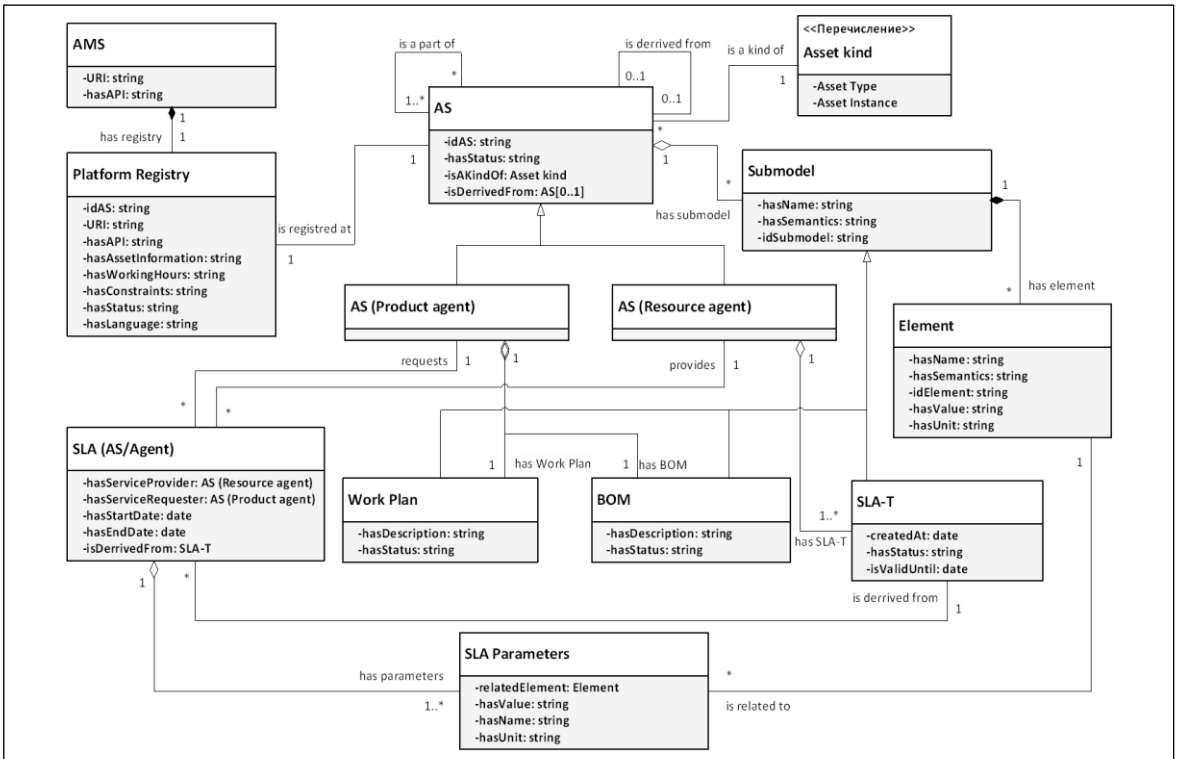


Рис. 2. Концептуальная модель архитектуры многоагентной системы управления производством сетевого предприятия

Fig. 2. Conceptual model of a multi-agent system architecture for network enterprise production management

которые расширяются за счет дополнения и уточнения части, связанной с организацией функционирования платформы как многоагентной системы, и части, описывающей характер договорных отношений о предоставлении сервиса между участниками, на основе соглашений об уровне обслуживания.

Отношение AS-агентов продуктов и ресурсов, представляющее собой соглашение об уровне обслуживания, может быть реализовано как пассивная административная оболочка, фиксирующая параметры SLA. Для формирования необходимого уровня доверия между участниками бизнес-экосистемы со стороны AS-агента продукта осуществляется контроль соблюдения SLA в рамках взаимодействия поставщиков и потребителей сервисов (видов работ) и составление рейтинга SLA на основе исторических данных.

База знаний агентов (продукта, ресурса) может структурироваться в виде подмоделей и содержать как целевые установки агентов, правила функционирования, так и статичные параметры [10].

Элементы подмоделей соответствуют как параметрам продукта, его структуре и способе производства, так и параметрам предоставления сервиса ресурсом (активом), значениям (диапазонам значений) параметров и применяемым единицам измерения. Уровни доступа к элементам подмоделей AS настраиваются, исходя из текущих потребностей участников бизнес-экосистемы, стадии жизненного цикла или хода исполнения производственного и/или бизнес-процесса.

Состав подмоделей в целом зависит от типа AS, обусловленного стадией жизненного цикла продукта. При этом можно выделить AS типа продукта (стадии замысла, проектирования) и AS конкретного экземпляра продукта (стадии планирования выпуска, производства/оказания услуг, сопровождения, списания) или партии продуктов. При сопровождении продукции данные из AS экземпляров могут использоваться для пересмотра параметров AS типа продукта. К основным подмоделям AS типа (экземпляра) продукта относятся состав работ по производству продукции и компонентная структура (Bill of materials, BOM).

Характерная особенность подмоделей AS производственных ресурсов (цехов, предприятий) заключается в формировании SLA-T, ко-

торые имеют ограниченный срок действия, что обусловлено наличием производственных планов и изменением уровня загрузки мощностей поставщика услуг, отражаемых в соответствующей административной оболочке.

SLA-T содержат элементы, относящиеся к будущим параметрам SLA. Эти элементы описывают различные характеристики, связанные с уровнем предоставления сервиса, например:

– бизнес-характеристики (назначение, цена, условия оплаты, сроки выполнения, наличие документов, предоставление гарантий, обеспечение конфиденциальности, выполнение работ в рамках законодательства, основные потребители, способ распространения, внешний вид);

– технические характеристики (необходимые функции, надежность, доступность, скорость, безопасность, время отклика);

– качественные характеристики (качество обслуживания, гибкость, масштабируемость, удобство использования, внешний вид, возможность поддержки, соответствие стандартам безопасности).

Разработанная концептуальная модель архитектуры многоагентной системы управления производством сетевого предприятия является основой для построения поведенческих моделей производственных и бизнес-процессов сетевого предприятия на различных стадиях жизненного цикла создания продукта и уровнях системы управления сетевым предприятием.

В качестве моделей основных бизнес-процессов сетевого предприятия разработаны поведенческие модели в виде диаграмм последовательностей процессов подключения нового участника к бизнес-экосистеме, конфигурирования производственной цепочки для выполнения производственного заказа и контроля за ним. Перечисленные бизнес-процессы относятся к стадии жизненного цикла, характеризующей производство продукции, и рассматриваются на уровне участников сетевого предприятия. В дальнейшем по сконфигурированным производственным цепочкам выполняются производственные процессы, которые формируют данные отчетов для контроля исполнения производственных заказов.

Модель процесса подключения нового участника к бизнес-экосистеме. Производственные и бизнес-процессы сетевых предприятий осуществляются на общей цифровой платформе, которая, как правило, поддерживается в облачной среде. Вокруг общей цифровой платформы собирается бизнес-экосистема предприятий, включающая возможных партнеров по совместному бизнесу. На основе бизнес-экосистемы, поддерживаемой цифровой платформой, создаются различные сетевые предприятия в соответствии с возникающими проектами по изготовлению новой продукции или оказанию услуг.

Провайдер цифровой платформы и соответствующий агент платформы осуществляют функции координатора бизнес-экосистемы, а также ведения репозитория соглашений об уровне обслуживания между участниками бизнес-экосистемы. Агент платформы устанавливает правила подключения к цифровой платформе, которые реализуются с помощью сервисов платформы (идентификация участников платформы, ведение профилей, проверка информационной безопасности и др.). Участники платформы вводят свои идентификационные и прочие данные профиля, предоставляют необходимые информационные ресурсы. В реестре участников бизнес-экосистемы регистрируются административные оболочки агентов ресурсов, соответствующих предприятиям, включающим предложения по выполнению различных видов деятельности в виде SLA-T.

Диаграмма последовательностей процесса подключения нового участника к бизнес-экосистеме, представленная на рисунке 3, отражает следующие шаги процесса.

Шаги 1–3. Процесс начинается с того, что потенциальный участник отправляет запрос на регистрацию агенту платформы и вносит свои идентификационные данные, которые сохраняются в реестре участников на платформе.

Шаги 4–7. Участник по запросу агента платформы передает свои организационно-юридические документы для проверки их агентом платформы на предмет достоверности и безопасности. Агент платформы проводит необходимые процедуры проверки в соответствии с правилами безопасности внутренней базы знаний и формирует решение о безопасности нового участника. В рамках данного шага возможен запрос дополнительной информации как от участника платформы, так и из внешних источников. Если участник признается небезопасным, платформа отказывает в регистрации. Результат проверки сохраняется в реестре участников на платформе.

Шаг 8. Для участника, признанного безопасным, агент платформы создает агента ресурса и передает ему управление для формирования SLA-T нового участника.

Шаги 9–12. Агент ресурса циклически формирует шаблоны соглашения в диалоге с но-

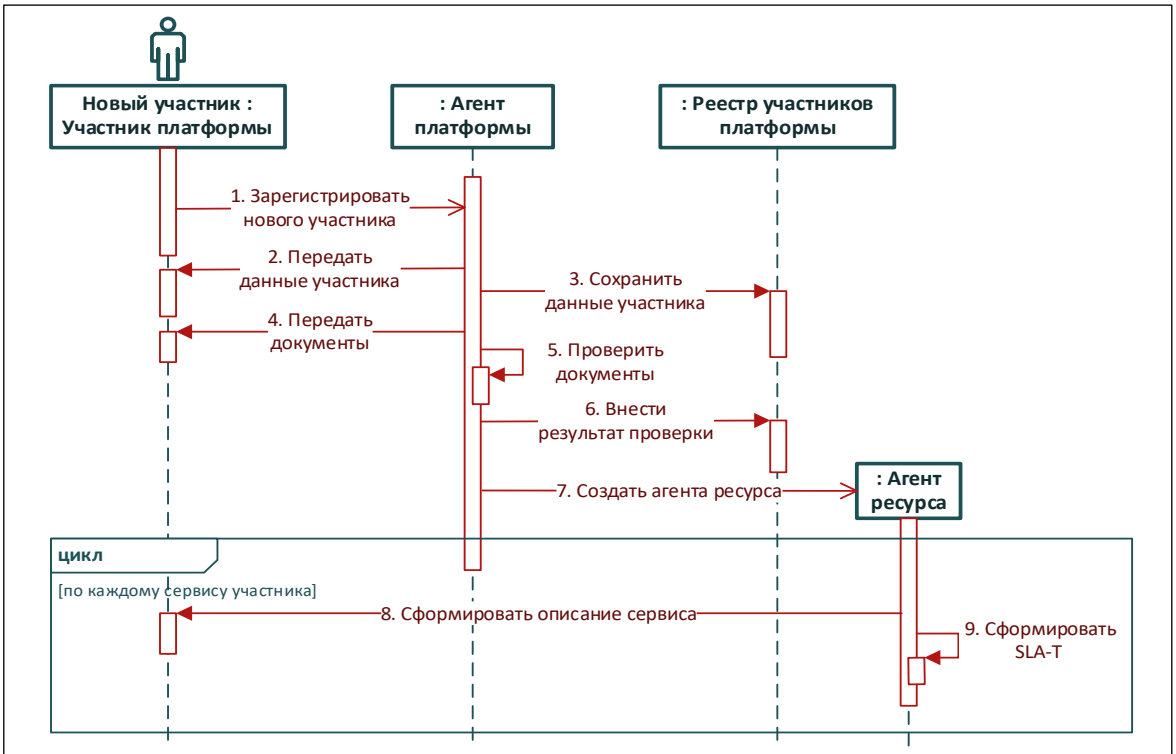


Рис. 3. Диаграмма последовательностей процесса подключения нового участника к бизнес-экосистеме

Fig. 3. Sequence diagram of connecting a new participant to a business ecosystem

вым участником на основе онтологии платформы. Результат сохраняется в подмоделях агента ресурса в соответствии с принципами построения AS.

По окончании процесса в системе сформирован новый агент ресурсов, который полностью готов к дальнейшему участию в сетевых предприятиях, формируемых на платформе.

Модель процесса конфигурирования производственной цепочки в соответствии с заказом. На платформе индустрии четвертого поколения различают процессы конструирования изделий на уровне типов продуктов и выпуска продукции на уровне экземпляров продуктов. Конструирование изделий на уровне типов означает, что разрабатывается первоначальная конструкция, которая непрерывно совершенствуется по мере производства конкретных изделий. Выпуск продукции (экземпляров продукта) предполагает организацию и планирование процесса под конкретный заказ или конфигурирование производственной цепочки, предполагающей адаптацию типовой конструкции изделия под требования конкретного заказа. В определенных случаях данный процесс в штатных ситуациях может выполняться в автоматическом режиме без непосредственного участия менеджеров, которые вмешиваются в процесс только в случае возникновения нештатных ситуаций.

Диаграмма последовательностей процесса конфигурирования производственной цепочки, представленная на рисунке 4, отражает следующие шаги процесса.

Шаг 1. Процесс начинается с поступления агенту типового продукта производственного заказа из подсистемы управления продажами ERP-системы предприятия.

Шаг 2. Агент типового продукта адаптирует типовой план работ по созданию продукта на основе значений параметров продукции, заданных в индивидуальном заказе, для дальнейшего выполнения в производственной системе сетевого предприятия.

Шаг 3. Агент типового продукта создает агента продукта (экземпляра продукта) и передает ему адаптированный план работ для дальнейшего подбора ресурсов под этот план.

Далее в рамках процесса шаги выполняются циклически для каждого шага плана работ.

Шаги 4, 5. Агент продукта передает агенту платформы запрос на подбор ресурсов для вы-

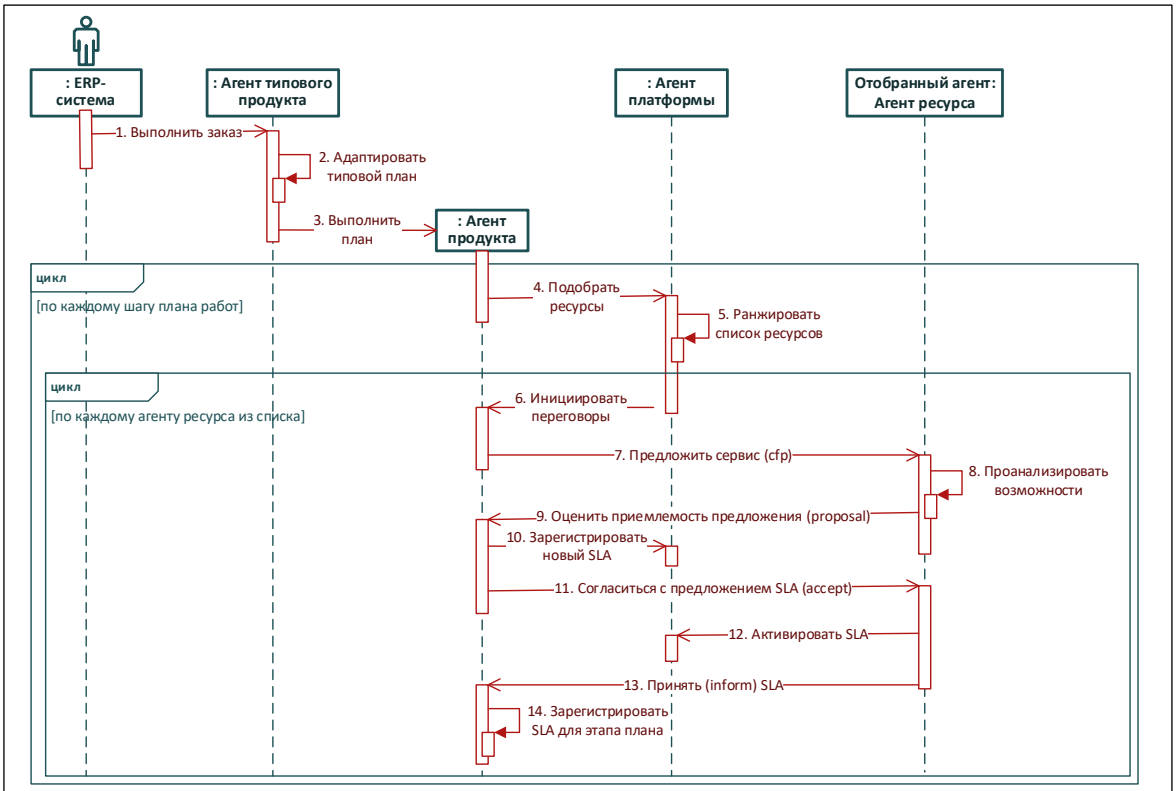


Рис. 4. Диаграмма последовательностей процесса конфигурирования производственной цепочки

Fig. 4. Sequence diagram of production chain configuration

полнения очередного шага плана. Запрос содержит описание сути работ и требования (параметры) к результату (продукту или услуге). В рамках этого агент платформы циклически анализирует и выбирает подходящих агентов ресурсов среди зарегистрированных на платформе и у каждого из них запрашивает список предлагаемых видов работ или сервисов (в виде набора SLA-T). Полученные шаблоны соглашений анализируются на предмет релевантности исходному запросу на ресурсы. Каждый релевантный SLA-T оценивается по уровню качества на основе ранее собранной статистики его выполнения в других заказах, в рамках чего агент платформы запрашивает оценки качества каждого реализованного соглашения об уровне обслуживания SLA, ранее созданного на основе данного SLA-T, и вычисляет интегральную оценку как среднее арифметическое всех оценок SLA [7].

Алгоритм оценки SLA-T выглядит следующим образом.

1. По каждому параметру *SLA* оценивается степень его фактического достижения:

$$I_{xy} = I_{xy}^{\text{факт}} \Big/ I_{xy}^{\text{план}},$$

где I_{xy} – степень достижения параметра y по SLA_x; $I_{xy}^{\text{факт}}$ – фактическое значение параметра y по SLA_x; $I_{xy}^{\text{план}}$ – плановое значение параметра y по SLA_x; $x \in X$, $y \in Y$, где X – множество *SLA* по одному *SLA-T*, Y – множество параметров по одному *SLA*.

2. Общая оценка уровня качества SLA_x (I_x) рассчитывается как среднее арифметическое всех степеней достижения параметров:

$$I_{xy} = \frac{\sum_y I_{xy}}{N},$$

где N – количество параметров SLA_x, участвующих в оценке.

3. Общая оценка уровня качества *SLA-T* ($I_{SLA\text{-}T}$) также по среднему арифметическому всех *SLA*, созданных на основе одного *SLA-T*:

$$I_{SLA-T} = \frac{\sum_x I_x}{M},$$

где M – количество *SLA* по одному *SLA-T*.

Управлением SLA занимается агент платформы, выступающий в роли посредника и гаранта уровня качества сервиса, который осу-

ществляется в рамках процесса контроля выполнения индивидуального заказа и, таким образом, на момент подбора предложений сервисов (SLA-T) от агентов ресурсов агент платформы уже имеет оценки качества ранее выполненных SLA.

На основе полученных интегральных оценок предложений сервисов (SLA-T) агент платформы строит ранжированный список соответствующих агентов ресурсов так, чтобы вверху списка оказались агенты с наивысшим уровнем сервиса (наивысшая оценка).

Далее процесс осуществляется последовательно циклически для каждого агента ресурсов в списке, то есть в порядке уменьшения оценки уровня сервиса. В рамках каждой итерации цикла агент продукта проводит переговоры с агентом ресурса на предмет заключения соглашения об уровне обслуживания (SLA) на основе стандартного протокола FIPA ACL.

Шаг 6. Агент платформы передает агенту продукта ссылку на очередного агента ресурса из ранее сформированного списка для переговоров на предмет заключения соглашения (SLA).

Шаг 7. Агент продукта отправляет запрос на выставление предложения сервиса (call-for-proposal, cfp) агенту ресурса.

Шаг 8. Агент ресурса анализирует параметры запроса на предмет возможности выполнения сервиса согласно внутренним правилам собственной базы знаний и имеющемуся плану загрузки производственных ресурсов и формирует предложение сервиса (SLA Offer), максимально близкое к исходному запросу.

Шаг 9. Агент ресурса высылает сформированное предложение (SLA Offer) сервиса агенту продукта на рассмотрение (proposal). Агент продукта анализирует приемлемость данного предложения по возможному отклонению значений параметров от исходных требований согласно правилам собственной базы знаний. Если предложение признается агентом продукта неприемлемым, агенту ресурса посылается сообщение с отказом и система переходит на следующую итерацию цикла, то есть к рассмотрению следующего по списку агента ресурса.

Шаг 10. Если предложение будет признано приемлемым, агент продукта формирует на его основе новый SLA и передает его на регистрацию агенту платформы. Регистрация на платформе означает согласие агента продукта с SLA.

Шаг 11. После регистрации агент продукта отсылает агенту ресурса сообщение со ссылкой на сформированный SLA в качестве согласия с условиями предложения (accept).

Шаги 12, 13. После получения согласия от агента продукта агент ресурса активирует SLA на платформе в качестве своего согласия с соглашением и информирует о заключении соглашения агента продукта.

Шаг 14. На заключительном шаге агент продукта регистрирует SLA в рассматриваемом шаге плана работ в качестве отметки о назначении ресурса на выполнение шага. Далее процесс подбора ресурсов выполняется для следующего шага плана работ.

Результатом процесса является сконфигурированная производственная цепочка (план работ) по созданию продукта с закрепленными ресурсами по каждому пункту плана. В дальнейшем сформированный производственный процесс осуществляется в соответствии с плановыми сроками и иными параметрами, отраженными в SLA, по результату выполнения которого проводится контроль SLA.

Модель процесса контроля выполнения производственного заказа. Процесс контроля выполнения производственного заказа осуществляется по завершении изготовления продукта (компонента продукта) и при передаче отчета со сформированными техническими параметрами заказчику. В многоагентной системе формирование отчета о выполнении производственного заказа осуществляется агентом ресурса, а проверка на соответствие требованиям заключенного соглашения об уровне обслуживания – агентом продукта. Принятие отчета о выполнении заказа вместе с оценкой качества фиксируется в соглашении об уровне обслуживания и регистрируется агентом платформы в административной оболочке, отражающей соглашения об уровне обслуживания. Диаграмма последовательностей процесса контроля выполнения производственного заказа, представленная на рисунке 5, отражает следующие шаги процесса.

Шаги 1–3. На первом этапе агент ресурса получает фактические данные в виде технических параметров от производственных линий и параметры SLA с запланированными значениями от административной оболочки соответствующего SLA. На основе этих данных формирует отчет о выполнении SLA.

Шаги 4, 5. Полученный отчет агент ресурса отправляет на согласование агенту продукта, заказавшему соответствующую работу по SLA, который, в свою очередь, сверяет данные отчета с данными собственного входного контроля результатов выполнения сервиса. Если в рамках контроля будут выявлены несоответ-

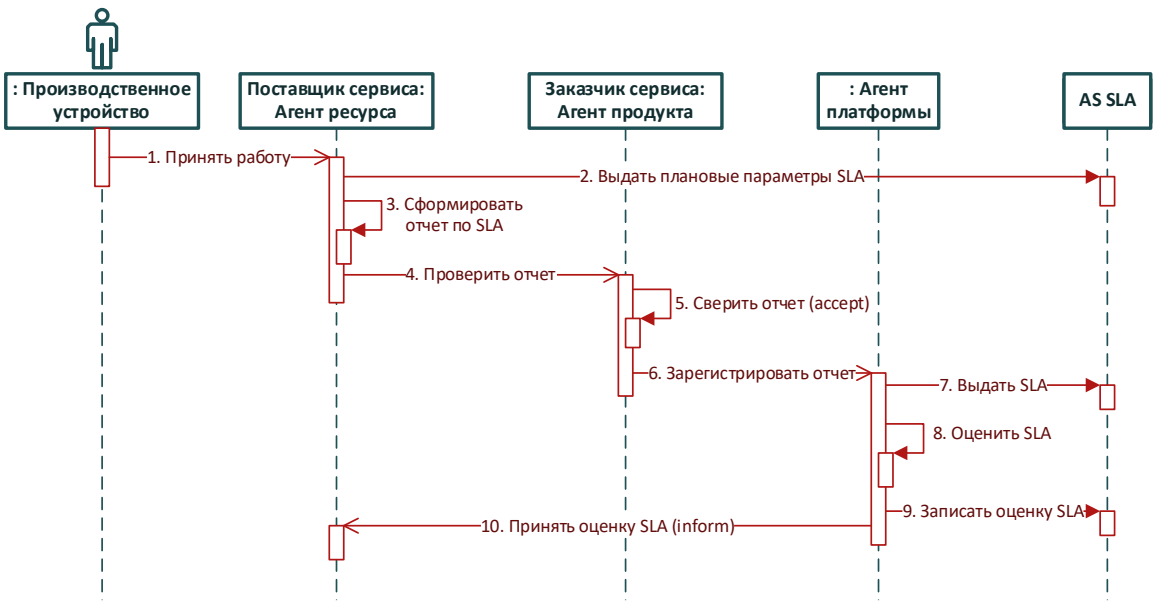


Рис. 5. Диаграмма последовательностей процесса контроля выполнения производственного заказа

Fig. 5. Sequence diagram of production order control

ствия, согласование отчета будет проходить на основе обмена информацией между агентами продукта и ресурса и проверки их точности, пока не будет достигнут консенсус.

Шаги 6–9. Согласованный отчет агент продукта регистрирует у агента платформы, который проводит сверку данных отчета с исходным SLA и формирует оценку выполнения (1-й и 2-й шаги алгоритма оценки SLA-T) как степень фактического выполнения показателей по отношению к плановому значению. Результат записывается в SLA.

Шаг 10. Агент платформы информирует агента ресурса о полученной оценке уровня сервиса выполненного производственного заказа.

Результатом описанного процесса является сформированная в SLA оценка уровня сервиса, которая в дальнейшем используется для рейтингования ресурсов на платформе при заключении новых соглашений SLA о выполнении производственных заказов.

На основе разработанных моделей бизнес-процессов в виде диаграмм последовательностей, относящихся к конфигурированию производственных цепочек создания продукции и контролю их исполнения, можно сделать вывод о функциональной способности многоагентных технологий осуществлять построение гибких систем оперативного планирования и управления позаказным производством, а также обеспечивать контроль качества выполнения производственных заказов за счет более точного согласования и оценки соглашений об уровне обслуживания между участниками сетевого предприятия.

Обсуждение

В отличие от известных работ по построению ролевой модели производственных и бизнес-процессов с использованием интеллектуальных агентов продуктов и ресурсов [10] в статье предложено развитие роли агента платформы, на которого возлагаются функции не только управления программными агентами и ведения их реестра, но и администрирования бизнес-экосистемы, подбора предложений, формируемых агентами ресурсов, для осуществления требований агента продукта, а также функций администрирования качества выполнения соглашений об уровне обслуживания, заключаемых совместно агентами продуктов и агентами ресурсов. В этом отношении предложенный в [7, 8] механизм управления соглашениями об уровне обслуживания (SLA) на стадии планирования и контроля исполнения заказов встраивается непосредственно в многоагентную систему управления производством сетевого предприятия, что позволяет повысить оперативность управления качеством и надежность построения производственных цепочек и их исполнения.

Более высокая степень гибкости конфигурирования производственных цепочек в отличие от работ [5, 10, 12] обеспечивается в результате реализации механизма адаптации типовых решений, поддерживаемых агентом типа продукта, к условиям реализации конкретного заказа, динамически формирующего административные оболочки агентов продукта (экземпляра продукта). При этом в полной мере реализуются рекурсивные процедуры разузлования конструкции изделия и динамического отбора агентов ресурсов и создания агентов продуктов (компонентов продуктов). С этой точки зрения используемые микросервисы могут базироваться как на правилах динамического реагирования на события, так и на статических правилах многокритериального принятия решений с использованием сочетания математического аппарата ресурсных вычислений и нечеткой логики на стадии оценки возможностей агентом ресурсов, а также полученного результата агентом продукта.

Заключение

В основу моделирования производственных и бизнес-процессов сетевых предприятий положены холонические и гетерархические принципы организации активов сетевого предприятия и соответствующей организации программных агентов, которые обеспечивают гибкость конфигурирования производственных цепочек под производственные заказы и оперативность их адаптации под изменения конъюнктуры рынка во внешней среде с динамическим подключением к сетевым предприятиям различных участников единой бизнес-экосистемы.

В статье развивается метод построения ролевой модели производственных и бизнес-процессов, предполагающей функциональное разделение ролей агентов продуктов, агентов ресурсов и агентов платформы, участвующих в построении производственных цепочек. Для повышения качества и надежности построения производственных цепочек предлагается метод, связанный с формированием и контролем исполнения соглашений об уровне обслуживания между предприятием, ответственным за создание продукта, и предприятием, осуществляющим производство или поставку необходимых компонентов. Эти компоненты оформляются в виде самостоятельных административных оболочек, отражающих взаимные обязательства предприятий, которые участвуют в сетевом предприятии, и накапливаемых в системе для последующего анализа эффективности работы участников бизнес-экосистемы и их выбора в новых заказах.

В статье формализованы в виде диаграммы классов концептуальная модель архитектуры многоагентной системы управления производством сетевого предприятия и в виде диаграмм последовательностей поведенческие модели процессов организации бизнес-экосистемы на основе общей цифровой платформы, конфигурирования производственной цепочки в соответствии с производственным заказом и контроля ее исполнения. В дальнейшем планируется реализация разработанных моделей в виде программного прототипа на платформе JADE в соответствии со стандартами FIPA. Реализация построенных моделей производственных и бизнес-процессов будет способствовать повышению уровня их интеграции в рамках корпоративной информационной системы, а также качества и надежности функционирования производственной системы на различных уровнях управления.

Список литературы

1. Salazar L.A.C., Ryashentseva D., Lüder A., Vogel-Heuser B. Cyber-physical production systems architecture based on multi-agent's design pattern – comparison of selected approaches mapping four agent patterns. The Int. J. of Advanced Manufacturing Tech., 2019, vol. 105, pp. 4005–4034. doi: 10.1007/s00170-019-03800-4.

2. Karnouskos S., Leitão P., Ribeiro L., Colombo A.W. Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering Industry 4.0. IEEE Industrial Electronics Magazine, 2020, vol. 14, no. 3, pp. 18–32. doi: 10.1109/MIE.2019.2962225.

3. Городецкий В.И., Бухвалов О.Л., Скобелев П.О., Майоров И.В. Современное состояние и перспективы индустриальных применений многоагентных систем // УБС. 2017. № 66. С. 94–157.

4. Belyaev A., Diedrich C. Specification "Demonstrator I4.0-Language" v3.0. Project: Industrie 4.0 Communication, 2019. URL: https://www.researchgate.net/publication/334429449_Specification_Demonstrator_I40-Language_v30 (дата обращения: 20.08.2023).

5. Sakurada L., Leitao P., De la Prieta F. Agent-based asset administration shell approach for digitizing industrial assets. IFAC PapersOnLine, 2022, vol. 55, no. 2, pp. 193–198. doi: 10.1016/j.ifacol.2022.04.192.

6. Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Денисов А.А. Требования к программной реализации системы Индустрии 4.0 для создания сетевых предприятий // Программные продукты и системы. 2022. Т. 35. № 4. С. 557–571. doi: 10.15827/0236-235X.140.557-571.

7. Li F., Cabrera C., Clarke S. A WS-agreement based SLA ontology for IoT services. In: LNISA. Proc. ICIOT, 2019, vol. 11519, pp. 58–72. doi: 10.1007/978-3-030-23357-0_5.

8. Noureddine S., Meriem B. ML-SLA-IoT: An SLA Specification and Monitoring Framework for IoT applications. Proc. ICISAT, 2021, pp. 1–12. doi: 10.1109/ICISAT54145.2021.9678460.

9. Salazar L.A.C., Vogel-Heuser B. A CPPS-architecture and workflow for bringing agent-based technologies as a form of artificial intelligence into practice. Automatisierungstechnik, 2022, vol. 70, no. 6, pp. 580–598. doi: 10.1515/auto-2022-0008.

10. Vogel-Heuser B., Ocker F., Scheuer T. An approach for leveraging Digital Twins in agent-based production systems. Automatisierungstechnik, 2021, vol. 69, no. 12, pp. 1026–1039. doi: 10.1515/auto-2021-0081.

11. Industrie 4.0. Reference Architectural Model Industrie 4.0 (RAMI4.0) – An Introduction. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf (дата обращения: 20.08.2023).

12. Komesker S., Motsch W., Popper J., Sidorenko A., Wagner A., Ruskowski M. Enabling a multi-agent system for resilient production flow in modular production systems. Procedia CIRP, 2022, vol. 107, pp. 991–998. doi: 10.1016/j.procir.2022.05.097.

13. Industrie 4.0. Details of the Asset Administration Shell, Pt 1. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf (дата обращения: 20.08.2023).

Software & Systems doi: 10.15827/0236-235X.142.632-643 2023, vol. 36, no. 4, pp. 632–643

Network enterprises: Production and business process models based on multi-agent systems

Yury F. Telnov
Vasily A. Kazakov
Andrey V. Danilov
Alexey A. Bryzgalov

For citation
Telnov, Yu.F., Kazakov, V.A., Danilov, A.V., Bryzgalov, A.A. (2023) 'Network enterprises: Production and business process models based on multi-agent systems', *Software & Systems*, 36(4), pp. 632–643 (in Russ.). doi: 10.15827/0236-235X.142.632-643
Article info
Received: 31.08.2023 After revision: 14.09.2023 Accepted: 15.09.2023

Abstract. The paper presents conceptual and behavioral models of production and business processes of network enterprises. The models are based on multi-agent technologies that implement holarchic and heterarchic principles of building cyberphysical production systems, which ensure constructing and functioning of flexible and adaptive production management systems. A research method is an improved method of decomposing multi-agent systems into autonomous components according to a role assignment of product agents, resource agents and platform agents in production and business processes. The proposed method of developing a unified conceptual model allows integrating production and business processes in production management systems of a network enterprise at different levels. To ensure reliability of production management systems in network enterprises, it is proposed to use the method of forming service level agreements. The main result of the study is a constructed conceptual model of a multi-agent production management system architecture for a network enterprise. The model reflects the interaction of software agents – administrative asset shells in accordance with their roles. Behavioral models of network enterprise processes are also proposed for organizing a business ecosystem, configuring and controlling production chain execution using UML language tools. The model of organization and control of production chain execution uses service level agreements. A specific feature of the constructed production and business process models is their integration reinforcement at various levels of production management, as well as introducing a quality management mechanism through creating and using service level agreements, which improves the quality and reliability of flexible configuration and execution of network enterprise production chains.

Keywords: fourth generation industry, business ecosystem, digital platform, multi-agent system, cyberphysical production system, production management system, asset administrative shell, production and business process model
Acknowledgements. The research was supported by grant № 22-11-00282 from the Russian Science Foundation

References

1. Salazar, L.A.C., Ryashentseva, D., Lüder, A., Vogel-Heuser, B. (2019) 'Cyber-physical production systems architecture based on multi-agent's design pattern – comparison of selected approaches mapping four agent patterns', *The Int. J. of Advanced Manufacturing Tech.*, 105, pp. 4005–4034. doi: 10.1007/s00170-019-03800-4.

2. Karnouskos, S., Leitão, P., Ribeiro, L., Colombo, A.W. (2020) 'Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering Industry 4.0', *IEEE Industrial Electronics Magazine*, 14(3), pp. 18–32. doi: 10.1109/MIE.2019.2962225.

3. Gorodetsky, V.I., Bukhvalov, O.L., Skobelev, P.O., Mayorov, I.V. (2017) 'Industrial applications of multi-agent systems: Current state and prospects', *UBS*, (66), pp. 94–157 (in Russ.).

4. Belyaev, A., Diedrich, C. (2019) *Specification "Demonstrator I4.0-Language" v3.0. Project: Industrie 4.0 Communication*, available at: https://www.researchgate.net/publication/334429449_Specification_Demonstrator_I40-Language_v30 (accessed August 20, 2023).

5. Sakurada, L., Leitao, P., De la Prieta, F. (2022) 'Agent-based asset administration shell approach for digitizing industrial assets', *IFAC PapersOnLine*, 55(2), pp. 193–198. doi: 10.1016/j.ifacol.2022.04.192.

6. Telnov, Yu.F., Kazakov, V.A., Danilov, A.V., Denisov, A.A. (2022) 'Requirements for the software implementation of the Industrie 4.0 system for creating network enterprises', *Software & Systems*, 35(4), pp. 557–571 (in Russ.). doi: 10.15827/0236-235X.140.557-571.

7. Li, F., Cabrera, C., Clarke, S. (2019). 'A WS-agreement based SLA ontology for IoT services', in *LNISA. Proc. ICIOT*, 11519, pp. 58–72. doi: 10.1007/978-3-030-23357-0_5.

8. Noureddine, S., Meriem, B. (2021) 'ML-SLA-IoT: An SLA Specification and Monitoring Framework for IoT applications', *Proc. ICISAT*, pp. 1–12. doi: 10.1109/ICISAT54145.2021.9678460.

9. Salazar, L.A.C., Vogel-Heuser, B. (2022) 'A CPPS-architecture and workflow for bringing agent-based technologies as a form of artificial intelligence into practice', *Automatisierungstechnik*, 70(6), pp. 580–598. doi: 10.1515/auto-2022-0008.

10. Vogel-Heuser, B., Ocker, F., Scheuer, T. (2021) 'An approach for leveraging Digital Twins in agent-based production systems', *Automatisierungstechnik*, 69(12), pp. 1026–1039. doi: 10.1515/auto-2021-0081.

11. *Industrie 4.0. Reference Architectural Model Industrie 4.0 (RAMI4.0) – An Introduction*, available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf (accessed August 20, 2023).

12. Komesker, S., Motsch, W., Popper, J., Sidorenko, A., Wagner, A., Ruskowski, M. (2022) 'Enabling a multi-agent system for resilient production flow in modular production systems', *Procedia CIRP*, 107, pp. 991–998. doi: 10.1016/j.procir.2022.05.097.

13. *Industrie 4.0. Details of the Asset Administration Shell, Pt 1*, available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf (accessed August 20, 2023).

Авторы

Тельнов Юрий Филиппович [1], д.э.н., профессор, заведующий кафедрой, Telnov.YUF@rea.ru
Казаков Василий Александрович [1], к.э.н., доцент, Kazakov.VA@rea.ru
Данилов Андрей Владимирович [1], старший преподаватель, Danilov.AV@rea.ru
Брызгалов Алексей Алексеевич [1], ассистент, Bryzgalov.AA@rea.ru

[1] Российский экономический университет им. Г.В. Плеханова, г. Москва, 117997, Россия

Authors

Yury F. Telnov [1], Dr.Sc. (Economics), Professor, Head of Chair, Telnov.YUF@rea.ru
Vasily A. Kazakov [1], Cand. of Sci. (Economics), Associate Professor, Kazakov.VA@rea.ru
Andrey V. Danilov [1], Senior Lecturer, Danilov.AV@rea.ru
Alexey A. Bryzgalov [1], Assistant, Bryzgalov.AA@rea.ru

[1] Plekhanov Russian University of Economics, Moscow, 117997, Russian Federation

УДК 004.021 doi: 10.15827/0236-235X.142.644-653 2023. Т. 36. № 4. С. 644–653

Моделирование структуры импортозамещения на базе модели системы оптимального распределения

Ф.Н. Абу-Абед
С.А. Жиронкин

Ссылка для цитирования
Абу-Абед Ф.Н., Жиронкин С.А. Моделирование структуры импортозамещения на базе модели системы оптимального распределения // Программные продукты и системы. 2023. Т. 36. № 4. С. 644–653. doi: 10.15827/0236-235X.142.644-653
Информация о статье
Поступила в редакцию: 29.08.2023 После доработки: 11.09.2023 Принята к публикации: 13.09.2023

Аннотация. Предметом данного исследования является модель системы оптимального распределения инвестиций в экономике, испытывающей потребность в импортозамещении в условиях внешних шоков и санкций. Необходимо построить математическую модель, связывающую отраслевую структуру инвестиций и выпуска, с учетом технологических ограничений. В настоящей статье исследуются возможности применения моделей структуры и динамики экономики для целей импортозамещения, использования в этих целях производственной функции, динамических стохастических моделей общего равновесия и байесовских векторных авторегрессий. Авторский метод, представленный в статье, основан на построении дерева решений для перебора циклов обхода – вариантов маршрута с отсечением. Метод может быть модифицирован для сокращения времени его работы. Рекомендуемая технология построения модели структуры импортозамещения сводится первоначально к построению ее целевой функции и представлению в виде графа. Впоследствии на основе структуры данных, позволяющих строить маршруты, авторами был представлен алгоритм, который является основой для реализации оптимального распределения инвестиций в отраслевой структуре и контроля их соответствия обозначенным требованиям импортозамещения. На базе данного алгоритма разработаны архитектура и структура программного средства с соответствующими классами, которое обеспечивает выходные данные в виде соответствия «инвестиции–отрасль». Полученные результаты необходимы для разработки программного средства, позволяющего решить важную отраслевую задачу моделирования структуры импортозамещения на основе распределения инвестиций по отраслям и продуктам, а также прогнозировать воздействие на этот процесс новых внешних шоков и технологических ограничений, изменения производительности труда и уровня загрузки производственных мощностей.

Ключевые слова: модель оптимального распределения, теория графов, алгоритмы оптимизации, неявный перебор, математическая модель, система, импортозамещение

Благодарности. Исследование выполнено за счет гранта Российского научного фонда № 23-28-01423, https://rscf.ru/project/23-28-01423/

Введение. В условиях усиления внешних санкций и шоков моделирование структуры импортозамещения становится все более актуальным. Эти модели должны послужить при разработке специализированного ПО для сценарного прогнозирования выпуска на внутреннем рынке.

Поиск по базе Единого реестра российских программ для ЭВМ и БД показал отсутствие программ, позволяющих моделировать различные аспекты импортозамещения. В отечественной и зарубежной литературе преобладают подходы к моделированию импортозамещения с позиции изменения экономической динамики, основанной на производственных функциях [1–3], которая является следствием, а не причиной структурных изменений. В свою очередь, в работах, посвященных моделированию структуры экономики [4, 5], практически не уделяется внимания импортозамещению. Модели импортозамещения, учитывающие структурные преобразования – фундамент устойчивого насыщения внутреннего рынка отечественной продукцией, за редким исключением не получили должного развития [6].

В публикациях, посвященных алгоритмам и ПО для моделирования отраслевых процессов в экономике, импортозамещению уделяется крайне мало внимания. Усилия авторов, как правило, сконцентрированы на анализе возможностей внутреннего рынка программных продуктов [7].

В целом в условиях усиления внешнего санкционного давления, ухудшающего структурные пропорции, моделирование импортозамещения с позиции экономической динамики теряет свою эффективность. Поэтому в данной работе предлагается моделировать импортозамещение на основе его структуры с помощью модели оптимального распределения факторов производства и отраслевого выпуска.

Для построения структурных моделей выпуска широко применяются известные динамические стохастические модели общего равновесия (DSGE) [8], а также байесовские векторные авторегрессии (BVAR) [9].

Задача моделирования структуры импортозамещения

Авторы исходили из необходимости моделирования импортозамещения как выпуска *j*-го количества продукции для внутреннего рынка в объеме *Q* в *i*-м количестве отраслей *I*. Таким образом, структура импортозамещения представлена как множество продуктов (конечным числом *m*) $Q = \{Q_1, Q_2, \ldots, Q_j, \ldots, Q_m\}$ и множества отраслей (конечным числом *n*) $I = \{I_1, I_2, \ldots, I_i, \ldots, I_n\}$.

Целевая функция модели построена на основе BVAR:

$$C(X) = \sum_{i-1}^{n} \sum_{m=1}^{j} \lambda C_{ij} + (1-\lambda)\tau_{ij}\pi_{ij}X_{ij} \to \max,$$

где C_{ij} – инвестиции в выпуск *j*-го продукта для внутреннего рынка в *i*-й отрасли; X_{ij} – доля *j*-го продукта *i*-й отрасли на внутреннем рынке; λ_{ij} – часть прироста выпуска *j*-го продукта *i*-й отрасли, связанная с технологическими инновациями; τ_{ij} – производительность труда по выпуску *j*-го продукта *i*-й отрасли для внутреннего рынка; π_{ij} – прирост занятости для выпуска *j*-го продукта *i*-й отрасли для внутреннего рынка.

Система ограничений, отражающая требования, предъявляемые к решению целевой функции модели, построена следующим образом.

Качественные ограничения, характерные для структурных моделей выпуска, рассмотренных выше, включают в себя постоянный объем сделок с каждым продуктом при замещении импорта внутренним производством, а весь объем необходимых инвестиций покрывается внутренними российскими источниками капитала. Количественные ограничения, связанные с максимальной производительностью труда, будут следующими:

$$\tau_{ij} = 1, \textit{если } \frac{c_{ij}}{c_{Q_j \min}} \geq 1, \frac{\tau_{ij}}{\tau_{Q_j дир}} \geq 1, \frac{r_{ij}}{r_{Q_j \min}} \geq 1,$$

$$\frac{p_{ij}}{p_{Q_j \min}} \geq 1,$$

где $\frac{c_{ij}}{c_{Q_j \max}} \geq 1$ гарантирует, что заявленный объем инвестиций C_{ij} в выпуск *j*-го продукта в *i*-й отрасли превысит минимальный уровень, установленный в рамках структурной политики $C_{Qj\,\min}$; $\frac{\tau_{ij}}{\tau_{Q_j дир}} \geq 1$ – производительность труда по выпуску *j*-го продукта в *i*-й отрасли превысит директивное значение, определяемое Индустрией 4.0 ($\tau_{Qj\,дир}$); $\frac{r_{ij}}{r_{Q_j \min}} \geq 1$ – ресурсов r_{ij} для выпуска *j*-го продукта в *i*-й отрасли достаточно для производства в объеме импортозамещения $r_{Qj\,\min}$; $\frac{p_{ij}}{p_{Q_j \min}} \geq 1$ – необходимый уровень загрузки производственных мощностей p_{ij} выше минимального уровня имеющегося на данный момент $p_{Qj\,min}$.

Требование обеспечения минимального объема инвестиций и учет возможности предприятий по освоению капиталовложений в установленные сроки приводят к необходимости учета дополнительных санкционных ограничений в финансовой и технологической сферах. В формализованном виде учет дальнейших санкций можно записать как

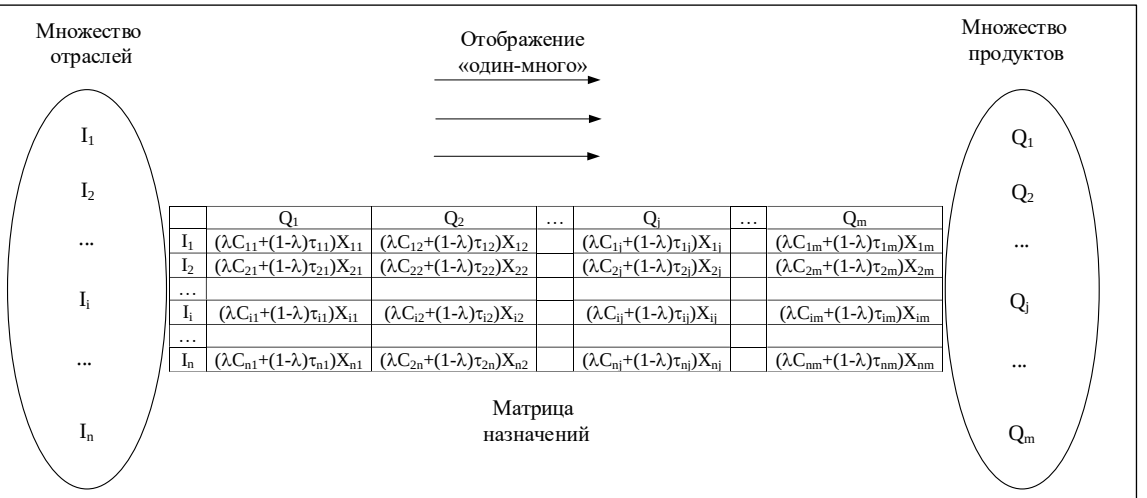
$$C_i^{\min} \leq \sum_{j=1}^{m} C_{ij}X_{ij} \leq C_i^{\max}, \; i = 1, 2, \ldots, n,$$

где $C_i^{\min}$ – минимально необходимый объем инвестиций в отрасли *i* с учетом максимального уровня санкционных финансовых ограничений; $C_i^{\max}$ – максимально возможный объем инвестиций в отрасли *i* при ослаблении санкций до их полной отмены.

Задача моделирования импортозамещения представлена на рисунке 1.

Из большого числа методов решения поставленной задачи выделим те, которые в значительной степени ей соответствуют: эвристические и генетические алгоритмы, метаэвристика, методы восхождения, ветвей и границ, полного перебора.

Наиболее емкими в плане машинного времени являются методы полного перебора, ветвей и границ. В свою очередь, генетические алгоритмы не гарантируют достижения желаемого результата при поиске оптимального решения, что может затруднить решение задачи нахождения различий оптимального и наилучшего распределений инвестиций между отраслями *i* и продуктами *j*. Это особенно важно с учетом дополнительных ограничений, заявленных в данном исследовании, которые необходимо учесть при обосновании выбора алгоритмических решений. Ограничения связаны с обязательным наличием минимального объема инвестиций в отрасли *i*, максимального уровня санкционных ограничений, а также минимально эффективного выпуска продукта *j*.



	Q_1	Q_2	…	Q_j	…	Q_m
I_1	$(\lambda C_{11}+(1-\lambda)\tau_{11})X_{11}$	$(\lambda C_{12}+(1-\lambda)\tau_{12})X_{12}$		$(\lambda C_{1j}+(1-\lambda)\tau_{1j})X_{1j}$		$(\lambda C_{1m}+(1-\lambda)\tau_{1m})X_{1m}$
I_2	$(\lambda C_{21}+(1-\lambda)\tau_{21})X_{21}$	$(\lambda C_{22}+(1-\lambda)\tau_{22})X_{22}$		$(\lambda C_{2j}+(1-\lambda)\tau_{2j})X_{2j}$		$(\lambda C_{2m}+(1-\lambda)\tau_{2m})X_{2m}$
…						
I_i	$(\lambda C_{i1}+(1-\lambda)\tau_{i1})X_{i1}$	$(\lambda C_{i2}+(1-\lambda)\tau_{i2})X_{i2}$		$(\lambda C_{ij}+(1-\lambda)\tau_{ij})X_{ij}$		$(\lambda C_{im}+(1-\lambda)\tau_{im})X_{im}$
…						
I_n	$(\lambda C_{n1}+(1-\lambda)\tau_{n1})X_{n1}$	$(\lambda C_{2n}+(1-\lambda)\tau_{2n})X_{n2}$		$(\lambda C_{nj}+(1-\lambda)\tau_{nj})X_{nj}$		$(\lambda C_{nm}+(1-\lambda)\tau_{nm})X_{nm}$

Рис. 1. Графическое представление задачи моделирования импортозамещения

Fig. 1. Graphical representation of an import substitution modeling problem

Алгоритм оперирования данными по импортозамещению

Алгоритмы нелинейной оптимизации, будучи частью численных итеративных методов, обладают определенными недостатками, ограничивающими их применение. Главным недостатком является вероятность зацикливания при поиске локального экстремума при решении задачи нахождения условного экстремума.

Поэтому при решении задачи моделирования оптимального распределения инвестиций целесообразно использовать алгоритмы, базирующиеся на графах и комбинаторике [10]. В рамках целочисленного программирования наиболее эффективной формой построения алгоритма оперирования входными данными с учетом ограничений видится представление множества отраслевых объемов выпуска каждого продукта в форме *m*-дольного графа.

Такой граф, у которого вершины каждой доли представляют собой объемы выпуска каждого продукта Q_j для внутреннего рынка в рамках импортозамещения, в общем виде представлен на рисунке 2.

Принцип работы поискового алгоритма, основанного на представлении исходных данных в виде *m*-дольного графа, отражен на рисунке 3.

Первый этап работы алгоритма – определение множества допустимых решений благодаря такому построению операторов, которое позволит отбрасывать бесперспективные начальные значения вариантов и всего множества их продолжений, за счет чего можно достичь значительной экономии вычислительных затрат.

Второй этап – сортировка списка отраслей – вершин *m*-дольного графа по каждому производимому продукту в целях импортозамещения. Такая вершина характеризуется следующим набором данных: наименование продукта I_i, требуемый объем инвестиций $C(X_{ij})$, показатель производительности труда $\tau(X_{ij})$, уровень загрузки производственных мощностей $P(X_{ij})$. Сортировка производится по увеличению объемов инвестиций. Если наиболее оптимальные варианты распределения отсутствуют, то опре-

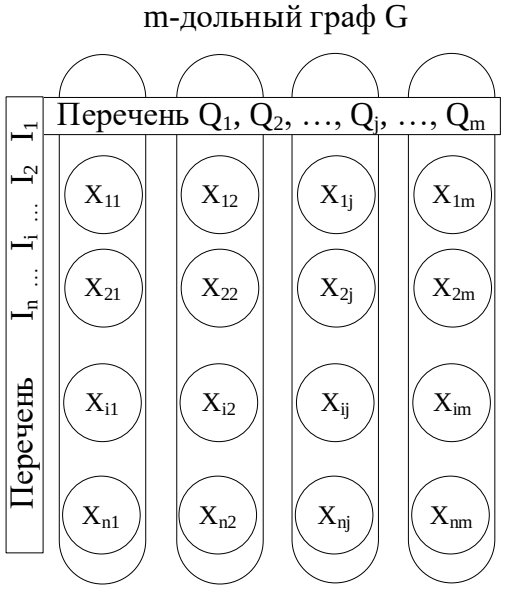


Рис. 2. Структура представления исходных данных в виде m-дольного графа

Fig. 2. Source data representation structure as an m-partite graph

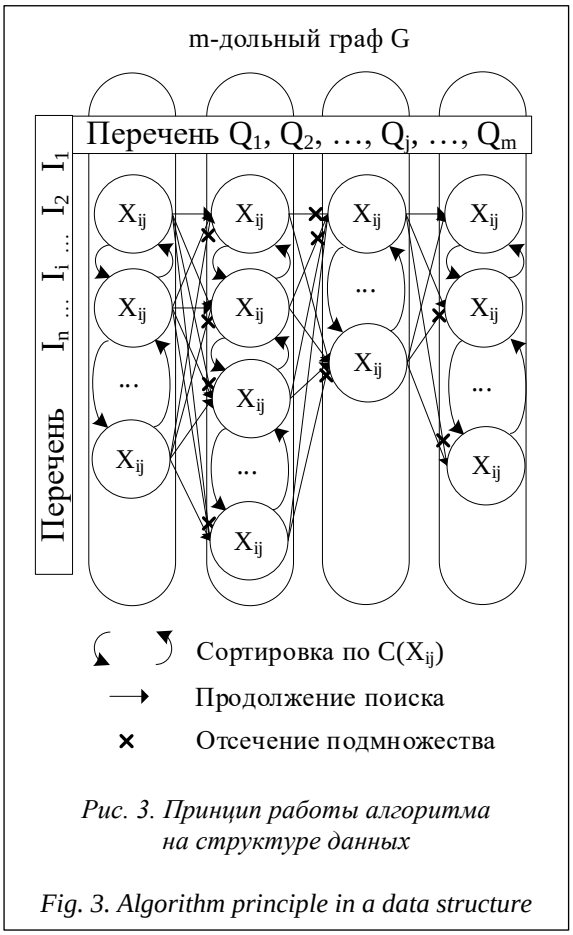


Рис. 3. Принцип работы алгоритма на структуре данных

Fig. 3. Algorithm principle in a data structure

деляются межотраслевые пропорции с более высоким уровнем инвестирования в обрабатывающих отраслях, в наибольшей степени задействованных в импортозамещении.

Третий этап – поиск в графе допустимых распределений производимых продуктов по отраслям (маршрутов) с учетом всех ограничений. В частности, ускорение поиска в глубину осуществляется обходом графа по стратегии DFS [11], что позволяет рассмотреть выпуск продуктов сначала в каждой отрасли с минимальными инвестициями, затем – находящихся в конце отсортированного списка по объему инвестиций.

При формировании алгоритма были использованы следующие вспомогательные структуры: стек для запоминания не обработанных ранее объемов выпуска, выборки продуктов, текущий формируемый список продуктов.

Поиск должен осуществляться в соответствии с графом, представленным на рисунке 3. По его окончании формируется список импортозамещающих продуктов по отраслям.

При помощи данного алгоритма дерево решений для перебора вариантов маршрута можно построить методом ветвей и границ. Сам алгоритм возможно модифицировать, отсекая частично построенные маршруты, длина которых превышает длину полного наилучшего маршрута (построенного ранее). В результате сокращается время работы алгоритма. Модифицированный таким образом алгоритм отображен на рисунке 4.

Программная реализация системы и взаимодействие с программным средством

В соответствии с требованием построения архитектуры программного средства совокупность принципов и зависимостей поведения объектов предметной области, их взаимодействие с уровнем пользовательского интерфейса и прочими сервисами приложения носят модульный характер [12].

Структура программы представляет собой несколько взаимодействующих подсистем, функции и задачи которых сведены в таблицу.

При конструировании и детальном описании структуры применяется упрощенная концепция MVC, Model-View, в которой View дополнительно выполняет задачи контроллера для реализации функционала по обработке действий пользователя, соответствующим образом меняющего модель и представление.

В качестве модели, описывающей внутреннюю структуру, используется подсистема внутреннего представления данных [13]. В роли представления выступает интерфейсная подсистема.

Таким образом, разработанное программное средство имеет модульную архитектуру, основные элементы которой представлены на рисунке 5.

Обоснование выбора практического инструментария

Реализация программного средства ведется на основании общих, функциональных, пользовательских, нефункциональных, системных и прочих требований, предъявляемых к конечному программному продукту.

Для разработки программного средства использован следующий инструментарий: кроссплатформенный фреймворк Qt, интегрированная среда разработки (IDE) Qt Creator, язык программирования C++, язык описания формата и структуры файла для хранения данных XML.

Функции модулей программного средства

Functions of software modules

Подсистема	Функция	Задача
Интерфейсная	Управление вводом данных; управление предоставлением входных и выходных данных; управление работой АС; поддержка диалога с пользователем; настройка внешнего вида	Визуальное отображение введенных и результирующих данных; предоставление форм ввода и редактирования данных; запуск и останов вычислительных процессов; настройка режимов и конфигурирование; вывод предупреждений об ошибках и недопустимых действиях; информирование о текущем состоянии АС
Сохранения и загрузки данных	Сохранение данных; загрузка данных	Построение DOM дерева для XML-формата; запись данных в файл; парсинг данных; анализ на предмет ошибок; построение внутренней структуры данных
Внутреннего представления данных	Представление данных в памяти ЭВМ	Представление данных с использованием соответствующих структур; предоставление интерфейсов доступа к данным
Анализа данных и поиска решений	Анализ соответствия отраслевого выпуска продукции потребностям импортозамещения; поиск допустимых и оптимальных назначений	Проверка соответствия отраслевого распределения капитала и рабочей силы между отраслями финансовым и технологическим секционным ограничениям; проверка технологических и инвестиционных возможностей отраслевых производств; поиск возможных назначений, удовлетворяющих ограничениям и критериям отбора, и оптимальных решений

Подход ООП и программная реализация

В соответствии с нотацией объектно-ориентированного программирования все подсистемы, модули и сущности, отражающие понятийный аппарат предметной области, приводятся в виде множества объектов, каждый из которых представляет собой экземпляр отдельного класса, а классы формируют иерархию наследования [14].

Такой подход позволяет обеспечить прозрачную декомпозицию структуры разрабатываемой системы, гибкость ее архитектуры, простоту модификации и сопровождения.

Элементы структуры программного приложения представлены следующими классами:

– Project – класс инвестиционного проекта, предназначенного для импортозамещения;

– Executor – класс, описывающий отрасль, принимающую участие в импортозамещении;

– Request – продукт, замещающий импортный аналог;

– Assignment – класс перевода отрасли на выпуск конкретного импортозамещающего продукта;

– Promodel – модель концепции MVC, обеспечивающая подсистему внутреннего представления данных об инвестиционных проектах;

– Exemodel – модель концепции MVC, обеспечивающая подсистему внутреннего представления данных об отраслях;

– Reqmodel – модель концепции MVC, обеспечивающая подсистему внутреннего представления данных о продуктах, производимых в определенных отраслях;

– Assignmodel – модель концепции MVC, обеспечивающая подсистему внутреннего представления данных о переводе отрасли на выпуск конкретного импортозамещающего продукта;

– Docwrite – класс, обеспечивающий реализацию подсистемы сохранения данных;

– Docread – класс, обеспечивающий реализацию подсистемы загрузки данных;

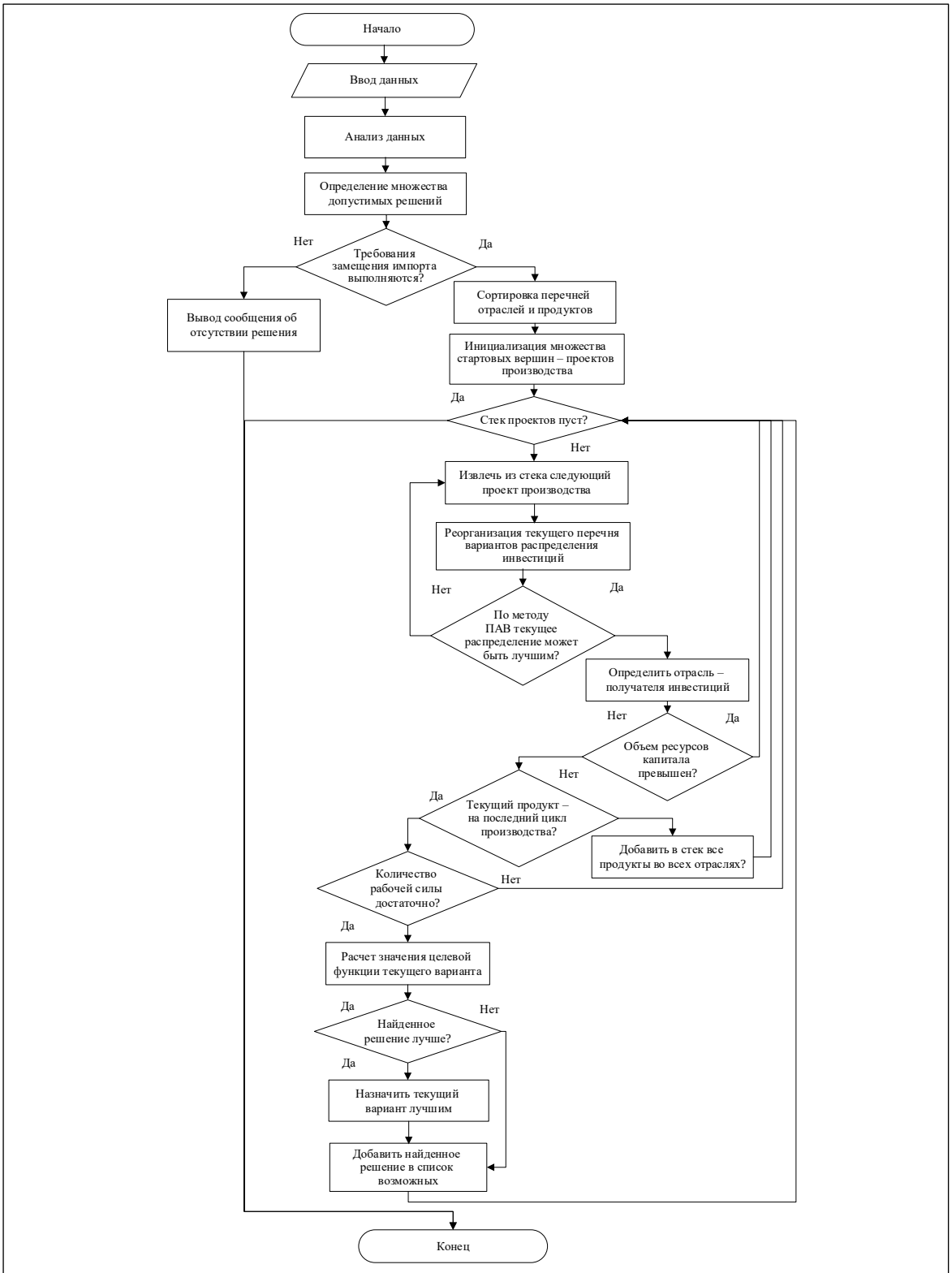


Рис. 4. Блок-схема алгоритма

Fig. 4. Algorithm flowchart

– Solutionalgorithm – класс, реализующий подсистему анализа данных и поиска решения;

– Formeditor – класс интерфейсной подсистемы, предоставляющий редактирование данных по инвестиционным проектам;

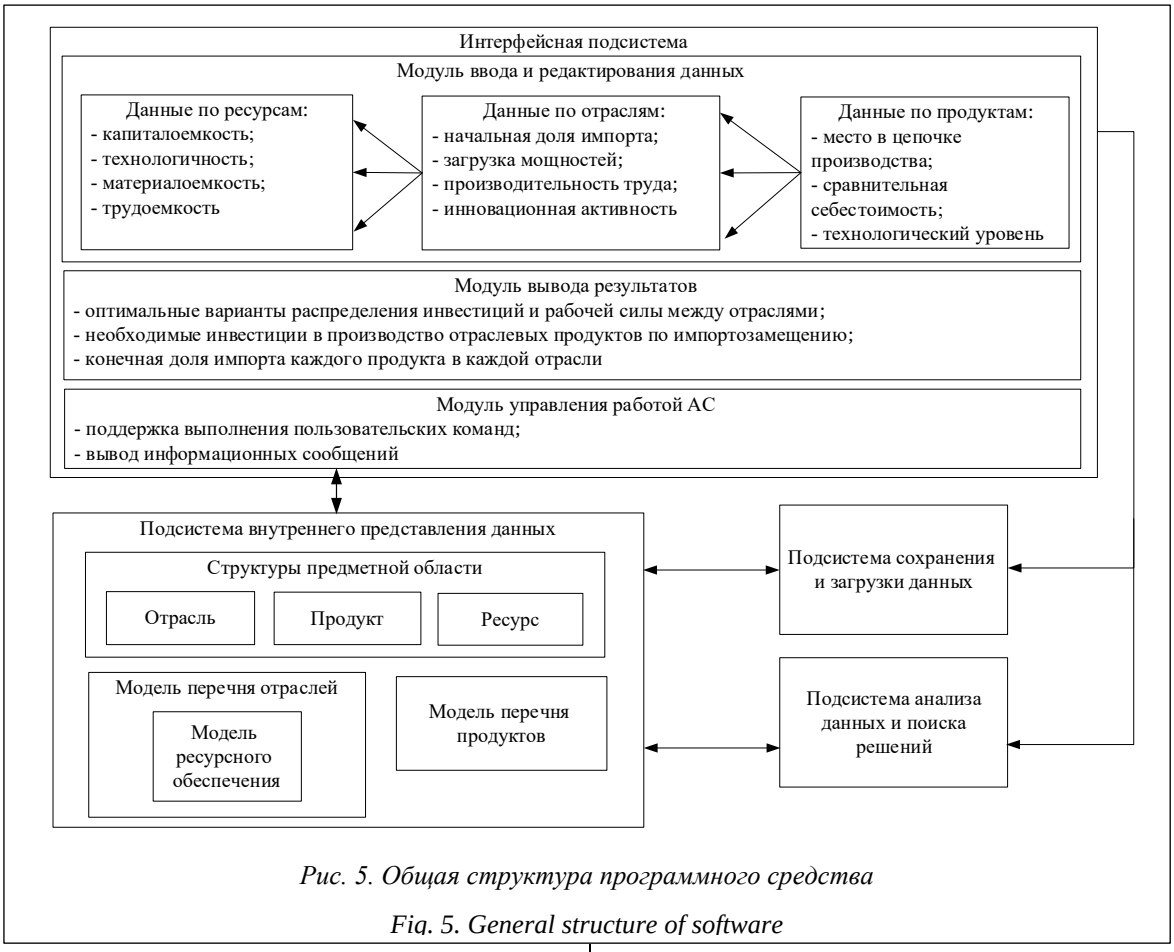


Рис. 5. Общая структура программного средства

Fig. 5. General structure of software

– Formexe – класс интерфейсной подсистемы, предоставляющий редактирование данных по отраслям и продуктам;

– TableView – модель концепции MVC и класс интерфейсной подсистемы, обеспечивающий отображение данных и поддерживающий возможности ввода и редактирования;

– Settings – класс интерфейсной подсистемы, предоставляющий форму с настройками процесса поиска решения;

– другие классы, относящиеся к интерфейсной подсистеме и обеспечивающие формы ввода, отображения и контроль правильности вводимых данных (Calendardelegate, Tabledelegate, Projectscombodelegate, Multilinedelegate).

При помощи программного средства можно решать ресурсозатратные задачи по производству сложных вычислений. Приемлемое быстродействие программного приложения определяется разделением поисковых вычислительных процессов, с одной стороны, и взаимодействием приложения с пользователем, с другой. Создание многопоточного приложения с декомпозицией потоков для поиска решения и взаимодействия программы с пользователем – наиболее эффективный подход к решению поставленной задачи [15].

При запуске процесса поиска решения оптимизационной задачи экземпляр класса Solutionalgorithm, выполняющий данные вычисления, запускается в отдельном потоке. По завершении его работы вызываются соответствующие обработчики, дальнейшие действия происходят уже в главном потоке.

Описание разработанного программного средства

Программное приложение предоставляет пользователю интерфейс главной формы, из меню которой осуществляются настройка всех параметров системы и вызов дочерних форм редактирования и ввода данных. По завершении процесса поиска оптимальных вариантов распределений пакета проектов между исполнителями на главной форме отображаются результаты.

Выходные данные наглядно представляются в виде соответствия «инвестиции–отрасль», указывается расчетная совокупная оценка долей разных отраслей в инвестициях,

производительности труда и загрузки производственных мощностей.

Если таковые отсутствуют, выводится соответствующее сообщение. Внизу окна также отображается статусная строка, сообщающая о состоянии процесса решения задачи (http://www.swsys.ru/uploaded/image/2023-4/3.jpg).

Настройки параметров осуществляются на форме, доступной через меню и панель элементов управления. Предоставляется возможность задания приоритетов в зависимости от желаемого результата – максимизация инвестиций в импортозамещающие отрасли или максимизация выпуска. Также можно задать длительность поиска оптимальных вариантов, по истечении указанного времени будут выведены все найденные решения (http://www.swsys.ru/uploaded/image/2023-4/4.jpg).

Ввод и редактирование информации выполняются через специальные формы, содержащие элементы управления, контролирующие формат вводимых данных и обеспечивающие удобство и быстроту работы (наличие выпадающих списков подстановок, календаря и др.). Табличная форма представления обеспечивает наглядность и возможность модификации области просмотра (сдвига, расширения и сужения диапазона и др.).

Сохранение и загрузка введенных данных доступны через верхнее меню главного окна или панель инструментов. При этом открывается стандартное диалоговое окно сохранения файла с возможностью выбора пути сохранения и имени.

Запуск процесса поиска оптимальных распределений осуществляется через соответствующий пункт меню. При необходимости можно принудительно прервать процесс моделирования, не выходя из программного приложения или нажав на соответствующий значок на панели инструментов.

Приложение информирует пользователя о совершении недопустимых операций: невозможности открыть, сохранить файл, работать с файлами, не созданными в данном ПО.

Предложенное ПО позволяет контролировать соответствие объемов инвестиций отраслевому выпуску с учетом требований импортозамещения и целевому уровню насыщенности технологиями, находить оптимальные распределения инвестиций в зависимости от приоритетов (рыночных, технологических, временных) и, как следствие, снижать экономические издержки на реализацию инвестиционных проектов импортозамещения.

Заключение

Построение математической модели структуры импортозамещения, основанного на системе оптимального распределения инвестиций между отраслями, призвано преодолеть ограничения существующих моделей, не учитывающих воздействие шоков и санкций. Применение модели в прогнозировании импортозамещения требует учета многих факторов, в том числе доступности технологий, загрузки производственных мощностей, производительности труда. Для этого предложен подход к построению алгоритма отраслевого распределения инвестиций, основанный на графах и комбинаторике, применительно к предложенной целевой функции, объединяющей инвестиции в выпуск импортозамещающей продукции, отраслевую структуру производства и технологий.

Разработанный алгоритм позволяет создать программное средство для оптимального распределения инвестиций, в структуру которого входят классы отраслей, импортозамещающих инвестиционных проектов, производимых продуктов, классы процесса поиска решения, подсистем интерфейса и работы с данными. Данное программное средство может быть использовано в отраслевом планировании и прогнозировании импортозамещающих инвестиций с учетом изменяющихся внешних шоков и санкций, доступа к национальным и зарубежным технологиям.

Дальнейшие исследования будут направлены на разработку программного средства, позволяющего найти оптимальное распределение продуктов по отраслям и проконтролировать соответствие инвестиций обозначенным требованиям импортозамещения.

Список литературы

1. Оленев Н.Н. Идентификация производственной функции с предельным возрастом мощностей // Математическое моделирование. 2019. Т. 31. № 11. С. 47–60. doi: 10.1134/S0234087919110042.

2. Kirilyuk I., Senko O. Verification of the returns to scale of production type for the Russian Federation regions. EPJ Web Conf. Proc. MNPS, 2019, vol. 224, art. 06011. doi: 10.1051/epjconf/201922406011.

3. Heikkinen T. A study of degrowth paths based on the von Neumann equilibrium model. J. of Cleaner Production, 2020, vol. 251, art. 119562. doi: 10.1016/j.jclepro.2019.119562.

4. Леонов Е.А., Полбин А.В. Численный поиск глобального решения в модели двухрежимной экономики с исчерпаемым запасом углеводородов // Математическое моделирование. 2021. Т. 33. № 8. С. 42–58. doi: 10.20948/mm-2021-08-03.

5. Gong D., Zheng K., Liu Y. Application of mathematical models in economic variable input and output models under the scientific visualization. Computational Intelligence and Neuroscience, 2022, vol. 1, art. 6269358, pp. 1–10. doi: 10.1155/2022/6269358.

6. Уразаева Л.Ю., Галимов И.А. Математическое моделирование импортозамещения // Вестн. Евразийской науки. 2022. Т. 14. № 3. С. 1–9.

7. Горелиц Н.К., Гукова А.С., Краснощеков Д.В. Анализ российского программного обеспечения для поддержки жизненного цикла разработки бортовых систем в условиях политики импортозамещения // Тр. ИСП РАН. 2020. Т. 32. № 2. С. 175–190 (in Eng.). doi: 10.15514/ISPRAS-2020-32(2)-14.

8. Drautzburg T. A structural approach to combining external and DSGE model forecasts. FRB of Philadelphia Working Paper, 2023, no. 23-10, pp. 1–34. doi: 10.21799/frbp.wp.2023.10.

9. Kuschnig N., Vashold L. BVAR: Bayesian vector autoregressions with hierarchical prior selection in R. J. of Statistical Software, 2021, vol. 14, pp. 1–27.

10. Dabrowski K., de Werra D., Lozin V., Zamaraev V. Combinatorics and algorithms for augmenting graphs. Graphs and Combinatorics, 2016, vol. 32, no. 4, pp. 1339–1352. doi: 10.1007/s00373-015-1660-0.

11. Пронина Е.А., Лебединская А.А., Шурхаленко П.Г. Компьютерная реализация поиска в глубину // Мировая наука. 2018. № 11. С. 234–237.

12. Елизаров Д.Э. Алгоритмизация решения задачи о размещении на основе модификации метода ветвей и границ // Вестн. ВГТУ. 2016. Т. 12. № 5. С. 42–48.

13. Кордюков Р.Ю., Допира Р.В., Иванова А.В., Абу-Абед Ф.Н., Мартынов Д.В. Модель и алгоритмизация оптимизационной задачи о назначениях в условиях дополнительных ограничений // Программные продукты и системы. 2016. Т. 29. № 2. С. 16–22. doi: 10.15827/0236-235X.114.016-022.

14. Abu-Abed F.N. Automation of work distribution in the management of mining enterprises. E3S Web of Conf., 2021, vol. 278, art. 01008. doi: 10.1051/e3sconf/202127801008.

15. Abu-Abed F.N., Ostanin O.A. Tools for forming an effective industry structure of mining region. E3S Web of Conf., 2021, vol. 315, art. 04022. doi: 10.1051/e3sconf/202131504022.

Software & Systems doi: 10.15827/0236-235X.142.644-653 2023, vol. 36, no. 4, pp. 644–653

Russian import-substitution structure based on an optimal distribution system model

Fares N. Abu-Abed
Sergey A. Zhironkin

For citation
Abu-Abed, F.N., Zhironkin, S.A. (2023) 'Russian import-substitution structure based on an optimal distribution system model', *Software & Systems*, 36(4), pp. 644–653 (in Russ.). doi: 10.15827/0236-235X.142.644-653
Article info
Received: 29.08.2023 After revision: 11.09.2023 Accepted: 13.09.2023

Abstract. The study focuses on a model of an optimal distribution system of economy investments that requires import substitution due to external shocks and sanctions. It is necessary to build a mathematical model that links a sectoral structure of investment and output in terms of technological limitations. This paper explores possibilities of using economic structure and dynamics models for import substitution purposes; using a production function for this purposes, dynamic stochastic general equilibrium models and Bayesian vector autoregressions. This paper presents author's method based on constructing a decision tree for enumerating bypass cycles – route options with a cut. The proposed method can be modified to reduce its running time. The recommended technology for constructing an import substitution structure model is initially reduced to constructing its objective function and presenting it in a graph form. Subsequently, based on a data structure for building routes, the authors present an algorithm that is a basis for implementing optimal distribution of investments in an industry structure and for monitoring their compliance with the designated import substitution requirements. This algorithm

is a basis for a software tool architecture and structure with corresponding classes, which provides output as an investment-industry correspondence. The results obtained are necessary for developing a software tool that can solve the important industry problem of modeling an import substitution structure based on investment distribution by industries and products, as well as predict the impact of new external shocks and technological restrictions on this process, changes in labor productivity and a production capacity utilization level.

Keywords: optimal distribution model, graph theory, optimization algorithms, implicit enumeration, mathematical model, system, import substitution

Acknowledgements. The study was supported by the Russian Science Foundation grant No. 23-28-01423, https://rscf.ru/project/23-28-01423/

References

1. Olenev, N.N. (2019) 'Identification of a production function with age limit for production capacities', *Math. Models and Comput. Simulations*, 31(11), pp. 47–60 (in Russ.). doi: 10.1134/S0234087919110042.

2. Kirilyuk, I., Senko, O. (2019) 'Verification of the returns to scale of production type for the Russian Federation regions', *EPJ Web Conf. Proc. MNPS*, 224, art. 06011. doi: 10.1051/epjconf/201922406011.

3. Heikkinen, T. (2020) 'A study of degrowth paths based on the von Neumann equilibrium model', *J. of Cleaner Production*, 251, art. 119562. doi: 10.1016/j.jclepro.2019.119562.

4. Leonov, E.A., Polbin, A.V. (2021) 'Numerical search for the global solution in the two-regime model with exhaustible resources', *Math. Models and Comput. Simulations*, 33(8), pp. 42–58 (in Russ.). doi: 10.20948/mm-2021-08-03.

5. Gong, D., Zheng, K., Liu, Y. (2022) 'Application of mathematical models in economic variable input and output models under the scientific visualization', *Computational Intelligence and Neuroscience*, 1, art. 6269358, pp. 1–10. doi: 10.1155/2022/6269358.

6. Urazzaeva, L.U., Galimov, I.A. (2022) 'Mathematical modeling of import substitution', *The Eurasian Sci. J.*, 14(3), pp. 1–9 (in Russ.).

7. Gorelits, N.K., Gukova, A.S., Krasnoschekov, D.V. (2020) 'Analysis of Russian software supporting onboard systems development lifecycle in the context of import substitution policy', *Proc. of ISP RAS*, 32(2), pp. 175–190. doi: 10.15514/ISPRAS-2020-32(2)-14.

8. Drautzburg, T. (2023) 'A structural approach to combining external and DSGE model forecasts', *FRB of Philadelphia Working Paper*, (23-10), pp. 1–34. doi: 10.21799/frbp.wp.2023.10.

9. Kuschnig, N., Vashold, L. (2021) 'BVAR: Bayesian vector autoregressions with hierarchical prior selection in R', *J. of Statistical Software*, 14, pp. 1–27.

10. Dabrowski, K., de Werra, D., Lozin, V., Zamaraev, V. (2016) 'Combinatorics and Algorithms for Augmenting Graphs', *Graphs and Combinatorics*, 32(4), pp. 1339–1352. doi: 10.1007/s00373-015-1660-0.

11. Pronina, E.A., Lebedinskaya, A.A., Shurkhalenko, P.G. (2018) 'Computer implementation of search into the depth', *World Sci.*, (11), pp. 234–237 (in Russ.).

12. Elizarov, D.E. (2016) 'Algorithmization the problem of placement solution on the basis of branch and bound method modification', *Vestn. of the VSTU*, 12(5), pp. 42–48 (in Russ.).

13. Kordyukov, R.Yu., Dopira, R.V., Ivanova, A.V., Abu-Abed, F.N., Martynov, D.V. (2016) 'A model and algorithmization of the assignment problem under additional constraints', *Software & Systems*, 29(2), pp. 16–22 (in Russ.). doi: 10.15827/0236-235X.114.016-022.

14. Abu-Abed, F.N. (2021) 'Automation of work distribution in the management of mining enterprises', *E3S Web of Conf.*, 278, art. 01008. doi: 10.1051/e3sconf/202127801008.

15. Abu-Abed, F.N., Ostanin, O.A. (2021) 'Tools for forming an effective industry structure of mining region', *E3S Web of Conf.*, 315, art. 04022. doi: 10.1051/e3sconf/202131504022.

Авторы
Абу-Абед Фарес Надимович [1],
к.т.н., доцент, декан,
aafares@mail.ru
Жиронкин Сергей Александрович [2],
д.э.н., профессор, zhironkin@tpu.ru

[1] Тверской государственный технический университет, г. Тверь, 170026, Россия
[2] Национальный исследовательский Томский политехнический университет, г. Томск, 634050, Россия

Authors
Fares N. Abu-Abed [1], Cand. of Sci. (Engineering),
Associate Professor, Dean,
aafares@mail.ru
Sergey A. Zhironkin [2], Dr.Sc. (Economics),
Professor, zhironkin@tpu.ru

[1] Tver State Technical University,
Tver, 170026, Russian Federation
[2] National Research Tomsk
Polytechnic University,
Tomsk, 634050, Russian Federation

УДК 81-139+81-112 doi: 10.15827/0236-235X.142.654-660 2023. Т. 36. № 4. С. 654–660

Поиск оптимального набора букв для стилевой классификации художественных текстов методом статистических индексов

Л.Г. Горбич

Ссылка для цитирования
Горбич Л.Г. Поиск оптимального набора букв для стилевой классификации художественных текстов методом статистических индексов // Программные продукты и системы. 2023. Т. 36. № 4. С. 654–660. doi: 10.15827/0236-235X.142.654-660
Информация о статье
Поступила в редакцию: 30.06.2023 После доработки: 12.07.2023 Принята к публикации: 14.07.2023

Аннотация. В статье рассматривается проблема улучшения методов стилевой классификации русскоязычных текстов. В качестве возможного направления исследований предложен метод оптимизации набора (множества) букв, применяемого для вычисления статистических индексов текстов. Для оптимизации и контроля результатов использованы поэтические и прозаические художественные тексты на русском языке. Объем текстов составлял порядка 300 тысяч знаков при оптимизации и 100 тысяч знаков при контрольной оценке. Для вычисления статистических индексов рассчитывались частотности биграмм и триграмм букв. При оптимизации опробован также и вариант совместного использования индексов биграмм и триграмм. В статье дано краткое описание метода статистических индексов, приведены применявшиеся в исследовании алгоритм пошаговой оптимизации, вид возможной оптимизационной функции и формула для нахождения границы классификации. Показано, что оптимизация набора букв улучшает классификацию по сравнению с вариантом использования как полного набора букв, так и набора из гласных букв в применении к задаче автоматического различения поэтических и прозаических художественных текстов на русском языке. Проведено сравнение результатов классификации по предложенной формуле границы классификации с результатами расчетов по классификации методом ROC-кривых. В итоге для разных сочетаний статистических индексов и способов определения границы классификации интервал верной классификации составил 72–74 % для набора, включающего все буквы, 82–86 % для набора, включающего только гласные буквы, и 80.5–92.5 % для разных наборов букв, полученных при оптимизации.
Ключевые слова: стилевая классификация, набор букв, автоматическая классификация текстов, статистический индекс, машинное обучение, метод оптимизации, ROC-кривая

Введение. Методы исследования текстов на основе расчета частотностей букв и их сочетаний были предложены очень давно [1] и пережили всплеск интереса к ним в связи с развитием компьютерных исследований [2, 3]. Подходы, основанные на графемах, естественным образом дополняют лексические, такие как анализ частотностей отдельных слов и словосочетаний [4], расчеты встречаемости определенных частей речи [5, 6]. В качестве значимых показателей исследователями рассматривались и другие свойства текстов, например, средняя длина предложения, ритмические характеристики текста [7] или индекс сжатия стандартными компьютерными алгоритмами [8]. Эти методы применялись для решения таких задач, как авторская атрибуция, стилевая классификация, определение тональности (эмоциональной окрашенности) различных текстов.

Методы, основанные на частотностях n-грамм букв, успешно применяются и в настоящее время как для русского языка [9, 10], так и для широкого спектра других языков [11, 12]. Однако такие исследования носят, скорее, феноменологический характер, поскольку теоретически обосновать значимость какого-либо сочетания букв сложнее, чем объяснить использование в тексте самостоятельных или служебных частей речи. Соответственно, и улучшение таких методов видится как экспериментально-эмпирическая задача. На той же слабости теоретического обоснования, по-видимому, основана и тенденция рассмотрения в исследованиях полного набора букв и буквосочетаний, поскольку объяснить выбор каких-то конкретных букв трудно и, как следствие, для статистического анализа все они видятся равнозначимыми.

В работах [13, 14] для вычисления статистических индексов использован набор гласных букв вследствие изначального предположения, что эти буквы несут меньше осознаваемой информативной нагрузки по сравнению с согласными. Успешное применение такого подхода (как и улучшение характеристик распознавания по сравнению с использованием полного набора букв) позволяет сформулировать проблему: можно ли, оптимизируя набор букв, используемых в вычислениях, улучшить показатели классификации? Ответу на этот вопрос и посвящена данная работа.

Используемые тексты

Для нахождения оптимального набора букв использовались русскоязычные тексты 19 авторов, имеющих достаточные объемы произведений как в поэзии, так и в прозе. Для каждого автора подбирались по два текста: один поэтический и один прозаический, что позволило выровнять выборку и уменьшить влияние индивидуальных лексических особенностей. Поскольку отдельные произведения авторов могут быть малы и недостаточны для статистического анализа (особенно это характерно для поэзии), в качестве текстов при оптимизации использовались корпуса произведений авторов. Объем каждого сборного текста был 300 тысяч символов. Использовались сочинения К.Д. Бальмонта, В.Я. Брюсова, И.А. Бунина, Д.Л. Быкова, В.С. Высоцкого, Е.А. Евтушенко, Н.М. Карамзина, И.А. Крылова, М.Ю. Лермонтова, С.Я. Маршака, В.В. Набокова, Н.А. Некрасова, Б.Ш. Окуджавы, Б.Л. Пастернака, А.С. Пушкина, К.М. Симонова, Ф.К. Сологуба, А.К. Толстого, Д.И. Хармса.

Пороги классификации вычислялись по фрагменту из первых 100 тысяч символов текстов, используемых для оптимизации.

Оценка релевантности найденных параметров классификации проводилась по массиву из 100 сборных текстов поэтического подстиля и 100 текстов прозаического подстиля разных авторов. При этом для контроля использовались тексты тех авторов, которые не входят в приведенный выше список. Каждый сборный текст принадлежал только одному автору. Объем использованных для контроля текстов составлял около 100 тысяч знаков.

Расчет статистических индексов

В качестве факторов классификации были использованы статистические индексы n-грамм букв, которые по своей сути являются интегральными показателями скоррелированности последовательно идущих букв. Чем ближе вероятности обнаружения сочетаний букв в последовательности к вероятностям для независимых событий, тем меньше величина индекса. Отметим, что *индексы биграмм* (ИБ) и *триграмм* (ИТ) букв в исследовании рассчитывались следующим образом: из текста исключались все буквы, кроме входящих в набор; оставшиеся буквы рассматривались как единая строка; заглавные и строчные варианты написания учитывались как одинаковые буквы; буква «ё» учитывалась как «е»; рассчитывалось количество двоек (или троек) последовательно идущих букв для каждого возможного сочетания; рассчитывался индекс по формуле критерия согласия Пирсона χ^2:

$$\chi^2 = \sum_{i=1}^{k} \frac{(p_i^{theor} - p_i^{emp})^2}{p_i^{theor}},$$

где p_i^{theor} – вероятность обнаружения биграмм и триграмм, вычисляемая как произведение вероятностей обнаружения отдельных букв; p_i^{emp} – вероятность, полученная из обработки реального текста; k – количество возможных сочетаний.

Алгоритм поиска

Используемый алгоритм вычисления статистических индексов не позволяет найти оптимальный набор букв простым расчетом частотностей всех их сочетаний для полного текста и последующим отбором наиболее значимых комбинаций. При таком способе расчета индексов становятся значимыми и графемы, входящие в набор, и те, которые в набор не входят и при расчете индексов пропускаются, поскольку пропуск букв порождает новые сочетания графем, не присутствующие изначально в анализируемом тексте. В то же время полный перебор всех возможных наборов букв для нахождения оптимального потребовал бы очень большого расчетного времени. Поэтому подбор в исследовании проводился пошаговым методом по следующему алгоритму.

1. Случайным образом генерировался исходный набор букв, и для него вычислялось значение оптимизационной функции.
2. На основании исходного набора формировались наборы-кандидаты путем удаления или добавления одной буквы. Для каждого такого набора рассчитывалось значение оптимизационной функции.
3. Если обнаруживалось, что существуют наборы с большим, чем в исходном наборе, значением оптимизационной функции, то наиболее удачный набор принимался в качестве текущего и алгоритм повторялся с шага 2.
4. Если текущий набор оказывался самым оптимальным из сравниваемых вариантов, значит, был достигнут локальный максимум и этот вариант запоминался.
5. Поскольку локальных максимумов в пространстве оптимизации может быть много (что подтверждалось на практике), вычисления повторялись с шага 1.

6. После накопления достаточной статистики по локальным максимумам лучший набор букв принимался как итоговый.

В качестве оптимизационной функции в исследовании использовалось значение нормированного расстояния между центрами кластеров:

$$f = \frac{2 \cdot |\bar{x}_2 - \bar{x}_1|}{sd_2 + sd_1}, \quad (1)$$

где $\bar{x}$ – средние арифметические значения индексов для каждого кластера; *sd* – их среднеквадратические отклонения. При оптимизации осуществлялся поиск максимального значения этой функции. Отметим, что при максимизации функции такого вида не только кластеры точек раздвигаются относительно друг друга, но и уменьшается разброс точек внутри кластера.

Если оптимизация осуществлялась одновременно по двум индексам (ИБ и ИТ), то общее значение функции рассчитывалось как

$$f_{\Sigma} = \sqrt{f_{\text{ИБ}}^2 + f_{\text{ИТ}}^2}, \quad (2)$$

где $f_{\text{ИБ}}$ и $f_{\text{ИТ}}$ – значения оптимизационной функции, вычисленные отдельно для индексов ИБ и ИТ.

За границу классификации принималось значение, соответствующее равенству нормированного расстояния до центров кластеров:

$$x_{\text{Г}} = \frac{\bar{x}_2 \cdot sd_1 + \bar{x}_1 \cdot sd_2}{sd_1 + sd_2}. \quad (3)$$

Для сравнения методов использовался также и другой способ классификации – метод ROC-кривых. Его принципиальное отличие в том, что для нахождения порога классификации используются не те или иные выражения для расстояний, а количество правильно классифицированных точек в обучающей выборке.

Результаты расчетов и проверка классифицирующей способности по методу нормированных расстояний

Оценка качества результатов оптимизационных процедур проводилась путем сравнения матриц ошибок (confusion matrix) классификации текстов из контрольной выборки. Основой для сравнения стали результаты расчетов для неоптимизированных наборов букв. Рассчитанные матрицы ошибок для всех букв (абвгдежзийклмнопрстуфхцчшщъыьэюя) приведены в таблице 1, расчеты для набора из гласных букв (аеиоуыэюя) – в таблице 2. Граница классификации находилась по тем же 19 парам текстов и по той же формуле, что и для оптимизированных наборов букв.

Таблица 1

Результаты классификации по набору из всех букв для индексов биграмм и триграмм

Table 1

Classification results by all letters for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	84	16	Поэзия	79	21
Проза	40	60	Проза	32	68

Таблица 2

Результаты классификации по набору из гласных букв для индексов биграмм и триграмм

Table 2

Classification results by vowels for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	74	26	Поэзия	74	26
Проза	5	95	Проза	3	97

При оптимизации только по индексу биграмм лучшие результаты были получены для набора букв «абгелмошщыэя», значение оптимизационной функции составило 3.81. Результаты классификации 200 текстов для этого набора приведены в таблице 3.

Таблица 3

Результаты классификации по набору «абгелмошщыэя» для индексов биграмм и триграмм

Table 3

Classification results by «абгелмошщыэя» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	98	2	Поэзия	95	5
Проза	17	83	Проза	12	88

При оптимизации только по индексу триграмм лучшие результаты были получены для набора букв «бгеичшы», значение оптимизационной функции равно 2.99. Результаты контрольной классификации для этого набора приведены в таблице 4.

При оптимизации по функции, учитывающей одновременно индексы биграмм и триграмм, лучшим набором оказался «абгеочш щыэя». Значение оптимизационной функции составило 3.68. Результаты классификации контрольных текстов приведены в таблице 5.

Таблица 4

Результаты классификации по набору «бгеичшы» для индексов биграмм и триграмм

Table 4

Classification results by «бгеичшы» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	97	3	Поэзия	96	4
Проза	30	70	Проза	22	78

Таблица 5

Результаты классификации по набору «абгеочшщыэя» для индексов биграмм и триграмм

Table 5

Classification results by «абгеочшщыэя» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	99	1	Поэзия	94	6
Проза	26	74	Проза	9	91

Пороговые значения при классификации, найденные по (3), для вычисленных индексов ИБ и ИТ для всех наборов букв приведены в таблице 6.

Таблица 6

Пороговые значения индексов при классификации по расстояниям для разных наборов букв

Table 6

Threshold values of indices when classifying by distances for different letter sets

Описание	**Набор букв**	**ИБ**	**ИТ**
Все буквы	абвгдежзийклмнопр стуфхцчшщъыьэюя	0.974	6.073
Гласные	аеиоуыэюя	0.022	0.073
Подбор по ИБ	абгелмошщыэя	0.131	0.418
Подбор по ИТ	бгеичшы	0.047	0.111
Подбор по ИБ и ИТ	абгеочшщыэя	0.097	0.270

Проверка классифицирующей способности при использовании метода ROC-кривых

Вторым методом нахождения порога классификации, применявшимся для оценки классифицирующей способности при использовании различных наборов букв, был метод ROC-кривых. Были использованы те же оптимизированные наборы букв и те же контрольные тексты, что и для метода классификации по нормированным расстояниям. Матрицы ошибок классификации, полученные при расчетах по этому методу, представлены в таблицах 7–11, а в таблице 12 приведены найденные пороговые значения.

Таблица 7

Результаты ROC-классификации по набору из всех букв для индексов биграмм и триграмм

Table 7

ROC classification results by all letters for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	84	16	Поэзия	66	34
Проза	36	64	Проза	22	78

Таблица 8

Результаты ROC-классификации по набору из гласных букв для индексов биграмм и триграмм

Table 8

ROC classification results by vowels for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	65	35	Поэзия	76	24
Проза	1	99	Проза	4	96

Таблица 9

Результаты ROC-классификации по набору «абгелмошщыэя» для индексов биграмм и триграмм

Table 9

ROC classification results using «абгелмошщыэя» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	100	0	Поэзия	99	1
Проза	25	75	Проза	18	82

Таблица 10

Результаты ROC-классификации по набору «бгеичшы» для индексов биграмм и триграмм

Table 10

ROC classification results by «бгеичшы» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	98	2	Поэзия	98	2
Проза	37	63	Проза	27	73

Таблица 11

Результаты ROC-классификации по набору «абгеочшщыэя» для индексов биграмм и триграмм

Table 11

ROC classification results by «абгеочшщыэя» for bigram and trigram indices

Индекс биграмм			Индекс триграмм		
	Прогноз			Прогноз	
Факт	Поэзия	Проза	Факт	Поэзия	Проза
Поэзия	99	1	Поэзия	95	5
Проза	26	74	Проза	17	83

Таблица 12

Пороговые значения индексов при ROC-классификации для разных наборов букв

Table 12

Threshold values of indices for ROC classification of different letter sets

Описание	Набор букв	ИБ	ИТ
Все буквы	абвгдежзийклмнопр стуфхцчшщъыьэюя	0.971	5.801
Гласные	аеиоуыэюя	0.021	0.074
Подбор по ИБ	абгелмошщыэя	0.137	0.440
Подбор по ИТ	бгеичшы	0.050	0.118
Подбор по ИБ и ИТ	абгеочшщыэя	0.097	0.278

Основные выводы

В работе ставились задачи определения целесообразности поиска оптимального набора графем и исследования работоспособности предложенного метода, а не получение максимально статистически обоснованного набора букв. Поэтому, хотя для подбора использовались лишь 19 пар текстов, подтверждение результативности метода на контрольной выборке в 200 текстов можно считать вполне убедительным.

Тем не менее асимметричность некоторых матриц ошибок позволяет надеяться, что с увеличением объема данных, используемых для оптимизации и вычисления пороговых значений, качество классификации еще больше возрастет.

Для удобства сравнения полученных результатов процент правильной классификации для различных наборов букв, двух индексов и двух способов классификации представлен в сводной таблице 13.

Результаты исследования позволили сделать следующие основные выводы.

Показатели классификации можно улучшить, найдя оптимальный набор букв. Предложенный метод пошаговой оптимизации применим для улучшения классификации текстов с помощью статистических индексов. Использованные в расчетах вид оптимизационной функции и формула для вычисления границы классификации показывают хорошую работоспособность. Процент правильной классификации в контрольной выборке возрастает по сравнению с классификацией как по всем буквам, так и по гласным буквам до 92.5.

Таблица 13

Процент правильной классификации для сочетания различных наборов букв, индексов и способов классификации

Table 13

Correct classification percentage for a combination of different sets of letters, indices and classification methods

Описание	Набор букв	ИБ, %	ИТ, %	ИБ, ROC, %	ИТ, ROC, %
Все буквы	абвгдежзийклмнопрстуфхцчшщъыьэюя	72.0	73.5	74.0	72.0
Гласные	аеиоуыэюя	84.5	85.5	82.0	86.0
Подбор по ИБ	абгелмошщыэя	90.5	91.5	87.5	90.5
Подбор по ИТ	бгеичшы	83.5	87.0	80.5	85.5
Подбор по ИБ и ИТ	абгеочшщыэя	86.5	92.5	86.5	89.0

Два способа нахождения пороговых значений классификации – метод, основанный на нормированных расстояниях, и метод ROC-кривых, показывают близкие результаты на контрольных выборках, однако метод ROC-кривых демонстрировал в среднем более слабые показатели качества классификации. При совместном использовании для оптимизации биграмм и триграмм результаты контрольной классификации оказались лучшими для индекса ИТ, но не самыми удачными для индекса ИБ.

Не всегда оптимальный набор, полученный для определенного индекса, являлся лучшим по результатам контрольных вычислений. Так, наборы букв, полученные при оптимизации по ИБ и ИБ+ИТ, демонстрируют лучшие результаты для индекса ИТ, чем набор, полученный при оптимизации собственно по ИТ.

В целом можно сделать заключение, что направление исследований, предложенное в работе, демонстрирует обнадеживающие результаты и может быть применимо при решении смежных задач классификации текстов.

Список литературы

1. Марков А.А. Примѣръ статистическаго изслѣдованія надъ текстомъ «Евгенія Онѣгина», иллюстрирующій связь испытаній въ цѣпь // Извѣстія Имп. Академіи Наукъ. 1913. Сер. VI. Т. 7. № 3. С. 153–162.
2. Хмелев Д.В. Распознавание автора текста с использованием цепей А.А. Маркова // Вестн. МГУ. Сер. 9: Филология. 2000. № 2. С. 115–126.
3. Keselj V., Peng F., Cercone N., Thomas C. N-gram-based author profiles for authorship attribution. Proc. PACLING, 2003, pp. 255–264.
4. Голубин Р.В., Судьин С.А., Дунаева Н.И., Афонин В.М., Ушаков А.В. Определение эмоциональной тональности текстов как инструмент социального управления: кейс COVID-19 // Теория и практика общественного развития. 2021. № 4. С. 13–19. doi: 10.24158/tipor.2021.4.1.
5. Kowsari K., Meimandi K., Heidarysafa M., Mendu S., Barnes L.E., Brown D.E. Text classification algorithms: A survey. Inf., 2019, vol. 10, no. 4, art. 150. doi: 10.3390/info10040150.
6. Батура Т.В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30. № 1. С. 85–99. doi: 10.15827/0236-235X.117.085-099.
7. Лагутина К.В., Лагутина Н.С., Бойчук Е.И. Классификация текстов по жанрам на основе ритмических характеристик // Моделирование и анализ информационных систем. 2021. Т. 28. № 3. С. 280–291. doi: 10.18255/1818-1015-2021-3-280-291.
8. Рябко Б.Я., Гуськов А.Е., Селиванова И.В. Теоретико-информационный метод классификации текстов // Проблемы передачи информации. 2017. Т. 53. № 3. С. 294–304.
9. Митин Н.А., Орлов Ю.Н. Статистический анализ биграмм специализированных текстов // Компьютерные исследования и моделирование. 2020. Т. 12. № 1. С. 243–254. doi: 10.20537/2076-7633-2020-12-1-243-254.
10. Воронина М.Ю., Кислицын А.А., Орлов Ю.Н. Алгоритм коррекции метода биграмм в задаче идентификации автора текста // Математическое моделирование. 2022. Т. 34. № 9. С. 3–20. doi: 10.20948/mm-2022-09-01.
11. Bojanowski P., Grave E., Joulin A., Mikolov T. Enriching word vectors with subword information. TACL, 2017, vol. 5, pp. 135–146. doi: 10.1162/tacl_a_00051.
12. Kruczek J., Kruczek P., Kuta M. Are n-gram categories helpful in text classification? Proc. ICCS. LNTCS, 2022, pp. 524–537. doi: 10.1007/978-3-030-50417-5_39.
13. Горбич Л.Г., Филимонов В.В., Живодёров А.А. Опыт различения поэтических и прозаических текстов на основе сравнения распределений биграмм гласных букв // Количественные методы в искусствознании: матер. Междунар. науч.-практич. конф. 2013. С. 163–166.
14. Горбич Л.Г., Живодеров А.А. Использование статистических индексов для различения научных и научно-популярных текстов на примере трудов А.Е. Ферсмана // Программные продукты и системы. 2020. Т. 33. № 4. С. 720–725. doi: 10.15827/0236-235X.132.720-725.

Software & Systems doi: 10.15827/0236-235X.142.654-660 2023, vol. 36, no. 4, pp. 654–660

Finding an optimal letter set for stylistic classification of literary texts by a statistical index method

Leonid G. Gorbich

For citation
Gorbich, L.G. (2023) ‘Finding an optimal letter set for stylistic classification of literary texts by a statistical index method’, *Software & Systems*, 36(4), pp. 654–660 (in Russ.). doi: 10.15827/0236-235X.142.654-660
Article info
Received: 30.06.2023 After revision: 12.07.2023 Accepted: 14.07.2023

Abstract. The paper concerns the problem of improving style classification methods of Russian-language texts. A method for optimizing a letter set for calculating statistical indices of texts is proposed as a possible research direction. The work used both poetic and prose literary Russian texts for optimization and monitoring the results. The volume of texts was about 300 thousand characters for optimization and about 100 thousand characters for control evaluation. Calculating statistical indices was based on calculating frequencies of letter bigrams and trigrams. Optimization also involved testing joint use of bigram and trigram indices. The paper provides a brief description of a method of statistical indices, a step-by-step optimization algorithm used in the study, a possible optimization function formula, and a formula for finding a classification boundary. It is shown that a letter set optimization improves classification in comparison with both using a full letter set

and using a vowel set, when applied to automatically distinguishing poetic and prose literary texts in Russian. Classification results are compared by the proposed classification boundary formula with ROC-curve method calculation results. Therefore, for different combinations of statistical indices and methods for determining a classification boundary, a correct classification range was: 72–74 % for an all letter set; 82–86 % for an only vowel set; 80.5–92.5 % for different sets of letters obtained during optimization.
Keywords: text style, automatic text classification, statistical index, machine learning, optimization method, ROC-curve

References

1. Markov, A.A. (1913) 'An example of a statistical study of the text "Eugene Onegin" illustrating the connection of tests in a chain', *News of the Imperial Academy of Sci. Ser. VI*, 7(3), pp. 153–162 (in Russ.).
2. Khmelev, D.V. (2000) 'Recognizing the author of the text using A.A. Markov chains', *Vestn. Moskovskogo Universiteta. Ser. 9. Philology*, (2), pp. 115–126 (in Russ.).
3. Keselj, V., Peng, F., Cercone, N., Thomas, C. (2003) 'N-gram-based author profiles for authorship attribution', *Proc. PACLING*, pp. 255–264.
4. Golubin, R.V., Sudin, S.A., Dunaieva, N.I., Afonin, V.M., Ushakov, A.V. (2021) 'The identifying the emotional content of texts as a social management tool: The case of COVID-19', *Theory and Practice of Social Development*, (4), pp. 13–19 (in Russ.). doi: 10.24158/tipor.2021.4.1.
5. Kowsari, K., Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L.E., Brown, D.E. (2019) 'Text classification algorithms: A survey', *Inf.*, 10(4), art. 150. doi: 10.3390/info10040150.
6. Batura, T.V. (2017) 'Automatic text classification methods', *Software & Systems*, 30(1), pp. 85–99 (in Russ.). doi: 10.15827/0236-235X.117.085-099.
7. Lagutina, K.V., Lagutina, N.S., Boychuk, E.I. (2021) 'Text classification by genre based on rhythm features', *Modeling and Analysis of Inform. Sys.*, 28(3), pp. 280–291 (in Russ.). doi: 10.18255/1818-1015-2021-3-280-291.
8. Ryabko, B.Ya., Gus'kov, A.E., Selivanova, I.V. (2017) 'Information-Theoretic method for classification of texts', *Problems of Inf. Transm.*, 53(3), pp. 294–304 (in Russ.).
9. Mitin, N.A., Orlov, Yu.N. (2020) 'Statistical analysis of bigrams of specialized texts', *Computer Research and Modeling*, 12(1), pp. 243–254 (in Russ.). doi: 10.20537/2076-7633-2020-12-1-243-254.
10. Voronina, M.Yu., Kislitsyn, A.A., Orlov, Yu.N. (2022) 'Algorithm of the correction of bigram method for the problem of the text author identification', *Math. Models and Comput. Simulations*, 34(9), pp. 3–20 (in Russ.). doi: 10.20948/mm-2022-09-01.
11. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017) 'Enriching word vectors with subword information', *TACL*, 5, pp. 135–146. doi: 10.1162/tacl_a_00051.
12. Kruczek, J., Kruczek, P., Kuta, M. (2022) 'Are n-gram categories helpful in text classification?', *Proc. ICCS. LNTCS*, pp. 524–537. doi: 10.1007/978-3-030-50417-5_39.
13. Gorbich, L.G., Filimonov, V.V., Zhivodyorov, A.A. (2013) 'The experience of distinguishing poetic and prose texts based on comparing the distributions of vowel bigrams', *Proc. Quantitative Methods in Art Studies*, pp. 163–166 (in Russ.).
14. Gorbich, L.G., Zhivoderov, A.A. (2020) 'Using statistical indexes to distinguish between scientific and popular science texts on the example of the works of A. E. Fersman', *Software & Systems*, 33(4), pp. 720–725 (in Russ.). doi: 10.15827/0236-235X.132.720-725.

Авторы
Горбич Леонид Геннадьевич [1],
научный сотрудник, glg@cbibl.uran.ru

[1] Центральная научная библиотека УрО РАН,
г. Екатеринбург,
620137, Россия

Authors
Leonid G. Gorbich [1],
Research Associate, glg@cbibl.uran.ru

[1] Central Scientific Library
of the Ural Branch of the RAS,
Yekaterinburg, 620137, Russian Federation

УДК 004.512 doi: 10.15827/0236-235X.142.661-667 2023. Т. 36. № 4. С. 661–667

Разработка программного модуля для решателя stressFoam платформы OpenFOAM

Д.И. Читалов

Ссылка для цитирования
Читалов Д.И. Разработка программного модуля для решателя stressFoam платформы OpenFOAM // Программные продукты и системы. 2023. Т. 36. № 4. С. 661–667. doi: 10.15827/0236-235X.142.661-667
Информация о статье
Поступила в редакцию: 15.06.2023 После доработки: 19.07.2023 Принята к публикации: 02.08.2023

Аннотация. Представленная статья отражает итоги работы по расширению перечня опций графической оболочки для платформы OpenFOAM в контексте применения программы-решателя stressFoam. Назначение программы – исследование изменения свойств твердых тел под влиянием напряжений. Предметом исследования являются процесс постановки численных экспериментов на базе решателя stressFoam и возможность его осуществления посредством графического интерфейса. Сформулирован главный недостаток среды OpenFOAM, обусловивший необходимость выполнения исследования. Дана информация о назначении программы stressFoam и основной сфере ее применения. Описан набор технологий, необходимых для реализации графической, программной части проекта и подсистемы хранения данных. Приведено обоснование применения каждой технологии, определены архитектура создаваемого продукта и среда разработки. Даны диаграммы структуры и логики работы модуля, где выделены главные программные компоненты модуля и алгоритм его применения для задач механики сплошных сред. Итогом выполненной работы является программное решение, заменяющее привычный подход применения командной строки на более эффективный, в соответствии с которым все шаги выполняются посредством графического интерфейса пользователя. Исходный код продукта размещен в репозитории сервиса GitHub для реализации доступа специалистов, применяющих OpenFOAM в экспериментах и исследованиях.
Ключевые слова: численное моделирование, программный модуль, OpenFOAM, решатель stressFoam, открытое программное обеспечение, графический интерфейс пользователя, язык программирования Python

Введение. Проводимые исследования по реализации графической оболочки для программной среды OpenFOAM [1–3] преследуют цель помочь инженерам и исследователям выполнять численное моделирование в области широкого перечня проблем *механики сплошных сред* (МСС), в том числе в задачах с деформируемыми твердыми телами. Результаты экспериментов могут применяться в конструкторских отделах машиностроительных предприятий на этапе проектирования изделий. По итоговым численным моделям определяются закономерности поведения реальных объектов в условиях влияния окружающих процессов.

Недостаток OpenFOAM заключается в отсутствии графического интерфейса пользователя, то есть набора экранных форм, посредством которых специалист управляет ходом численного эксперимента. Он включает этапы препроцессинга (задаются параметры, соответствующие условиям задачи МСС), решения (запускается процесс численного моделирования на базе одного из встроенных решателей) и постпроцессинга (результаты визуализируются, определяется степень их соответствия реальному объекту или процессу). В рамках каждого из этапов специалист вручную формирует структуру расчетного случая, создает и заполняет служебные файлы с расчетными параметрами, запускает необходимые утилиты и программы-решатели OpenFOAM. Этот подход трудоемок и сопряжен с вероятностью возникновения ошибок.

Исходя из представленных факторов, создание графической оболочки для взаимодействия с OpenFOAM позволит устранить указанную проблему. Графический интерфейс программного приложения должен соответствовать принципам дружественности, простоты, гибкости и эстетической привлекательности. Поэтому оптимальным вариантом можно считать традиционные экранные формы с элементами управления [4]. Принципиальным моментом является то, что интерфейс призван поддерживать специалиста, а не добавлять ему сложностей при работе с приложением. Он должен быть интуитивно понятным и сопровождаться простой и понятной документацией [5].

Решением данной проблемы занимались международные исследователи: как компании, специализирующиеся на создании ПО, так и независимые разработчики. Результатом стали ряд приложений с графическим интерфейсом пользователя. В частности, распространение получили линейки программных решений Visual-CFD, Salome, Helyx-OS. При этом актуальность проблемы сохраняется в силу наличия платной технической поддержки указанных программных средств, необходимости приобретения лицензии, отсутствия полноценной документации пользователя.

Российские программисты и исследователи также заинтересованы в решении указанной проблемы и предлагают собственные подходы, которые воплощаются в оригинальных графических оболочках для OpenFOAM [6–8]. В одном из таких продуктов предлагается усовершенствованная версия традиционного подхода, когда процессом численного моделирования специалист по-прежнему управляет вручную, через командную строку, но этот процесс упрощается благодаря разработанному на базе библиотеки TextControlPages консольному менеджеру.

В данном случае ценность продукта заключается в том, что ряд важных шагов пре- и постпроцессинга, а также непосредственно численного моделирования переносится на программу, однако в целом численный эксперимент, как и раньше, выполняется специалистом вручную. Поэтому данный подход, хотя и упрощает взаимодействие с OpenFOAM и повышает производительность исследователя, полностью не решает проблему. Необходим полноценный оконный интерфейс, где взаимодействие с OpenFOAM осуществлялось бы посредством привычных элементов управления. Таким образом, проблема не является полностью решенной и сохраняет свою актуальность, тем более, что новых разработок по указанному направлению начиная с 2017 года найти не удалось.

Представленное исследование направлено на разработку и интеграцию в базовую версию графической оболочки [1] нового модуля, который расширит возможности специалистов при проведении численных экспериментов на базе OpenFOAM. Созданный программный модуль позволит сократить время, затрачиваемое специалистами, и минимизировать вероятность подготовки расчетного случая с неверными входными параметрами. Этот компонент расширяет и исходный код комплекса OpenFOAM.

Предложенное приложение заменяет традиционный консольный подход, когда пользователь работает с решателем stressFOAM. Встроенная в OpenFOAM программа позволяет осуществлять линейный анализ напряжений для упругих твердых тел и применяется к стационарным и переходным задачам МСС. Программа-решатель – компонент дистрибутива OpenFOAM, который отвечает за процесс численного моделирования определенной задачи МСС.

Постановка целей и задач

Настоящее исследование направлено на замену традиционного консольного подхода при подготовке расчетного случая для работы с решателем stressFOAM на подход, предусматривающий применение графических средств управления. Под такими средствами понимаются экранные формы, панели меню и инструментов. Представленный модуль должен обеспечивать не только возможность определения исходных параметров задачи МСС посредством графических средств, но и автоматическое добавление в директорию расчетного случая служебных файлов с указанными параметрами.

Для достижения поставленной цели предполагается выполнить следующие задачи:

– разработка макетов экранных форм для служебных файлов, необходимых при работе с решателем stressFOAM;

– разработка и перенос на программный код алгоритмов, обеспечивающих валидацию расчетных параметров, которые указываются через элементы управления экранных форм, а также отвечающих за формирование в директории расчетного случая служебных файлов с расчетными параметрами задачи МСС;

– разработка и реализация механизма сохранения и восстановления исходных расчетных параметров для обеспечения возможности их дальнейшего редактирования;

– реализация и интеграция в базовую версию графической оболочки элементов управления, обеспечивающих взаимодействие пользователя с разрабатываемым модулем.

Назначение решателя stressFOAM

Как уже отмечено, stressFOAM – одна из стандартных программ дистрибутива OpenFOAM для моделирования проблем МСС. Программа специализируется на задачах в области изучения напряжений применительно к твердым телам. У ее пользователя при подготовке расчетного случая существует возможность включения или выключения опции влияния температуры. Если говорить о реальных численных экспериментах, то большее внимание уделяется именно изотермическому механическому напряжению, а температурные воздействия не учитываются.

Перед применением программы-решателя пользователь выполняет построение расчетной сеточной модели. Это обязательный этап эксперимента, относящийся к препроцессингу решения. В ходе построения расчетных сеток определяются геометрические модели тел, в данном случае речь идет о твердых телах. Для

задач МСС применяются различные типы сеток, но чаще блочные. В OpenFOAM такие сетки создаются средствами стандартной утилиты blockMesh.

Традиционная сфера применения решателя stressFOAM – проблемы в области гидродинамики, например, изучение особенностей взаимодействия морского дна, потоков воды и металлических конструкций. Эти вопросы исследуются в морской инженерии, а также при проектировании подводных аппаратов. Специалистами выявляются возможные изменения характеристик тел, в том числе твердости и прочности. Полученные результаты применяются в ходе проектирования продукции для обеспечения необходимых качественных характеристик.

Технологии разработки

Каждый программный продукт получает свою реализацию в виде набора скриптов. Они описывают разработанные алгоритмы с помощью выражений конкретного языка программирования, который выбирается перед созданием программного продукта. На этом этапе также определяются технология реализации графического интерфейса, способ хранения данных, среда разработки, архитектура будущего продукта. В данном случае программный модуль будет иметь многослойную архитектуру и функционировать в качестве настольного программного средства.

- *Язык программирования логики*. Поскольку представленный модуль расширяет возможности базовой версии графической оболочки, применяемый язык должен обладать гибкостью и масштабируемостью. Логику программного модуля решено реализовать средствами высокоуровневого языка Python 3.7. Визуальная часть приложения, сформированная за счет Python, не требует установки и настройки сервера [9].
- *Технология программирования графического интерфей*са. Важно применить библиотеку, поддерживающую синтаксис выбранного языка программирования и при этом позволяющую создавать привычные оконные интерфейсы и сопровождаться подробной документацией. Одной из таких библиотек для Python является PyQt. Важная особенность PyQt в том, что для построения экранных форм применяется Qt Designer, который позволяет формировать структуры различной сложности простым перетаскиванием виджетов в рабочую область. Экранные формы, созданные средствами PyQt, обладают возможностью динамического изменения характеристик в зависимости от параметров монитора и типа ОС [10–12].
- *Технология реализации подсистемы хранения данных*. Все указываемые пользователем расчетные параметры должны быть определенным образом сохранены и в последующем восстановлены. Оптимальным вариантом будет применение реляционных БД. Для настольных программных продуктов применяются БД формата SQLite.
- *Среда разработки*. Это оконное программное средство, которое обеспечивает не только набор кода, но и его отладку, тестирование, а также запуск всего проекта для оценки работоспособности. Для приложений на Python популярной средой разработки является PyCharm.
- *Операционная система*. Дистрибутивы OpenFOAM доступны для всех основных видов операционных систем, в том числе Windows, Linux, macOS. Предложенное программное решение также поддерживается на указанных платформах и может загружаться в формате whl-пакета или через исполняемый файл.

Структура модуля

Представленный программный модуль состоит из ряда файлов, содержащих программный код скриптов (рис. 1). Это не окончательный набор, поскольку модуль предполагает расширение функциональных возможностей и подключение новых блоков. Файлы объединены в служебные директории. Для модуля реализованы корневая директория и запускаемый файл.

Корневая директория, помимо запускаемого файла (run.py), содержит несколько вложенных директорий. Директория windows содержит макеты окон для экранных форм. По сути для каждой формы создается оболочка (окно-виджет, диалоговое окно и т.д.). В директории forms для каждой формы определяется структура элементов управления. Эти элементы привязываются к окну, и получается готовая экранная форма. Директория threads содержит файлы со скриптами запуска генерации расчетных сеток, программ-решателей, отвечающих непосредственно за численное моделирование, утилит визуализации результатов. В директории functions находятся файлы с кодом служебных функций, отвечающих, например, за переключение окон графической оболочки, вывод служебных сообщений и др.

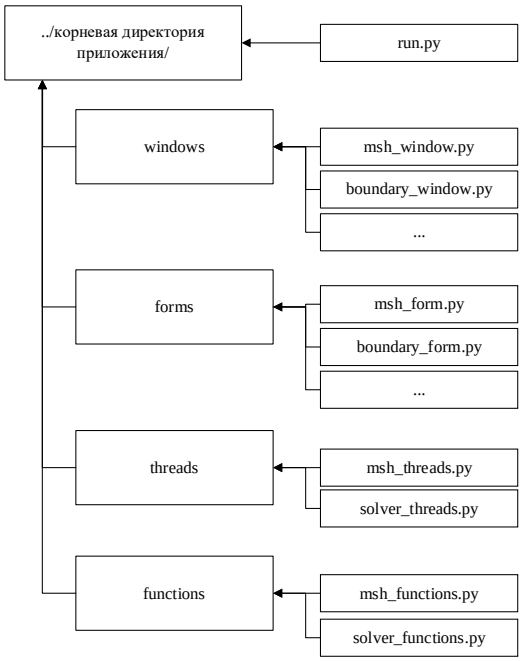


Рис. 1. Диаграмма структуры модуля

Fig. 1. Module structure diagram

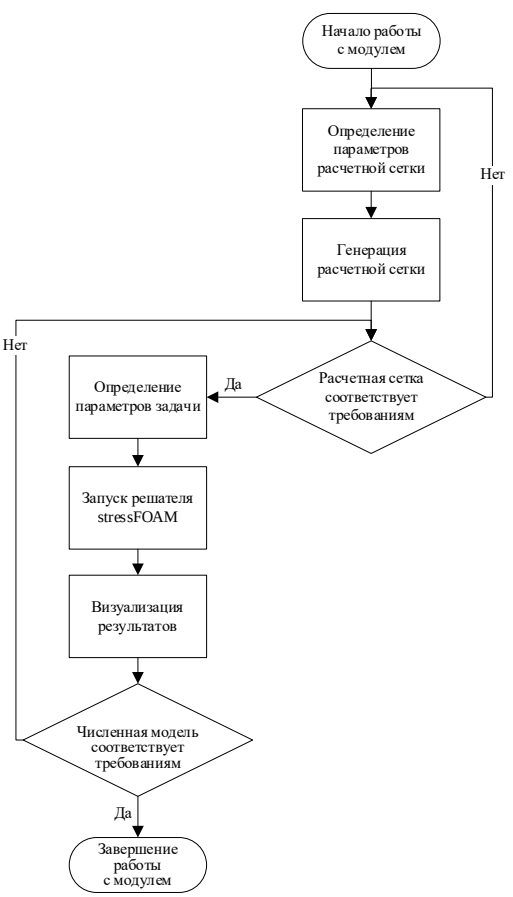


Рис. 2. Диаграмма логики работы модуля

Fig. 2. Module operation logic diagram

Логика работы модуля

Работа с представленным модулем включает этапы препроцессинга, решения и постпроцессинга (рис. 2).

После запуска модуля специалист посредством экранной формы задает параметры расчетной сетки и запускает одну из утилит генерации сеточной модели, например, blockMesh. Далее посредством пакета визуализации научных результатов ParaView определяется степень соответствия расчетной сетки заявленным требованиям. В случае недостаточного соответствия специалист может вновь перейти к редактированию параметров сетки и выполнить повторную генерацию. При положительном результате он переходит к определению параметров самой задачи МСС и запуску программы-решателя stressFoam. При этом формируется структура расчетного случая с необходимыми служебными файлами. Результаты научного эксперимента визуализируются также посредством пакета ParaView и при необходимости корректируются (постпроцессинг).

Проверка работы модуля

Представленный модуль разработан с помощью свободно распространяемых технологий и не требует приобретения лицензии. Он расширяет исходный код OpenFOAM и может дополняться новыми компонентами. Актуальная версия модуля размещена в репозитории сервиса GitHub (URL: https://github.com/DmitryChitalov/OpenFOAM_GUI). Приложение может применяться инженерами предприятий машиностроения, использующими OpenFOAM при проектировании продукции.

Сфера применения модуля – эксперименты в области анализа напряжений применительно к твердым телам. Работа модуля протестирована на примере одной из классических задач МСС – plateHole, в которой твердое тело – квадратная пластина с круглым отверстием по центру. Пластина нагружена горизонтальной силой натяжения с правой стороны. Главное окно приложения после завершения эксперимента приведено на рисунке 3.

Дискуссия

Существующие приложения-аналоги обладают несколько избыточным интерфейсом.

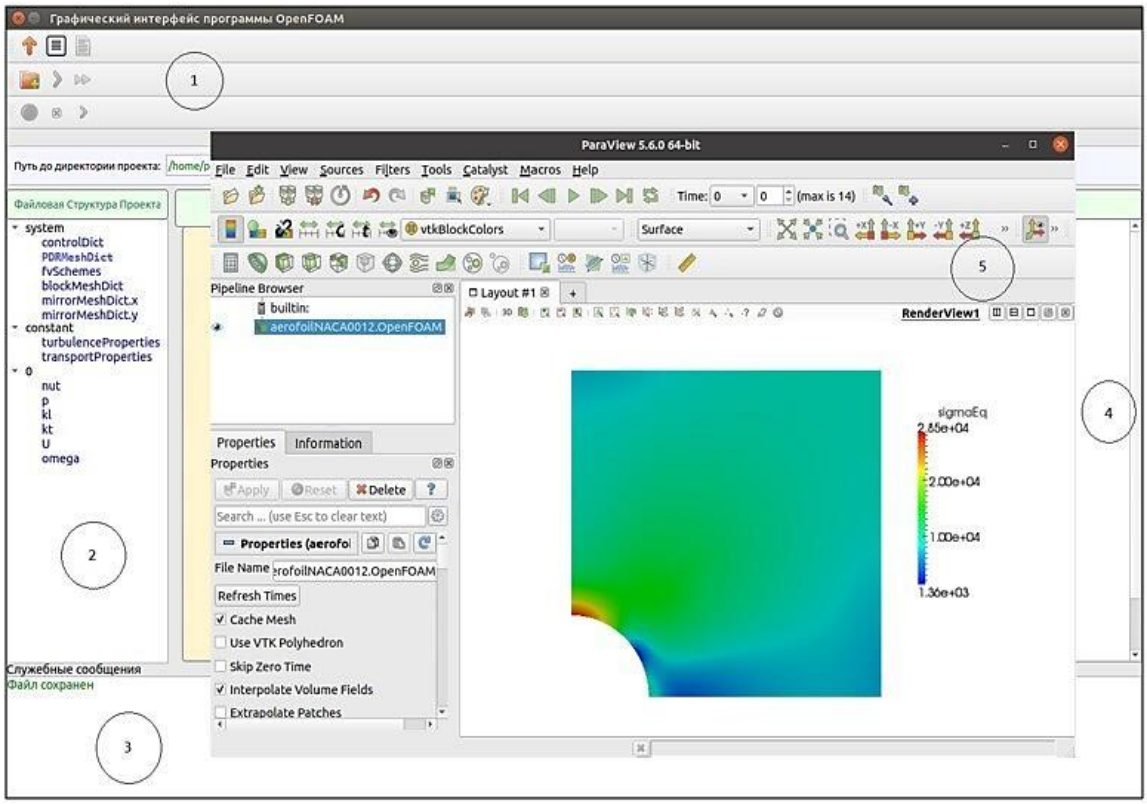


Рис. 3. Главное окно модуля после завершения эксперимента:
1 – панель управления моделированием; 2 – компоненты директории расчетного случая;
3 – строка состояния; 4 – панель ввода расчетных параметров; 5 – окно системы ParaView

Fig. 3. Main module window after completing an experiment:
1 – modeling control panel; 2 – components of a design case directory; 3 –status bar;
4 –panel for entering calculation parameters; 5 – ParaView system window

Для их освоения требуются значительное время, подробная документация, а зачастую и помощь специалистов. Графическая оболочка, предложенная автором, отличается тем, что реализована по модульному принципу, то есть для расширения возможностей подключаются дополнительные модули. В этой связи интерфейс представленного продукта можно охарактеризовать как интуитивно понятный, не требующий предварительной подготовки для освоения.

Кроме того, благодаря модульному принципу можно говорить о большом потенциале дальнейших исследований в указанной области. Предполагаются разработка и интеграция в базовую версию графической оболочки новых модулей, которые расширят возможности специалистов в части препроцессинга (увеличение списка доступных утилит для генерации расчетных сеток), решения (доступ к большему числу программ-решателей для моделирования новых проблем МСС), постпроцессинга (увеличение списка утилит для исследования полученных результатов).

Заключение

Результатом выполненной работы является программный модуль, расширяющий возможности численного моделирования задач МСС на базе комплекса OpenFOAM. Модуль предоставляет пользователю графические и программные (скрипты) средства, обеспечивающие создание расчетного случая для экспериментов в области анализа напряжений применительно к твердым телам. В OpenFOAM подобные проблемы МСС моделируются с помощью программы-решателя stressFoam.

Были предложены и реализованы в графической и программной составляющей модуля следующие подходы:

– отделение исходного кода окон экранных форм от кода элементов управления форм;

– применение bash-скриптов для запуска утилит и программ-решателей OpenFOAM;

– механизм проверки типов данных, вводимых через экранные формы;

– механизм проверки комплектности расчетного случая.

Представленный модуль заменяет традиционный, ручной подход к созданию расчетного случая и применению командной строки на более удобный и эффективный, когда все шаги численного эксперимента осуществляются с помощью привычных экранных форм. Помимо улучшения удобства, предполагаемая практическая ценность заключается в снижении временных затрат специалиста.

Список литературы

1. Читалов Д.И., Меркулов Е.С., Калашников С.Т. Разработка графического интерфейса пользователя для программного комплекса OpenFOAM // Программная инженерия. 2016. Т. 7. № 12. С. 568–574. doi: 10.17587/prin.7.568-574.

2. Читалов Д.И., Калашников С.Т. Разработка приложения для подготовки расчетных сеток с градуирующими и изогнутыми краями для программной среды OpenFOAM // Системы и средства информатики. 2018. Т. 28. № 4. С. 122–135. doi: 10.14357/08696527180412.

3. Читалов Д.И., Калашников С.Т. Разработка приложения для подготовки расчетных сеток посредством утилиты snappyHexMesh программной среды OpenFOAM // Программные продукты и системы. 2018. Т. 31. № 4. С. 715–722. doi: 10.15827/0236-235X.124.715-722.

4. Салий Т.М. Принципы создания пользовательского интерфейса // Вестн. ИнЕУ. 2016. № 3. С. 87–89.

5. Мондрий К.П. Графический интерфейс как инструмент взаимодействия человека с компьютером // Науч. электрон. журнал Меридиан. 2020. № 7. С. 69–71. URL: https://elibrary.ru/download/elibrary_42403946_69666421.pdf (дата обращения: 6.06.2023).

6. Ченцов П.А. Об одном подходе к построению интерфейсов консольных приложений: технология TextControlPages // Программная инженерия. 2016. Т. 7. № 12. С. 537–546.

7. Пащенко Д.С. Отражение в российской практике мировых тенденций в технологиях, средствах и подходах в разработке программного обеспечения // Программная инженерия. 2017. Т. 8. № 8. С. 339–344. doi: 10.17587/prin.8.339-344.

8. Banerjee I. Advances in model-based testing of GUI-based software. Advances in Computers, 2017, vol. 105, pp. 45–78. doi: 10.1016/bs.adcom.2016.12.003.

9. Сторожева В.К. Сферы применения языка Python // Актуальные проблемы авиации и космонавтики: сб. матер. VIII Междунар. науч.-практич. конф. 2022. С. 498–500.

10. Пальмов С.В., Посиваткина А.А. Сравнение библиотек tkInter, PyQt, wxPython // ЕНО. 2020. № 11-2. С. 122–126.

11. Шокодько Ф.А. Разработка графического интерфейса клиентского приложения, графических элементов и их динамизация для ПК // Студент года 2023: сб. статей XXV Междунар. науч.-исслед. конкурса. 2023. С. 24–29.

12. Нехорошева Е.М., Конецкая Е.А. Графические возможности Python // Информационно-телекоммуникационные системы и технологии: матер. науч.-практич. конф. 2021. С. 99–101.

Software & Systems doi: 10.15827/0236-235X.142.661-667 2023, vol. 36, no. 4, pp. 661–667

Software module for stressFoam solver in OpenFOAM platform

Dmitry I. Chitalov

For citation
Chitalov, D.I. (2023) ‘Software module for stressFoam solver in OpenFOAM platform’, *Software & Systems*, 36(4), pp. 661–667 (in Russ.). doi: 10.15827/0236-235X.142.661-667

Article info
Received: 15.06.2023 After revision: 19.07.2023 Accepted: 02.08.2023

Abstract. The paper summarizes the results of the work on expanding a graphical shell option list for OpenFOAM platform in terms of using stressFoam solver. The purpose of the solver is studying solids property changes under stress influence. There is a formulated main disadvantage of OpenFOAM environment, which necessitates the study. The author reveals relevance of the problem under study, studies available software solutions and their shortcomings. The goals of the work

and necessary steps to achieve them are determined. The purpose of the stressFoam program and the main scope of its application is given. The paper describes a set of technologies required for implementing a graphical, software part of the project and a data storage subsystem. It also gives a rationale for using each technology and determines a product architecture and a development environment. There are structure and logic diagrams of module operation, highlighted main program components of the module and an algorithm for applying in continuum mechanics problems. The module main window is shown at the stage of completing numerical simulation of one of MCC fundamental problems; using stressFoam solver is studied. Work results are expressed in preparing a software solution that replaces a usual command line approach with a more efficient one, which implies performing all steps through a graphical user interface. The paper determines work scientific novelty components and the expected practical product value for a user. The product source code is placed in a GitHub service repository for providing access to specialists using OpenFOAM in experiments and research.
Keywords: numerical simulation, software module, OpenFOAM, stressFoam solver, open source software, graphical user interface, Python programming language

References

1. Chitalov, D.I., Merkulov, Ye.S., Kalashnikov, S.T. (2016) 'Development of a graphical user interface for the OpenFOAM toolbox', *Software Engineering*, 7 (12), pp. 568–574 (in Russ.). doi: 10.17587/prin.7.568-574.

2. Chitalov, D.I., Kalashnikov, S.T. (2018) 'Development of an application for preparation of computational meshes with graduating and curved edges for the OpenFOAM software', *Systems and Means of Inform.*, 28 (4), pp. 122–135 (in Russ.). doi: 10.14357/08696527180412.

3. Chitalov, D.I., Kalashnikov, S.T. (2018) 'Application development for preparing meshes using snappyHexMesh of OpenFOAM', *Software & Systems*, 31(4), pp. 715–722 (in Russ.). doi: 10.15827/0236-235X.124.715-722.

4. Saliy, T.M., Bazheneyeva, Z.B. (2016) 'Principles of creation of user interface', *Bull. of the InEU*, (3), pp. 87–89 (in Russ.).

5. Mondriy, K.P. (2020) 'Graphical interface as a tool for human-computer interaction', *Meridian*, (7), pp. 69–71, available at: https://elibrary.ru/download/elibrary_42403946_69666421.pdf (accessed June 06, 2023) (in Russ.).

6. Chentsov, P.A. (2016) 'New way to construct console application interfaces: technology TextControlPages', *Software Engineering*, 7 (12), pp. 537–546 (in Russ.).

7. Pashchenko, D.S. (2017) 'Reflection in the Russian practice of world trends in technologies, tools and approaches to software development', *Software Engineering*, 8 (8), pp. 339–344 (in Russ.). doi: 10.17587/prin.8.339-344.

8. Banerjee, I. (2017) 'Advances in model-based testing of GUI-based software', *Advances in Computers*, 105, pp. 45–78. doi: 10.1016/bs.adcom.2016.12.003.

9. Storozheva, V.K. (2022) 'Areas of application of the python language', *Proc. Int. Sci.-Pract. Conf. Actual Problems of Aviation and Cosmonautics*, pp. 498–500 (in Russ.).

10. Palmov, S.V., Poskivatkina, A.A. (2020) 'Comparison of tkInter, PyQt, wxPython Libraries', *ESA*, (11-2), pp. 122–126 (in Russ.).

11. Shokodko, F.A. (2023) 'Development of the graphic interface of the client application, graphic elements and their dynamization for PC', *Proc. XXV Int. Research Compet. Student of the Year 2023*, pp. 24–29 (in Russ.).

12. Nekhorosheva, E.M., Konetskaya, E.A. (2021) 'Python graphical capabilities', *Proc. Sci.-Pract. Conf. Inform. and Telecommunication Systems and Tech.*, pp. 99–101 (in Russ.).

Авторы
Читалов Дмитрий Иванович [1],
младший научный сотрудник, cdi9@yandex.ru

[1] Федеральное государственное бюджетное научное учреждение «Южно-Уральский научный центр»,
г. Миасс, 456317, Россия

Authors
Dmitry I. Chitalov [1], Junior Researcher,
cdi9@yandex.ru

[1] Federal State Budget Scientific Institution
"South Ural Scientific Center",
Miass, 456317, Russian Federation

УДК 004.94:621.372.852 doi: 10.15827/0236-235X.142.668-677 2023. Т. 36. № 4. С. 668–677

Проектирование фильтра из шпилечных резонаторов с металлизированными отверстиями с помощью программной системы

А.В. Лабынцев
В.В. Игнатьев
В.В. Васильев
Вит.А. Михайловский
Д.В. Харланов

Ссылка для цитирования
Лабынцев А.В., Игнатьев В.В., Васильев В.В., Михайловский Вит.А., Харланов Д.В. Проектирование фильтра из шпилечных резонаторов с металлизированными отверстиями с помощью программной системы // Программные продукты и системы. 2023. Т. 36. № 4. С. 668–677. doi: 10.15827/0236-235X.142.668-677
Информация о статье
Поступила в редакцию: 19.09.2023 После доработки: 10.10.2023 Принята к публикации: 23.10.2023

Аннотация. В статье описывается программная система, представляющая собой прикладное программное обеспечение для проектирования микрополосковых полоснопропускающих фильтров. В ее основе лежит программное обеспечение Ansoft HFSS, предназначенное для трехмерного электромагнитного моделирования СВЧ-структур. Система состоит из двух основных взаимосвязанных компонентов – программы для решения задачи анализа фильтра и программы для решения задачи синтеза многополюсника, заменяющего фильтр в процессе его проектирования. В работе приведено описание структуры программной системы и взаимосвязи ее отдельных компонентов. Использование программной системы проиллюстрировано на примере проектирования фильтра, состоящего из трех противонаправленных шпилечных резонаторов с металлизированными отверстиями в середине сгиба шпильки. Введение в середину резонатора короткозамкнутого отверстия позволяет почти в два раза расширить верхнюю полосу заграждения фильтра за счет перехода от простого полуволнового резонатора к совокупности двух четвертьволновых резонаторов. Для проектирования фильтра с отверстиями использована программная система, основанная на переходе от фильтра к многополюснику, состоящему из связанных друг с другом резонаторов. В результате спроектирован фильтр, обладающий требуемыми электрическими характеристиками. Описанная в статье программная система для проектирования данного фильтра подтвердила свою состоятельность и эффективность. Практическая значимость работы заключается в более чем двукратном расширении верхней полосы заграждения фильтра с металлизированными отверстиями по сравнению с аналогичным фильтром из простых шпилечных резонаторов без отверстий.

Ключевые слова: проектирование, микрополосковый фильтр, шпилечный резонатор, металлизированное отверстие, программная система, алгоритм синтеза, характеристика затухания, полоса пропускания, полоса заграждения
Благодарности. Работа поддержана Южным федеральным университетом, проект № ВнГр/23-01-РТ

Введение. Проектирование *микрополосковых фильтров* (МПФ) невозможно без применения современных программных систем электродинамического моделирования СВЧ-устройств. Поэтому задача проектирования любого СВЧ-устройства лежит в области создания методик, алгоритмов и программных продуктов, которые являются фундаментом для построения конкретной программной системы.

Стремление обеспечить наилучшие электрические и массогабаритные характеристики МПФ нацеливает разработчиков на поиск новых эффективных топологических решений и методов проектирования микрополосковых устройств. Широко распространенные фильтры из полуволновых шпилечных резонаторов обладают слишком узкой верхней полосой заграждения, расширить которую почти в два раза позволяет введение металлизированного отверстия в точке сгиба шпильки. Такое преобразование приводит к необходимости поиска новых подходов, пригодных для проектирования фильтра с отверстиями в резонаторах при любом заданном числе резонирующих элементов.

В данной работе эта задача решается путем создания программной системы, состоящей из двух основных взаимосвязанных компонентов:

– программы «Анализ» для решения задачи анализа фильтра;

– программы «Синтез» для решения задачи синтеза многополюсника, заменяющего фильтр в процессе его проектирования.

Переход от фильтра к многополюснику дает возможность представить фильтр как совокупность взаимосвязанных резонаторов, каждый из которых характеризуется собственной проводимостью и проводимостями связи с соседними резонаторами. Методика синтеза дает возможность сформулировать математические требования к указанным проводимостям в пределах *основной полосы пропускания* (ОПП).

Удовлетворение этих требований путем целенаправленного подбора геометрических размеров позволяет спроектировать многополюсник и в результате получить фильтр с требуемой шириной ОПП и с заданным положением этой полосы на частотной оси.

В данной работе приводятся основные положения методики синтеза, описаны компоненты разработанной программной системы и приведены результаты проектирования трехрезонаторного *полоснопропускного фильтра* (ППФ) из шпилечных резонаторов с металлизированными отверстиями в середине сгиба шпильки. Этот результат свидетельствует о научной новизне решаемой задачи и ее практической значимости.

Анализ современного состояния исследований по проблеме проектирования компактных МПФ

Разработке и исследованию МПФ посвящено немало научных работ. В [1–3] приведены результаты проектирования компактных МПФ на резонаторах типа шпилька и скрепка, которые обладают малым количеством резонаторов и довольно сложной топологией. Показано, как за счет существенного усложнения топологии фильтра удается сформировать по несколько полюсов затухания справа и слева от ОПП и управлять положением этих полюсов. Такой подход позволяет добиться высокой прямоугольности характеристики затухания в окрестности ОПП [1], однако провалы между полюсами затухания могут достигать 25–35 дБ, что говорит о недостаточно высокой избирательности фильтра в широком диапазоне частот. Вследствие использования полуволновых резонаторов первая паразитная полоса пропускания формируется в окрестности удвоенной средней частоты ОПП.

Описания микрополосковых шпилечных ППФ, которые позволяют сформировать два полюса затухания в непосредственной близости от ОПП и тем самым повысить прямоугольность АЧХ-фильтра, даны в источниках [4, 5]. Однако данное решение имеет недостаток – узкая полоса заграждения, обусловленная присутствием паразитной полосы пропускания в окрестности удвоенной средней частоты ОПП.

Металлизированные отверстия традиционно применяются для заземления одного из концов резонаторов, используемых в фильтрах на четвертьволновых резонаторах – фильтрах гребенчатого или встречностержневого типа. Однако о специальном применении таких отверстий для преобразования полуволнового резонатора в совокупность двух четвертьволновых в литературе упоминается довольно редко. Можно привести лишь несколько сравнительно устаревших, но единственно актуальных для данной работы статей, например [6–8], в которых предлагаются решения, использующие металлизированные отверстия в центре полуволнового резонатора. В этих работах показано, что введение металлизированных отверстий в резонатор в той или иной степени позволяет улучшить электрические характеристики фильтра, но методика проектирования фильтров в явном виде в них не описана. В данной работе предлагается алгоритм проектирования фильтра из шпилечных резонаторов с отверстиями.

Методика решения задачи проектирования фильтра

С точки зрения методики проектирования фильтров данная статья является логическим продолжением работ [9–11], в которых представлены результаты синтеза в HFSS фильтров сравнительно сложной структуры и различной топологии.

Процесс проектирования устройства в HFSS базируется на решении двух главных задач – синтеза и анализа устройства. Программа HFSS позволяет решить только задачу анализа устройства, а именно, найти электрические характеристики устройства для определенной конструкции со всеми заданными геометрическими размерами его конструктивных элементов.

Напротив, в ходе проектирования фильтра необходимо решить обратную задачу – задачу синтеза устройства, а именно, осуществить поиск геометрических размеров всех элементов конструкции по известным электрическим характеристикам объекта. Правильность полученного конструктивного решения подтверждается выполнением задачи анализа спроектированного макетного образца, который должен обладать электрическими характеристиками, удовлетворяющими техническому заданию.

Задача проектирования МПФ является многопараметрической и не имеет аналитического решения. Прямой подход к ее решению основан на расчете и оптимизации характеристики фильтра $|S_{21}(f)|$ путем подбора геометрических размеров. Данную характеристику можно рассчитать лишь численными методами при всех

заданных размерах элементов фильтра в выбранной разработчиком программной среде электродинамического моделирования, например в HFSS [12]. Поскольку значения размеров в начале синтеза неизвестны, берутся приблизительные. Далее выполняется анализ фильтра с этими размерами, в результате которого становится ясно, что полученная характеристика $|S_{21}(f)|$ далеко не соответствует требованиям технического задания.

Изменение любого геометрического размера и повторный расчет характеристики, который, кстати, может занять от одного до нескольких часов машинного времени, приводят к такому же отрицательному результату. Поэтому прямой путь решения задачи синтеза, основанный на многократных расчетах и параметрической оптимизации $|S_{21}(f)|$ в процессе подбора геометрических размеров, как правило, не может привести к успешному результату проектирования.

В работах [9–11] предлагается иной подход к проектированию МПФ, основанный на переходе от фильтра к так называемому 2*n*-полюснику, где *n* – число резонансных элементов фильтра. Для реализации такого перехода были разработаны два программных продукта, которые получили названия «Анализ» и «Синтез» и являются основными компонентами программной системы для проектирования фильтра. В данной работе программная система применена для проектирования фильтра из трех противонаправленных шпилек с металлизированными отверстиями в середине сгиба шпильки. Топология фильтра с обозначениями задаваемых и искомых геометрических размеров представлена на рисунке 1.

Для упрощения задачи предполагается, что фильтр обладает осевой геометрической симметрией и сопротивление нагрузок на его входе и выходе одинаковое. К искомым размерам конструкции относятся длины l_1, l_2 и l_3 резонансных элементов, зазоры s_1, s_2 и s_3 между резонансными элементами, точки подключения *lt* подводящих линий, соединяющих вход и выход фильтра с крайними резонансными элементами. Ширина подводящей линии *wl* выбирается из условия равенства ее волнового сопротивления сопротивлениям нагрузок. Ширина резонансных элементов w_1, w_2 и w_3 и ширина перемычки шпильки *w* задаются произвольно, но в ходе синтеза могут изменяться для более точного решения задачи.

Схематическое изображение фильтра и соответствующего ему 2*n*-полюсника приведено

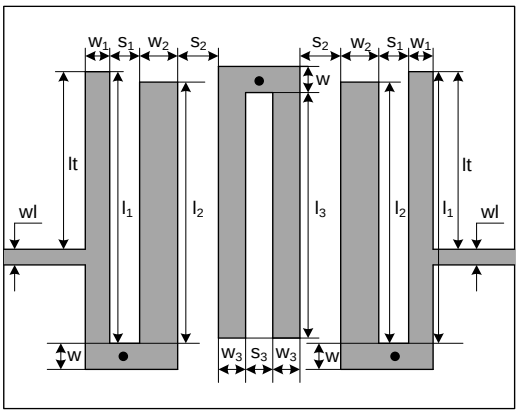


Рис. 1. Топология фильтра с обозначениями геометрических размеров

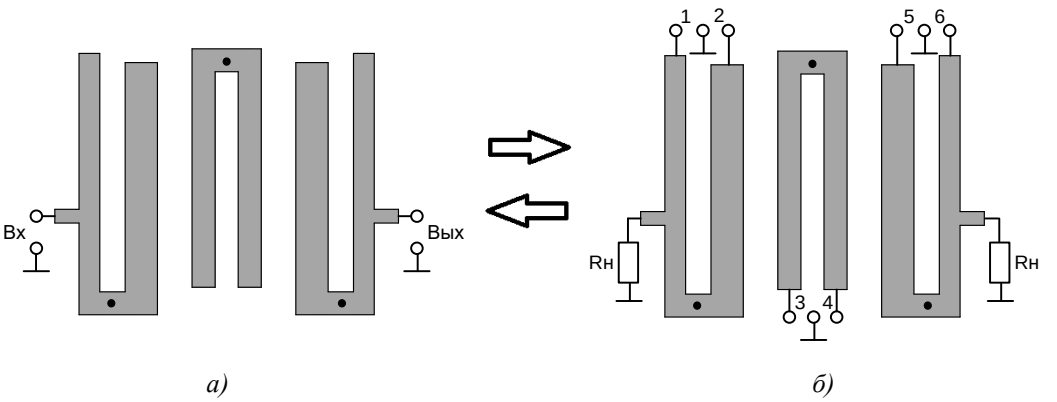
Fig. 1. Filter topology with geometric dimensions

на рисунке 2. В данном случае 2*n*-полюсник содержит $n = 6$ резонансных элементов, поэтому является 12-полюсником.

Как видно на рисунке 2, для перехода от фильтра к 12-полюснику необходимо выполнить всего два действия. Во-первых, в точках входа и выхода устройства надо разместить сосредоточенные резисторы с сопротивлением, равным входному сопротивлению на внешних зажимах фильтра. Во-вторых, в середине разомкнутых концов каждого резонансного элемента надо создать новый виртуальный полюс, который будет являться зажимом для определения новых электрических характеристик резонансных элементов фильтра. Такими характеристиками являются собственные проводимости резонансных элементов $\dot{Y}_{ii}(f)$ и проводимости связи $\dot{Y}_{ij}(f)$ между резонансными элементами.

Таким образом, в результате перехода от фильтра к 2*n*-полюснику фактически осуществляется переход от одной характеристики фильтра $|S_{21}(f)|$ к совокупности большого числа характеристик 2*n*-полюсника. Ими являются проводимости $\dot{Y}_{ij}(f)$, каждая из которых представляет собой комплексную функцию частоты, характеризующую свойства отдельного резонансного элемента во взаимосвязи с другими элементами 2*n*-полюсника. Матрица проводимостей 12-полюсника на рисунке 2б имеет вид

$$\left[\dot{Y}\right]_6 = \begin{bmatrix} \dot{Y}_{11} & \cdots & \dot{Y}_{16} \\ \vdots & \ddots & \vdots \\ \dot{Y}_{61} & \cdots & \dot{Y}_{66} \end{bmatrix}.$$



а) *б)*

Рис. 2. Структурные схемы: а) фильтр; б) 12-полюсник

Fig. 2. Block diagrams: a) filter; б) 12-pole

Требования к проводимостям 12-полюсника формулируются в пределах ОПП и представляют собой совокупность условий резонанса и условий согласования. Исследования показали, что эти условия для 2*n*-полюсника на рисунке 2б выглядят следующим образом:

– условия резонанса

$$\frac{\mathrm{Im}\,\dot{Y}_{i,i}(f)\cdot\rho_i}{\mathrm{Im}\,\dot{Y}_{i,i+1}(f)\cdot\gamma}\le 1,\ f\in[f_н, f_в],\ i=1,2,3; \qquad (1)$$

– условия согласования

$$\frac{\rho_1}{\left|\mathrm{Im}\,\dot{Y}_{1,2}(f)\right|}=\frac{\rho_2}{\left|\mathrm{Im}\,\dot{Y}_{2,3}(f)\right|}= \\ =\frac{1}{\mathrm{Re}\,\dot{Y}_{1,1}(f)+\mathrm{Re}\,\dot{Y}_{2,2}(f)},\ f\in[f_н, f_в]. \qquad (2)$$

В соотношениях (1) и (2) использованы обозначения: $\mathrm{Im}\,\dot{Y}_{i,i}(f)$ – мнимая составляющая собственной проводимости *i*-го резонансного элемента; $\mathrm{Im}\,\dot{Y}_{i,i+1}(f)$ – мнимая составляющая проводимости связи между *i*-м и *i*+1-м элементами; $\mathrm{Re}\,\dot{Y}_{i,i}(f)$ – реальная составляющая собственной проводимости *i*-го резонансного элемента; $f_н$ – нижняя граничная частота ОПП; $f_в$ – верхняя граничная частота ОПП; ρ_i, γ – вещественные константы, определяемые на основе параметров лестничного прототипа g_i по формулам:

$$\gamma = g_1,\ \rho_j=\frac{g_1}{\sqrt{g_j\cdot g_{j+1}}},\ j=1,2,...,n-1. \qquad (3)$$

Важно, что собственные проводимости отдельных резонансных элементов $\mathrm{Im}\,\dot{Y}_{i,i}(f)$, фигурирующих в (1), зависят прежде всего от геометрических параметров именно того элемента, на зажиме которого они определяются. Другие элементы тоже оказывают определенное влияние на эту характеристику, но оно существенно меньше. Это позволяет удовлетворить требования (1) к проводимостям 2*n*-полюсника в пределах ОПП в ходе итерационного подбора значений геометрических размеров при сравнительно небольшом числе итераций.

Выполнить условие согласования (2) можно лишь в одной или двух точках в пределах полосы пропускания, поскольку практически во всех случаях функции $\frac{\rho_1}{\left|\mathrm{Im}\,\dot{Y}_{1,2}(f)\right|}$, $\frac{\rho_2}{\left|\mathrm{Im}\,\dot{Y}_{2,3}(f)\right|}$ и $\frac{1}{\mathrm{Re}\,\dot{Y}_{1,1}(f)+\mathrm{Re}\,\dot{Y}_{2,2}(f)}$ ведут себя неодинаково. Основное влияние на значение последней функции оказывают геометрические параметры крайней шпильки, к которой подключаются внешние цепи.

Управлять значением первых двух функций можно величиной зазоров s_1 и s_2, но надо учитывать, что значения $\mathrm{Im}\,\dot{Y}_{1,2}(f)$ и $\mathrm{Im}\,\dot{Y}2_{2,3}(f)$ фигурируют и в условиях резонанса (1), влияя на наклон резонансной характеристики, а следовательно, на ширину полосы пропускания. Поэтому основным способом обеспечения согласования фильтра с внешними цепями является верный выбор значения $\frac{1}{\mathrm{Re}\,\dot{Y}_{1,1}(f)+\mathrm{Re}\,\dot{Y}_{2,2}(f)}$.

Важно отметить, что условие (1) так или иначе выполняется при любых геометрических размерах элементов фильтра. То есть резонансная характеристика $\frac{\mathrm{Im}\,\dot{Y}_{i,i}(f)\cdot\rho_i}{\mathrm{Im}\,\dot{Y}_{i,i+1}(f)\cdot\gamma}$, определенная для каждого из резонансных элементов, всегда будет пересекать частотную ось на своей резонансной частоте и в окрестности этого нуля резонансной характеристики будет формировать-

ся *потенциальная полоса пропускания* (ППп) данного резонансного элемента, ширина которой будет удовлетворять соответствующее неравенство (1).

В ходе проектирования необходимо обеспечить максимальное совпадение всех ППп по всем резонансным элементам, причем границы этих полос должны совпадать с границами *заданной полосы пропускания* (ЗПП) или быть близкими к ним. Если выполнить и условия согласования (2) в пределах ППп, то полученная ППп перейдет в *фактическую полосу пропускания* (ФПП) фильтра, определяемую как диапазон частот, в пределах которого функция $20\lg|S_{21}(f)|$ принимает значения менее 3 дБ. Этот факт обязательно требует подтверждения в итоге проектирования, которое осуществляется путем перехода от 2*n*-полюсника к фильтру и расчетом его характеристик.

Компоненты программной системы и алгоритм проектирования

Описанный алгоритм проектирования был реализован в виде программной системы, структурная схема которой представлена на рисунке 3. Далее описываются структура этой системы и алгоритм ее использования для достижения поставленной цели.

В программной среде HFSS [12] разрабатывается программа «Синтез», содержащая макет многополюсника. В этой программе задаются первоначальные значения геометрических параметров устройства и определяется диапазон частот, в котором будет производиться синтез многополюсника. Проектирование фильтра начинается с расчета характеристик многополюсника, фигурирующих в выражениях (1) и (2), в пределах ЗПП и ее окрестности.

По результатам анализа определяются положение ППп по факту удовлетворения соотношения (1), а также степень выполнения условий согласования (2) в этой ППп. Выявленные различия между параметрами ППп и ЗПП позволяют скорректировать размеры элементов многополюсника, причем важно, что корректировка производится с помощью аргументированных численных расчетов, а не «слепым» перебором. Это, в свою очередь, существенно уменьшает число итераций по подбору требуемых геометрических размеров и сокращает общее время проектирования.

Расчет характеристик многополюсника и корректировка его размеров повторяются до тех пор, пока разработчик не будет удовлетворен степенью выполнения требований (1) и (2) и соответствия ППп и ЗПП. На этом первый этап синтеза устройства заканчивается.

На следующем этапе из макета многополюсника в программе «Синтез» создается макет фильтра в программе «Анализ» при неизменных значениях всех размеров элементов устройства. Выполняется расчет характеристик фильтра, основной из которых является элемент матрицы рассеяния $S_{21}(f)$, представленный в логарифмическом виде $20\lg|S_{21}(f)|$. В результате расчета определяются границы ФПП по уровню -3 дБ и устанавливается степень соответствия между ФПП, ППп и ЗПП. В большинстве случаев это соответствие удовлетворяет разработчика, а некоторое расхождение между ФПП и ЗПП при необходимости можно ликвидировать незначительной корректировкой длин резонаторов в самом фильтре.

Анализ полученных результатов

Перейдем к синтезу фильтра, топология которого представлена на рисунке 1. Прежде всего для заданного уровня пульсаций 3 дБ максимально плоской аппроксимации и числа резонансных элементов $n = 6$ определяются параметры g_i и выполняется расчет параметров цепочечного прототипа ρ_i, γ по формулам (3): $g_1 = g_6 = 0{,}5176$, $g_2 = g_5 = 1{,}414$, $g_3 = g_4 = 1{,}932$, $\gamma = 0{,}5176$, $\rho_1 = \rho_5 = 0{,}605$, $\rho_2 = \rho_4 = 0{,}313$, $\rho_3 = 0{,}268$.

Для определенности на первом этапе проектирования будем считать, что нижняя и верхняя границы ЗПП по техническому заданию равны: $f_н = 1{,}05$ ГГц и $f_в = 1{,}35$ ГГц.

В среде HFSS в программе «Синтез» создается макет 12-полюсника на диэлектрической подложке Arlon AD 1000 толщиной 1 мм с относительной диэлектрической проницаемостью 10,2. После нескольких итераций по подбору геометрических размеров 12-полюсника путем удовлетворения соотношений (1) и (2) согласно структурной схеме и алгоритму, представленным на рисунке 3, получены результаты, приведенные ниже.

На рисунке 4 изображена модель спроектированного 12-полюсника в HFSS и приведена таблица его геометрических размеров в соответствии с обозначениями, представленными на рисунке 1.

На рисунке 5 представлены итоговые характеристики 12-полюсника, подтверждающие выполнение условий резонанса (1) и условий согласования (2). В отличие от первого и тре-

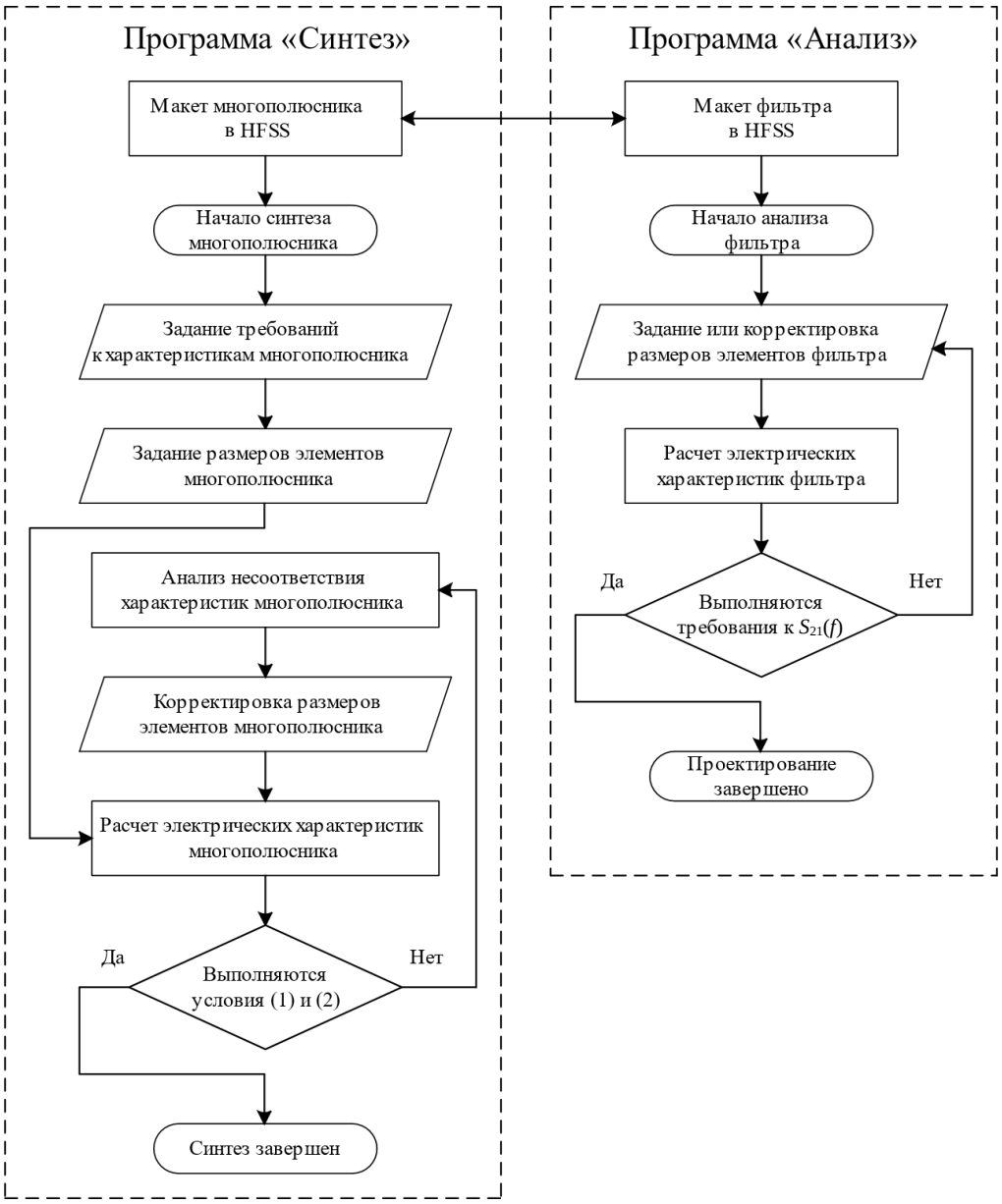


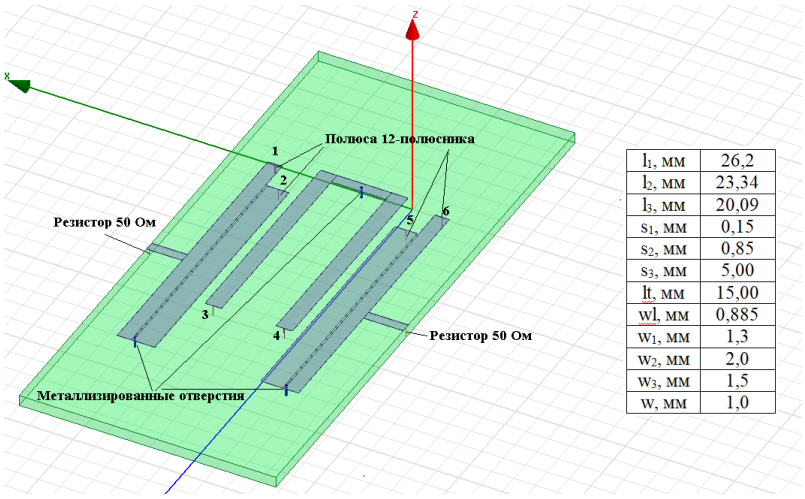
Рис. 3. Структурная схема программной системы проектирования фильтра

Fig. 3. Block diagram of a filter design software system

тьего резонансных элементов, для которых условия резонанса (1) выполняются с достаточной точностью и границы ППп соответствуют границам ЗПП, резонансная характеристика второго резонансного элемента имеет слишком большую крутизну, хотя и проходит через ноль в середине ЗПП. Данное обстоятельство сужает ППп по второму резонансному элементу, объясняется это тем, что выбор зазора s_2 осуществлялся, исходя из обеспечения условия согласования (2), которое выполняется в середине ЗПП в окрестности частоты 1,2 ГГц.

По характеристикам фильтра $20\lg|S_{21}(f)|$ и $20\lg|S_{11}(f)|$, полученным после перехода к фильтру от 12-полюсника, представленного на рисунке 2б, построен график (http://www.swsys.ru/uploaded/image/2023-4/14.jpg).

Анализ этих характеристик показывает их хорошее соответствие характеристикам 12-полюсника, что подтверждает применимость и эффективность использованной для проектирования методики синтеза. ППп всех резонансных элементов не удалось выровнять в связи с тем, что ППп первого и третьего резонансных



l_1, мм	26,2
l_2, мм	23,34
l_3, мм	20,09
s_1, мм	0,15
s_2, мм	0,85
s_3, мм	5,00
lt, мм	15,00
wl, мм	0,885
w_1, мм	1,3
w_2, мм	2,0
w_3, мм	1,5
w, мм	1,0

Рис. 4. Модель спроектированного 12-полюсника в HFSS и таблица геометрических размеров

Fig. 4. 12-pole in HFSS model and a table of geometric dimensions

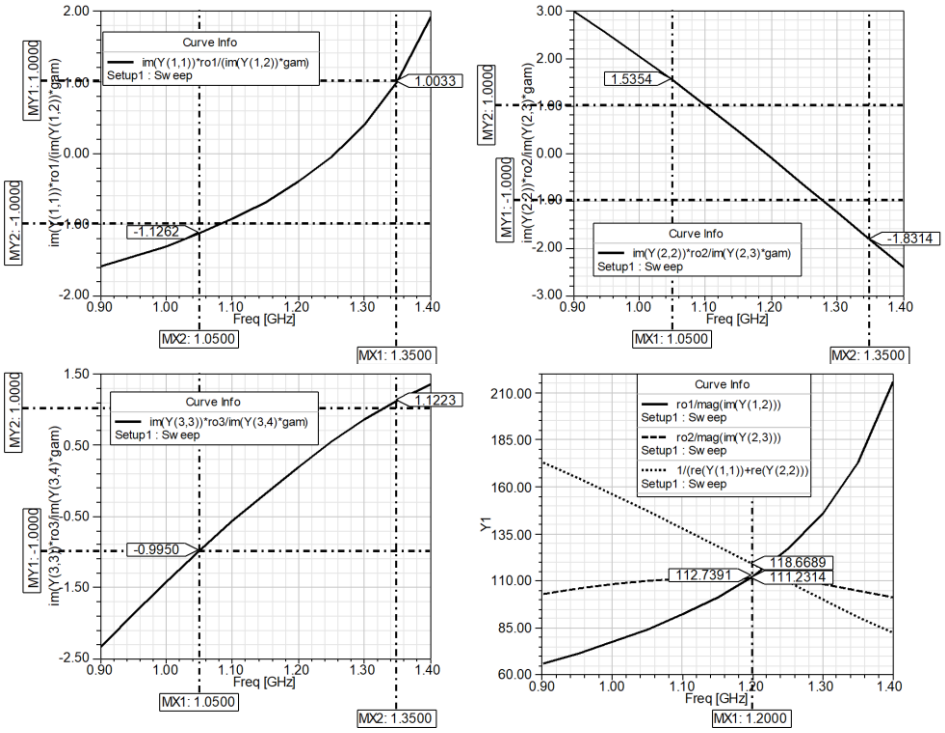


Рис. 5. Итоговые характеристики 12-полюсника

Fig. 5. 12-pole final characteristics

элементов соответствуют границам ЗПП, а ППп второго резонансного элемента расположилась в диапазоне от 1,1 до 1,28 ГГц. ФПП фильтра занимает диапазон от 1,075 ГГц до 1,255 ГГц, ширина ФПП 180 МГц, что соответствует ширине ППп второго резонансного элемента, но уже, чем ЗПП.

Характеристики спроектированного фильтра в широком диапазоне частот (http://www.swsys.ru/uploaded/image/2023-4/15.jpg) показывают, что полоса заграждения по уровню -30 дБ расположена от 1,35 до 2,65 ГГц и составляет 1,3 ГГц.

Для подтверждения эффекта от введения в шпильки металлизированных отверстий на рисунке 6 приведены результаты расчета характеристик фильтра без металлизированных отверстий.

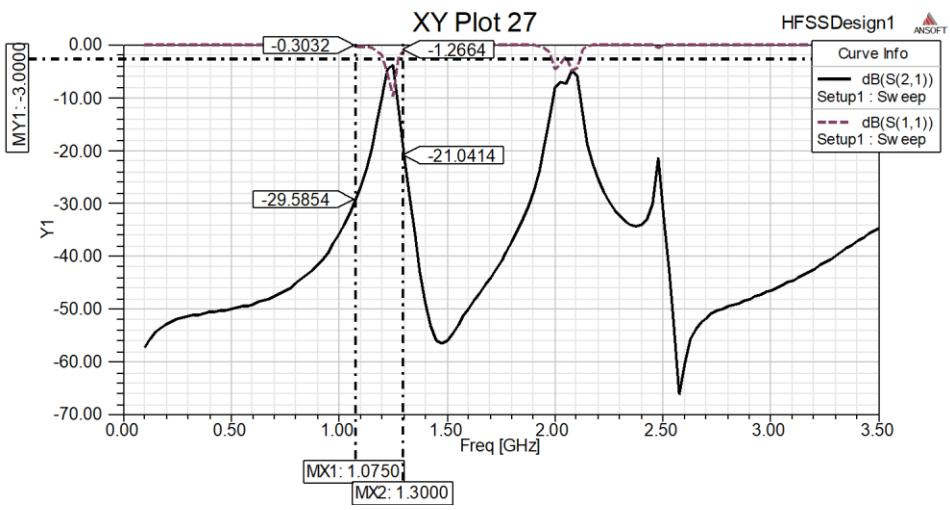


Рис. 6. Характеристики фильтра без металлизированных отверстий в широком диапазоне

Fig. 6. Filter characteristics without metallized holes in a wide range

Данный фильтр был получен из фильтра с итоговыми характеристиками в широком диапазоне простым удалением металлизированных отверстий. Фильтр без отверстий в шпильках не подвергался синтезу, поэтому его характеристики в ОПП не являются оптимальными. Это и не требовалось, поскольку основная задача исследования на данном этапе заключалась только в оценке ширины полосы заграждения фильтра. Первая паразитная полоса фильтра без отверстий расположена в окрестности удвоенной средней частоты ОПП. По уровню -30 дБ полоса заграждения фильтра без отверстий занимает диапазон от 1,35 до 1,9 ГГц, то есть ее ширина составляет 0,55 ГГц. Отношение ширины полосы заграждения фильтра с отверстиями к полосе фильтра без отверстий составляет 1,3/0,55 = 2,36. Таким образом, введение отверстий в шпилечные резонаторы позволило более чем в два раза расширить полосу заграждения устройства.

Заключение

Результаты данной работы иллюстрируют использование разработанной программной системы для проектирования фильтров из шпилечных резонаторов с металлизированными отверстиями на сгибе шпилек. Введение отверстий в резонаторы позволило расширить верхнюю полосу заграждения фильтра более чем в два раза. Полученные результаты свидетельствуют о том, что использование представленной методики синтеза и реализующей ее программной системы для проектирования аналогичных фильтров, состоящих из большего числа резонаторов, приведет к повышению затухания в полосе заграждения фильтра при сохранении прочих положительных результатов проектирования.

Список литературы

1. Аристархов Г.М., Кириллов И.Н. Компактные микрополосковые фильтры с повышенной частотной избирательностью на основе полуволновых резонаторов // СИНХРОНИНФО. 2020. Т. 11. № 2. С. 40–44. doi: 10.1109/SYNCHROINFO49631.2020.9166122.

2. Аристархов Г.М., Кириллов И.Н., Корчагин А.И., Кувшинов В.В. Компактные высокоизбирательные микрополосковые фильтры на свернутых сонаправленных шпилечных резонаторах // Радиотехника. 2021. Т. 85. № 4. С. 126–137. doi: 10.18127/j00338486-202104-14.

3. Aristarkhov G.M., Arinin O.V., Kirillov I.N., Markovskiy A.V., Doronina A.D. Multi-band bandpass microstrip filters based on two codirectional hairpin resonators. Proc. Systems of Signals Generating and Processing in the Field of on Board Communications, 2023, pp. 1–5. doi: 10.1109/IEEECONF56737.2023.10091997.

4. Беляев Б.А., Сержантов А.М., Бальва Я.Ф., Лексиков Ан.А., Грушевский Е.О. Высокоселективный полосно-пропускающий фильтр на резонаторах с двухсторонним рисунком полосковых проводников на подвешенной подложке // Письма в журнал технич. физики. 2019. Т. 45. № 10. С. 13–16. doi: 10.21883/PJTF.2019.10.47749.17703.

5. Leksikov A.A., Serzhantov A.M., Govorun I.V., Afonin A.O., Ugryumov A.V., Leksikov An.A. A method of stop-band widening in BPF based on two-conductor suspended-substrate resonators. Progress in Electromagnetics Research Letters, 2018, vol. 72, pp. 11–16. doi: 10.2528/pierl17102302.

6. Xiao F., Norgren M., He S. Compact third-order microstrip bandpass filter using hybrid resonators. Progress in Electromagnetics Research C, 2011, vol. 19, pp. 93–106. doi: 10.2528/PIERC10092706.

7. Girbau D., Lázaro A., Pérez A., Pradell L. Dual-band bandpass filter based on a hole resonator. Microwave and Optical Technology Letters, 2009, vol. 51, no. 7, pp. 1649–1652. doi: 10.1002/mop.24410.

8. Lin W.-J., Li J.-Y., Houng M.-P. Miniaturized wideband ring-type bandpass filters with upper stopband characteristic. J. of Electromagnetic Waves and Applications, 2010, vol. 24, no. 7, pp. 931–939. doi: 10.1163/156939310791285191.

9. Obukhovets V.A., Labyntsev A.V., Kharlanov D.V., Ponomarev N.E., Sarzhanov A.N. Example of filter synthesis on two anti-directional stud resonators with one metalized hole in each resonator. Proc. RSEMW, 2023, pp. 160–163. doi: 10.1109/RSEMW58451.2023.10202019.

10. Labyntsev A.V., Ponomarev N.E., Kharlanov D.V., Sarzhanov A.N., Maksimov M.N. Design of a small-sized filter on quarter-wave hairpin resonators. Proc. RSEMW, 2023, pp. 164–167. doi: 10.1109/RSEMW58451.2023.10202118.

11. Лабынцев А.В., Харланов Д.В., Пономарёв Н.Е., Саржанов А.Н. Полосовой микрополосковый фильтр из двух трехсекционных резонаторов // Антенны. 2022. № 6. С. 69–78. doi: 10.18127/j03209601-202206-06.

12. Банков С.Е., Курушин А.А. Расчет антенн и СВЧ структур с помощью HFSS Ansoft. М.: ЗАО «НПП „Родник“», 2009. 256 с.

Software & Systems doi: 10.15827/0236-235X.142.668-677 2023, vol. 36, no. 4, pp. 668–677

Designing a filter from metallized-hole hairpin resonators using a software system

Alexey V. Labyntsev
Vladimir V. Ignatyev
Vasily V. Vasilyev
Vitaly A. Mikhaylovsky
Dmitry V. Kharlanov

For citation
Labyntsev, A.V., Ignatyev, V.V., Vasilyev, V.V., Mikhaylovsky, Vit.A., Kharlanov, D.V. (2023) ‘Designing a filter from metallized-hole hairpin resonators using a software system’, *Software & Systems*, 36(4), pp. 668–677 (in Russ.). doi: 10.15827/0236-235X.142.668-677
Article info
Received: 19.09.2023 After revision: 10.10.2023 Accepted: 23.10.2023

Abstract. The paper describes a software system that is an application software for designing microstrip bandpass filters. The developed system is based on Ansoft HFSS software designed for three-dimensional electromagnetic modeling of microwave structures. The system consists of two main interrelated components, namely: a program for solving a filter analysis problem and a program for solving a multi-pole synthesis problem when a multi-pole replaces a filter during its design. The paper describes the software system structure and the relationship of its individual components; presents an algorithm of using the system. The software system use is illustrated by the example of designing a filter that consists of three counter-directional hairpin resonators with metallized holes in the middle of a hairpin bend. A short-circuited hole in the middle of a resonator expands a filter barrier upper band almost twice due to a transition from a simple half-wave resonator to a combination of two quarter-wave resonators. A filter with holes is designed using a software system based on a transition from a filter to a multi-pole consisting of resonators connected to each other. The result is a designed filter that has the required electrical characteristics. The software system described in the paper for designing this filter has confirmed its viability and effectiveness. The practical significance of the work is in the more than twofold expansion of a filter barrier upper band with metallized holes compared with a similar filter made of simple hairpin resonators without holes.

Keywords: design, microstrip filter, hairpin resonator, metallized hole, software system, synthesis algorithm, attenuation characteristic, bandwidth, barrier band
Acknowledgements. The paper was supported by the Southern Federal University, project no. VnGr/23-01-RT “Research of software-defined radio systems”

References

1 Aristarkhov, G.M., Kirillov, I.N. (2020) 'Compact microwave half-wave resonator filters with increased frequency selectivity', *SYNCHROINFO*, 11(2), pp. 40–44 (in Russ.). doi: 10.1109/SYNCHROINFO49631.2020.9166122.

2 Aristarkhov, G.M., Kirillov, I.N., Korchagin, A.I., Kuvshinov, V.V. (2021) 'Compact highly selective microstrip filters on folded co-directional hairpin resonators', *Radioengineering*, 85(4), pp. 126–137 (in Russ.). doi: 10.18127/j00338486-202104-14.

3 Aristarkhov, G.M., Arinin, O.V., Kirillov, I.N., Markovskiy, A.V., Doronina, A.D. (2023) 'Multi-band bandpass microstrip filters based on two codirectional stud resonators', *Proc. Systems of Signals Generating and Processing in the Field of on Board Communications*, pp. 1–5. doi: 10.1109/IEEECONF56737.2023.10091997.

4 Belyaev, B.A., Serzhantov, A.M., Bal'va, Ya.F., Leksikov, An.A., Grushevskiy, E.O. (2019) 'Highly selective band-pass filter on resonators with a two-sided pattern of strip conductors on a suspended substrate', *Tech. Phys. Letters*, 45(10), pp. 13–16 (in Russ.). doi: 10.1134/S1063785019050225.

5 Leksikov, A.A., Serzhantov, A.M., Govorun, I.V., Afonin, A.O., Ugryumov, A.V., Leksikov, An.A. (2018) 'A method of stopband widening in BPF based on two-conductor suspended-substrate resonators', *Progress in Electromagnetics Research Letters*, 72, pp. 11–16. doi: 10.2528/pierl17102302.

6 Xiao, F., Norgren, M., He, S. (2011) 'Compact third-order microstrip bandpass filter using hybrid resonators', *Progress in Electromagnetics Research C*, 19, pp. 93–106. doi: 10.2528/PIERC10092706.

7 Girbau, D., Lázaro, A., Pérez, A., Pradell, L. (2009) 'Dual-band bandpass filter based on a hole resonator', *Microwave and Optical Technology Letters*, 51(7), pp. 1649–1652. doi: 10.1002/mop.24410.

8 Lin, W.-J., Li, J.-Y., Houng, M.-P. (2010) 'Miniaturized wideband ring-type bandpass filters with upper stopband characteristic' *J. of Electromagnetic Waves and Applications*, 24(7), pp. 931–939. doi: 10.1163/156939310791285191.

9 Obukhovets, V.A., Labyntsev, A.V., Kharlanov, D.V., Ponomarev, N.E., Sarzhanov, A.N. (2023) 'Example of filter synthesis on two anti-directional stud resonators with one metalized hole in each resonator', *Proc. RSEMW*, pp. 160–163. doi: 10.1109/RSEMW58451.2023.10202019.

10 Labyntsev, A.V., Ponomarev, N.E., Kharlanov, D.V., Sarzhanov, A.N., Maksimov, M.N. (2023) 'Design of a small-sized filter on quarter-wave hairpin resonators', *Proc. RSEMW*, pp. 164–167. doi: 10.1109/RSEMW58451.2023.10202118.

11 Labyntsev, A.V., Kharlanov, D.V., Ponomarev, N.E., Sarzhanov, A.N. (2022) 'Bandpass microstrip filter from two three-section resonators', *Antennas*, (6), pp. 69–78 (in Russ.). doi: 10.18127/j03209601-202206-06.

12 Bankov, S.E., Kurushin, A.A. (2009) *Calculation of Antennas and Microwave Structures Using HFSS Ansoft*. Moscow, 256 p. (in Russ.).

Авторы

Лабынцев Алексей Викторович [1], к.т.н., доцент, старший научный сотрудник, labyntsev@yandex.ru
Игнатьев Владимир Владимирович [1], к.т.н., ведущий научный сотрудник, vvignatev@sfedu.ru
Васильев Василий Васильевич [1], младший научный сотрудник, vasily@sunsdr.com
Михайловский Виталий Алексеевич [1], младший научный сотрудник, vitaly@sunsdr.com
Харланов Дмитрий Валентинович [1], соискатель, mitya999@mail.ru

[1] Южный федеральный университет, г. Таганрог, 347900, Россия

Authors

Alexey V. Labyntsev [1], Cand. of Sci. (Engineering), Associate Professor, Senior Researcher, labyntsev@yandex.ru
Vladimir V. Ignatyev [1], Cand. of Sci. (Engineering), Leading Researcher, vvignatev@sfedu.ru
Vasily V. Vasilyev[1], Junior Researcher, vasily@sunsdr.com
Vitaly A. Mikhaylovsky [1], Junior Researcher, vitaly@sunsdr.com
Dmitry V. Kharlanov [1], Candidate, mitya999@mail.ru

[1] Southern Federal University, Taganrog, 347900, Russian Federation

УДК 004.94; 621.7.04 doi: 10.15827/0236-235X.142.678-685 2023. Т. 36. № 4. С. 678–685

Построение параметризованной 3D-модели рабочей зоны шнека экструдера

Ю.Г. Попов
Г.С. Малов

Ссылка для цитирования
Попов Ю.Г., Малов Г.С. Построение параметризованной 3D-модели рабочей зоны шнека экструдера // Программные продукты и системы. 2023. Т. 36. № 4. С. 678–685. doi: 10.15827/0236-235X.142.678-685
Информация о статье
Поступила в редакцию: 24.04.2023 После доработки: 30.06.2023 Принята к публикации: 01.08.2023

Аннотация. Экструзия является широко распространенным способом переработки материалов в различных отраслях промышленности. При этом конструкция шнека в значительной мере определяет качество экструдата и эффективность самого процесса его получения. В данной статье рассмотрены основные существующие перспективные подходы к определению оптимальных геометрических параметров шнеков. Предложена концепция системы автоматизированного проектирования, базирующаяся на использовании предварительно построенных параметризованных моделей шнеков. Их геометрия определяется переменными, которые могут варьироваться в широких пределах. Такой подход позволяет использовать прикладные программы аналитического определения важнейших геометрических характеристик шнека, обеспечивать быструю генерацию 3D-модели и проверку методом конечных элементов с одновременной оптимизацией параметров. Рассмотрено построение в Компас-3D параметризованной рабочей зоны шнека, включающей основной гребень переменного шага, барьерный гребень, а также зону основной и барьерной нарезки. Дана оценка основных методов построения сложных геометрических поверхностей в Компас-3D с точки зрения производительности моделирования. Предлагаемый метод может значительно увеличить скорость и улучшить точность проектирования шнеков экструдера, а также повысить качество и эффективность производственного процесса экструзии.
Ключевые слова: шнек экструдера, параметрическая модель, Компас-3D, твердотельное моделирование, поверхностное моделирование

Введение. В промышленности экструзия применяется уже 200 лет. Ранние поршневые и короткошнековые экструдеры использовались в 19 веке для изготовления труб из мягких металлов и переработки каучука. С 1930-х годов появились экструдеры современного типа, снабженные электронагревателями и шнеками большой длины для изготовления термопластов. С тех пор технология значительно продвинулась. Однако, несмотря на достижения в методиках проектирования и производства, теоретическое понимание процесса экструзии и связанных с ним параметров остается ограниченным [1]. Различия в реологических и технологических характеристиках всего многообразия перерабатываемых материалов делают невозможным прямое заимствование элементов теории экструзии из одной отрасли в другую [2]. Это, в свою очередь, приводит к тому, что ряд параметров экструдеров вообще и оптимальных геометрических характеристик шнеков, в частности, подбирается в процессе производства итерационно, методом проб и ошибок. Такой эмпирический подход к проектированию, помимо увеличения сроков, нередко ведет к снижению эффективности работы и повышенному износу оборудования.

В связи с этим все большее значение приобретают подходы, основанные на имитационных компьютерных моделях процесса экструзии. В них объединяют выходные характеристики (такие как производительность, потребляемая мощность, давление и температура) с регулируемыми параметрами процесса и данными о материалах [3]. Ввиду сложности моделирования экструзии его обычно осуществляют по частям, создавая математические модели отдельных этапов общего технологического процесса.

- *Транспортировка шнеком твердой фракции материала*. Это достаточно изученный процесс, первые публикации, описывающие его математически, появились еще в 1950-х годах. Тем не менее, сегодня не существует единой модели, учитывающей многообразие материалов и режимов работы. Перспективным для решения данной задачи считается использование метода дискретных элементов (DEM) [4]. Теоретически с его помощью можно учитывать любые виды твердых материалов, особенности подачи, геометрию и скорость вращения шнека.
- *Процесс расплавления материала*. Преимущественно модели основаны на описании механизма плавления и опираются на многочисленные экспериментальные исследования. При этом учитываются свойства материала, режимы нагрева, геометрия шнека [5] и другие

данные. Однако более перспективным представляется подход, основанный на расчетах средствами вычислительной гидродинамики (CFD). При этом не требуется знание механизма плавления и не имеет значения сам тип экструзии [6]. Это обеспечивает универсальность методики, однако требует значительной вычислительной мощности при моделировании.

- *Транспортировка шнеком расплава*. Базовые подходы преимущественно рассматривают жидкий материал как неньютоновскую жидкость в двухмерном [7] или трехмерном [8] течении. Также используются и одномерные модели, которые показывают удовлетворительные результаты, но в ограниченном диапазоне условий [9]. В настоящее время активно развивается моделирование методом конечных элементов для одношнековой и двухшнековой экструзии с минимальными упрощениями процесса. Это позволяет получать точные значения скорости и распределение температуры в расплаве, а также соотношение давления и расхода в рабочей зоне экструдера.
- *Моделирование течения материала в матрице*. Итог работы всего технологического процесса экструзии – получение качественного экструдата. Моделирование процессов, протекающих непосредственно в матрице, осуществляется в основном средствами конечно-элементного анализа, эти подходы развиты достаточно хорошо и реализованы в таких программных продуктах, как QForm [10] и Altair Inspire Extrude [11].

Системы автоматизированного проектирования экструдеров

По большому счету САПР экструдеров не имеют смысла без использования каких-либо расчетных или эмпирических моделей поведения материала внутри оборудования. Наиболее часто используемым подходом является разбиение канала экструдера на короткие осевые участки, в которых входные параметры материала являются выходными для предыдущего участка. Это позволяет комбинировать различные рабочие элементы шнеков и канала экструдера, получая данные о скорости, температуре и давлении материала в ключевых точках экструдера. Однако при использовании аналитических моделей с значительными упрощениями возникает определенная погрешность в расчетах, которая накапливается на каждом отдельном участке.

Благодаря огромному прогрессу в вычислительной гидродинамике стало возможным проведение комплексных исследований трехмерных моделей рабочего оборудования экструдера [12]. Такой подход принципиально является более правильным, так как экструзия представляет собой непрерывный процесс взаимодействия шнека, канала экструдера, матрицы и материала. И все преобразования и изменения материала также протекают непрерывно (хотя и локализованы в определенных зонах рабочего оборудования). Однако такой подход (в отличие от аналитических методов) не имеет обратного решения, то есть нельзя по заданным характеристикам процесса таким образом построить модель экструдера, можно только проверить существующий вариант. Значит, поиск глобального проектного решения напрямую с помощью CFD пока невозможен. В настоящее время активно развиваются комплексные автоматизированные методики проектирования экструдеров. При этом используются перспективные подходы, такие как применение элементов искусственного интеллекта в архитектуре САПР [13], генетических алгоритмов и нейронных сетей [14] для построения и оптимизации формы шнека, внедрение комплексных моделей процесса пластификации в САПР [3]. Однако задача построения оптимальной геометрии шнека решается в них только частично. При этом фактически все необходимые составляющие комплексного подхода имеются. Есть множество проработанных и верифицированных аналитических моделей определения технологических параметров и геометрических характеристик шнека при работе с различными материалами. Существуют готовые высококлассные решения вычислительной гидродинамики, такие как Ansys Fluent, PolyFlow, COMSOL Multiphysics, NASTRAN, Flow Vision, позволяющие получать наилучшие результаты анализа потока материала внутри экструдера. Также разработаны высокоэффективные алгоритмы оптимизации, с помощью которых можно получить наилучшие расчетные параметры шнека. Для объединения этих подходов в единый комплекс предлагается концепция САПР, основанная на использовании предварительно построенных параметризованных моделей.

Проектирование на основе параметризованных моделей

Построение 3D-модели само по себе является трудоемким процессом, учитывая сложность геометрии рабочей зоны шнека [13, 15].

Дополнительные сложности вызывает адекватная параметризация, позволяющая проводить итеративную оптимизацию с помощью варьирования ключевых характеристик модели. Существующие программы и модули автоматизированной генерации шнеков создают законченную твердотельную модель. Изменение любых геометрических параметров при этом требует повторной генерации шнека. В связи с этим представляется целесообразным использование программного комплекса, который при создании моделей будет опираться на заранее сформированную базу параметризованных элементов (рис. 1).

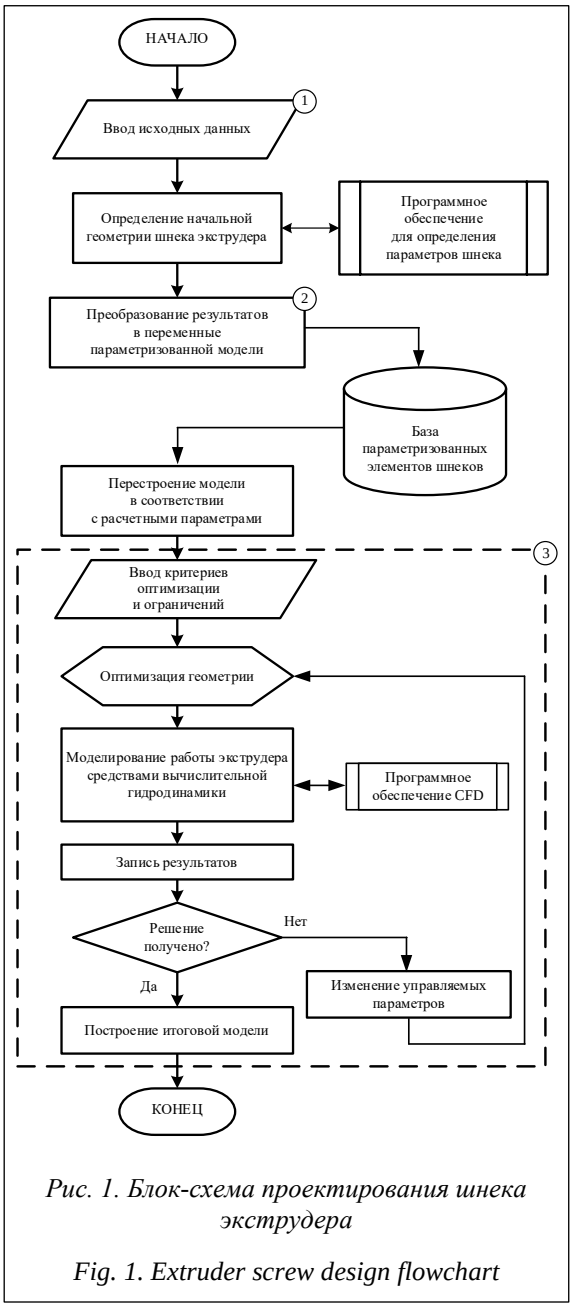


Рис. 1. Блок-схема проектирования шнека экструдера

Fig. 1. Extruder screw design flowchart

В модуле 1 задаются исходные данные для предварительного расчета и построения шнека. Расчет производится аналитическими методами в среде программного комплекса или внешнего приложения. В модуле 2 определяется тип элемента шнека из БД, а полученные расчетные характеристики преобразуются в численные значения переменных параметризованной модели. Блок 3 предназначен для итеративной оптимизации построенной 3D-модели. Здесь сначала определяются набор критериев оптимизации, а также перечень варьируемых параметров модели и пределы их изменения. Далее выполняется цикл оптимизации. На каждой итерации вычисляются критерии при помощи внешнего CFD-приложения. Ввиду ресурсоемкости расчетов алгоритм оптимизации должен быть достаточно совершенным, поэтому целесообразно использовать готовые решения. Например, отечественный программный комплекс IOSO отличается высокой эффективностью поиска решений даже в сложных многокритериальных и многопараметрических задачах [16]. Кроме того, комплекс может напрямую работать с файлами моделей Компас-3D и расчетными модулями ANSYS, Flow Simulation, Flow Vision и Flow-3D. При наличии готовой 3D-модели это позволяет достаточно легко определять оптимизационную задачу, причем как в ручном, так и в автоматизированном режиме при помощи управления API. Завершение процесса оптимизации осуществляется при нахождении заданного числа точек Парето или при невозможности продолжать поиск решений в указанном диапазоне исходных данных.

Формирование и параметризация геометрии рабочей зоны шнека экструдера в Компас-3D

Для параметризации был выбран шнек экструдера (http://www.swsys.ru/uploaded/image/2023-4/5.jpg) с переменным шагом основного витка и переменной глубиной нарезки как основной, так и барьерной зоны. Для полного описания геометрии рабочей зоны шнека потребовалось 18 переменных (рис. 2), плюс четыре значения радиусов скруглений при основном и барьерном гребнях.

Ввиду сложности геометрии все элементы шнека должны строиться как отдельные тела и объединяться только непосредственно перед конечно-элементным анализом. Это позволяет избегать ресурсозатратных булевых операций при проектировании и перестроении модели.

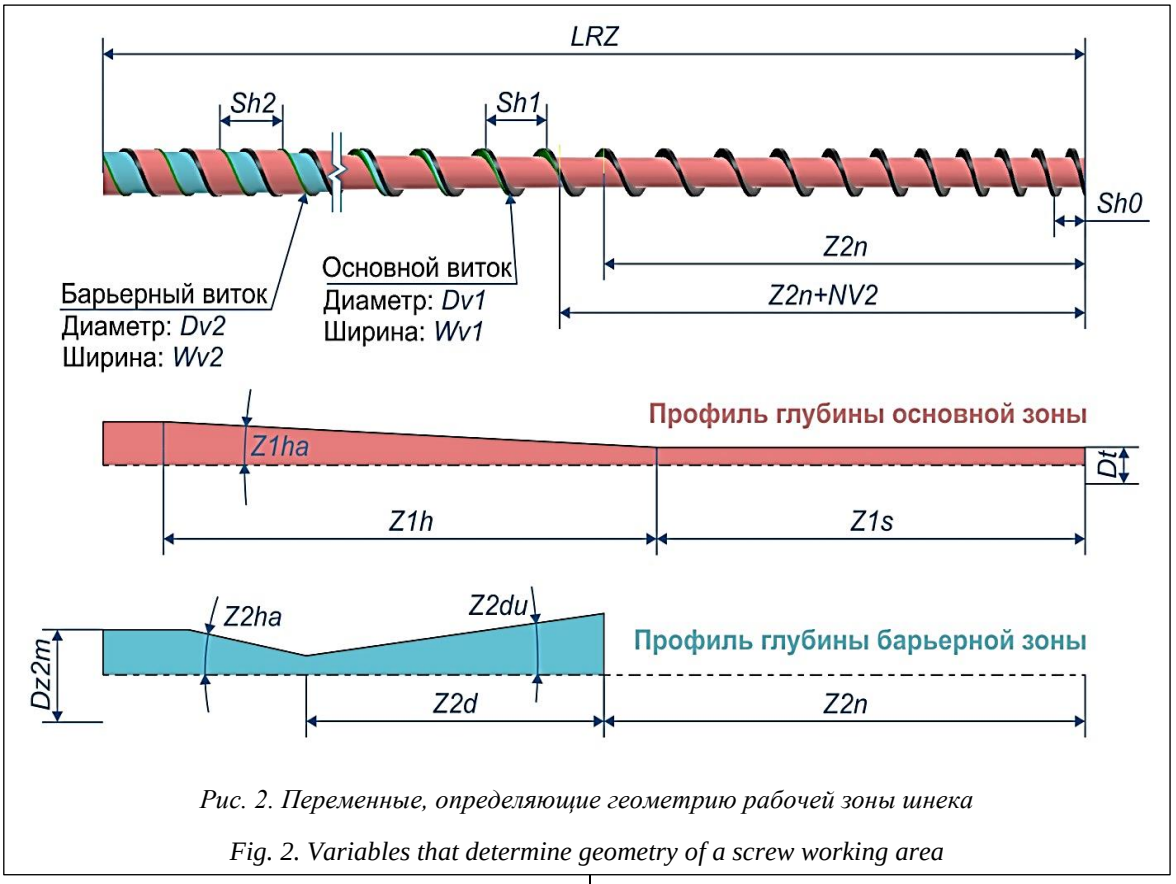


Рис. 2. Переменные, определяющие геометрию рабочей зоны шнека

Fig. 2. Variables that determine geometry of a screw working area

Первым создается цилиндр диаметром *Dt* и длиной *LRZ*. Это тело будет основой для всех последующих построений. Далее необходимо построить основной гребень. Сложность заключается в том, что на участке *Z*2*n* он имеет переменный шаг витка, изменяющийся от *Sh*0 в начале до *Sh*1 в конце участка. В САПР Компас-3D пока нет специализированного инструмента для построения такой спирали, поэтому использован инструмент «кривая по закону».

Формулы для участка спирали с постоянным шагом в прямоугольной системе координат будут следующими:

$$x(t) = \frac{Dv1}{2}\sin\left(2\pi t\left(\frac{1}{Sh1}\right)\right),$$

$$y(t) = \frac{Dv1}{2}\cos\left(2\pi t\left(\frac{1}{Sh1}\right)\right),$$

$$z(t) = t,$$

где $x(t)$, $y(t)$, $z(t)$ – функции кривой вдоль координатных осей; t – переменная длины участка спирали, изменяющаяся от $Z2n$ до LRZ; $Dv1$ – внешний диаметр спирали; $Sh1$ – постоянный шаг спирали.

Представим формулы для участка спирали с переменным шагом в прямоугольной системе координат:

$$x(t) = \frac{Dv1}{2}\sin\left(2\pi\left(\int f(t)dt\right)\right),$$

$$y(t) = \frac{Dv1}{2}\cos\left(2\pi\left(\int f(t)dt\right)\right),$$

$$z(t) = t,$$

$$f(t) = \frac{1}{Sh0 + (Sh1 - Sh0)\left(\frac{t}{Z2n}\right)},$$

где $f(t)$ – функция изменения шага спирали; $Sh0$ – начальный шаг спирали; t – переменная длины участка спирали, изменяющаяся от 0 до $Z2n$.

Проинтегрировав выражения для участка с переменным шагом, можно записать формулу кривой (рис. 3).

На рисунке 3 формулы обозначают следующее: $t < Z2n$? отражает условие, благодаря которому с помощью одной команды можно построить всю спираль с участками переменного и постоянного шагов. Оно проверяет текущую координату t. Если она находится в пределах от нуля до $Z2n$, кривая строится по формуле

$$(DV/2)^* \sin\left(\begin{array}{l} M_PI2^* \\ \left(\left(Z2n^* In\left((Sh1 - Sh0)^* t + Z2n^* Sh0\right)\right)\right. \\ \left./(Sh1 - Sh0)\right) \end{array}\right).$$

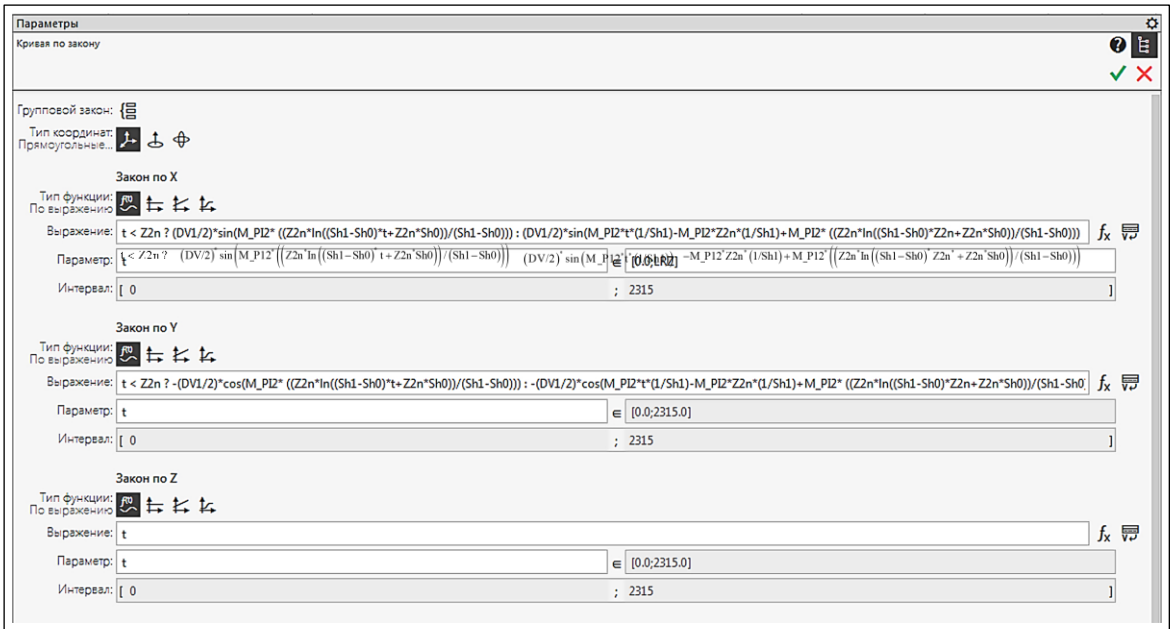


Рис. 3. Формула спирали шнека в Компас-3D

Fig. 3. Screw spiral formula in Kompas-3D

Это участок с переменным шагом. Соответственно, при *t* от *Z2n* до *LRZ* спираль строится по формуле $\left(DV/2\right)^{*}\sin\left(M_PI2^{*}t^{*}\left(1/Sh1\right)\right)$.

Формулой

$$-M_PI2^{*}Z2n^{*}\left(1/Sh1\right)+M_PI2^{*}\left(\left(Z2n^{*}In\left(\left(Sh1-Sh0\right)^{*}Z2n^{*}+Z2n^{*}Sh0\right)\right)/\left(Sh1-Sh0\right)\right)$$

обозначено математическое выражение, по которому начало спирали с постоянным шагом помещается в ту же точку, в которой закончилась спираль с переменным шагом. Это позволяет создать параметризованную кривую без разрывов.

Далее путем смещения спирали вдоль ее оси на величину *Wv*1 строится винтовая поверхность и командой «Придать толщину» – основной гребень (рис. 4). Альтернативный способ, основанный на использовании команды «Элемент по траектории», который заключается в том, что прямоугольное сечение гребня выдавливается вдоль построенной спирали, оказался менее эффективным. Тело, полученное таким образом, перестраивается примерно в четыре раза дольше при изменении параметров модели.

Затем аналогично создается барьерный гребень. Эти элементы естественным образом разбивают поверхность базового цилиндра *Dt* на зоны: основную и барьерную. Далее каждой зоне придается толщина, равная высоте соответствующего гребня. На последнем этапе строятся эскизы профилей зон, после чего с помощью операции «Вращение» с пересечением тел основной и барьерной зонам придается окончательная форма (рис. 5).

Были опробованы также другие способы моделирования рабочей зоны шнека экструдера. Построение гребней по траектории дало неоправданное увеличение времени обработки модели при перестроении. Также создавались поверхностные границы элементов рабочей зоны с последующим преобразованием их в твердотельные объекты. Однако это потребовало

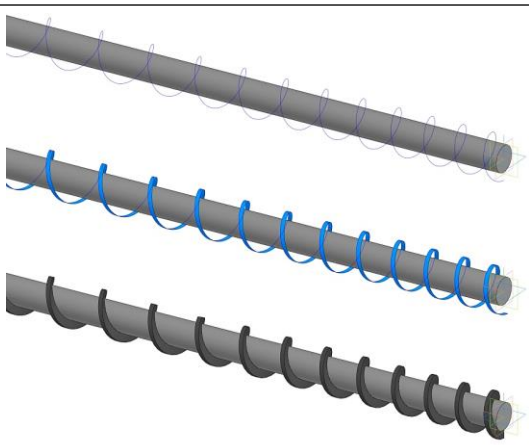


Рис. 4. Построение основного гребня

Fig. 4. Main ridge design

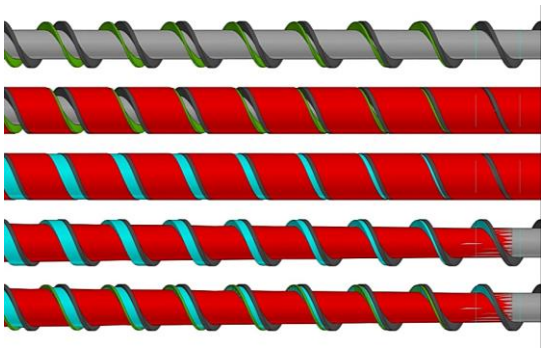


Рис. 5. Построение основной и барьерной зон

Fig. 5. Main and barrier zone design

значительно большего (на 42 %) числа операций. Кроме того, возросло число ошибок при перестроениях модели, которые требовали ручной корректировки операций.

Описанный выше метод через построения с помощью команды «Придать толщину» оказался самым надежным. Ошибки возникают только при вводе несовместимых с геометрией численных значений параметров. Например, основной и барьерный гребни не должны пересекаться. Соответственно, шаг Sh2 должен быть больше Sh1. Кроме того, нужно убедиться, что гребни не пересекутся на рабочей длине шнека, что возможно при слишком большой разнице в шаге. Это должно проверяться на моменте определения ограничений оптимизационной задачи. Впрочем, даже при наличии возможных ошибок построения алгоритм оптимизации может продолжать работу, отбрасывая точки, в которых перестроение шнека не удалось.

Заключение

Рассмотренная в данной статье методика проектирования рабочей зоны шнека экструдера может выполняться как в ручном, так и в автоматизированном режиме при помощи средств API. Предлагаемый метод подходит для создания и более сложных конструкций шнеков с использованием большего числа параметров. Причем это возможно в рамках одной и той же базы сформированных моделей. В данной методике нет необходимости в строго одинаковом числе варьируемых параметров шнеков. В сочетании с использованием приложений CFD-расчетов и современных алгоритмов многопараметрической оптимизации это позволит повысить точность и скорость процесса проектирования шнеков экструдера под конкретные технологические режимы работы. Вместе с тем построение параметризованной модели в САПР Компас-3D сопряжено с определенными сложностями ввиду отсутствия инструмента прямого построения винтовой линии с переменным шагом. Стоит отметить, что в Компас-3D любая трехмерная геометрия, опирающаяся при построении на команду «кривая по закону» перестраивается медленнее из-за необходимости расчета формулы с несколькими переменными с определенной точностью. Для сравнения: в системе автоматизированного проектирования Autodesk Inventor начиная с версии 2019 есть инструмент построения винтовой линии с переменным шагом. Благодаря этому параметризованную модель можно построить всего за 22 операции (против 29 у Компас-3D) и перестроение при изменении параметров производится быстрее при прочих равных. Тем не менее, применение отечественных средств автоматизированного проектирования в настоящее время предпочтительнее. В связи с этим дальнейшее развитие данной работы подразумевает создание программной оболочки для ввода исходных данных, взаимодействия с Компас-3D для перестроения параметризованных моделей из базы и формирования задачи для программной платформы управления расчетами и оптимизации IOSO в связке с программой CFD-анализа Flow Vision.

Список литературы

1. Hyvärinen M., Jabeen R., Kärki T. The modelling of extrusion processes for polymers–a review. Polymers, 2020, vol. 12, no. 6, art. 1306. doi: 10.3390/polym12061306.

2. Пятов В.В., Голубев А.Н., Ширяев П.С. Методика проектирования шнековых машин для экструзии пластичных сред // Вестн. ВГТУ. 2017. № 2. С. 45–52.

3. Iwko J., Steller R., Wróblewski R., Kaczmar J. Experimental verification of computer model for polymer plastication process in injection molding. Polimery, 2015, vol. 60, no. 10, pp. 644–656. doi: 10.14314/polimery.2015.644.

4. Trippe J., Schöppner V. Modeling of solid conveying pressure throughput behavior of single screw smooth barrel extruders under consideration of backpressure and high screw speeds. Int. Polymer Processing, 2018, vol. 33, no. 4, pp. 486–496. doi: 10.3139/217.3507.

5. Jin X.-M., Jia M.-Y., Xue P., Cai J.-Ch., Pan L., Yu D.-Q. Study on the melting performance of single screw extruder with grooved melting zone and barr screw. J. of Materials Processing Tech., 2014, vol. 214, no. 12, pp. 2834–2842. doi: 10.1016/j.jmatprotec.2014.06.022.

6. Lewandowski A., Wilczynski K. General model of polymer melting in extrusion process. Polimery, 2018, vol. 63, no. 06, pp. 444–452. doi: 10.14314/polimery.2018.6.5.

7. Pachner S., Löw-Baselli B., Affenzeller M., Miethlinger J. A generalized 2D output model of polymer melt flow in single-screw extrusion. Int. Polymer Processing, 2017, vol. 32, no. 2, pp. 209–216. doi: 10.3139/217.3326.

8. Marschik C., Roland W., Löw-Baselli B., Miethlinger J. A heuristic method for modeling three-dimensional non-Newtonian flows of polymer melts in single-screw extruders. J. of Non-Newtonian Fluid Mechanics, 2017, vol. 248, pp. 27–39. doi: 10.1016/j.jnnfm.2017.08.007.

9. Durin A., De Micheli P., Nguyen H.-C., David C., Valette R., Vergnes B. Comparison between 1D and 3D approaches for twin-screw extrusion simulation. Int. Polymer Processing, 2014, vol. 29, no. 5, pp. 641–648. doi: 10.3139/217.2951.

10. QForm Extrusion. URL: https://qform3d.co.uk/products/qfextrusion (дата обращения: 20.04.2023).

11. Design for Efficient Metal Extrusion. Inspire Extrude. URL: https://altair.com/inspire-extrude-metal (дата обращения: 20.04.2023).

12. Lewandowski A., Wilczyński K. Global modeling for single screw extrusion of viscoplastics. Int. Polymer Processing, 2020, vol. 35, no. 1, pp. 26–36. doi: 10.3139/217.3866.

13. Зубкова Т.М., Мустюков Н.А., Токарева М.А. Построение архитектуры САПР одношнековых экструдеров с применением элементов искусственного интеллекта // Программные продукты и системы. 2016. Т. 29. № 4. С. 176–182. doi: 10.15827/0236-235X.116.176-182.

14. Kowalski R.J., Pietrysiak E., Ganjyal G.M. Optimizing screw profiles for twin-screw food extrusion processing through genetic algorithms and neural networks. J. of Food Eng., 2021, vol. 303, art. 110589. doi: 10.1016/j.jfoodeng.2021.110589.

15. Шилько Д.А. Построение 3D модели шнека экструдера на основе уравнений Навье–Стокса // Изв. ТулГУ. Технич. науки. 2022. № 2. С. 577–581.

16. Goriachkin E., Popov G., Baturin O., Matveev V., Volkov A. Method for improving multi-stage compressor parameters using the IOSO optimizer. Proc. 10th Int. Conf. SIMULTECH, 2020, pp. 163–170. doi: 10.5220/0009769601630170.

Software & Systems doi: 10.15827/0236-235X.142.678-685 2023, vol. 36, no. 4, pp. 678–685

Constructing a parametric 3D model of an extruder screw working area

Yury G. Popov
Gleb S. Malov

For citation
Popov, Yu.G., Malov, G.S. (2023) 'Constructing a parametric 3D model of an extruder screw working area', *Software & Systems*, 36(4), pp. 678–685 (in Russ.). doi: 10.15827/0236-235X.142.678-685
Article info
Received: 24.04.2023 After revision: 30.06.2023 Accepted: 01.08.2023

Abstract. Extrusion is a widespread way of processing materials in various industries. At the same time, screw design largely determines extrudate quality and its production process efficiency. This paper discusses the main existing and promising approaches to determining optimal geometric parameters of screws. The authors propose a concept of a computer-aided design system based on using pre-built parameterized screw models; screw geometry is determined by variables that can vary over a wide range. This approach allows using applications for analytical determination of the most important screw geometry, fast 3D model generation and finite element verification while optimizing parameters. The paper considers constructing a parametrized working area of a screw in Compass-3D. The working area includes the main comb of variable pitch, a barrier comb, as well as the main and barrier cutting zone. The authors assess the main methods for constructing complex geometric surfaces in Compass-3D in terms of modeling performance. The proposed method can significantly increase extruder screw design speed and accuracy, improve the extrusion production process quality and efficiency.
Keywords: extruder screw, parametric model, Kompass-3D, solid modeling, surface modeling

References

1. Hyvärinen, M., Jabeen, R., Kärki, T. (2020) 'The modelling of extrusion processes for polymers–a review', *Polymers*, 12(6), art. 1306. doi: 10.3390/polym12061306.

2. Pyatov, V.V., Golubev, A.N., Shiryaev, P.S. (2017) 'Design technique of screw-type machines for extrusion of plastic media', *Vestn. of VSTU*, (2), pp. 45–52 (in Russ.).

3. Iwko, J., Steller, R., Wroblewski, R., Kaczmar, J. (2015) 'Experimental verification of computer model for polymer plastication process in injection molding', *Polimery*, 60(10), pp. 644–656. doi: 10.14314/polimery.2015.644.

4. Trippe, J., Schöppner, V. (2018) 'Modeling of solid conveying pressure throughput behavior of single screw smooth barrel extruders under consideration of backpressure and high screw speeds', *Int. Polymer Processing*, 33(4), pp. 486–496. doi: 10.3139/217.3507.

5. Jin, X.-M., Jia, M.-Y., Xue, P., Cai, J.-C., Pan, L., Yu, D.-Q. (2014) 'Study on the melting performance of single screw extruder with grooved melting zone and barr screw', *J. of Materials Processing Tech.*, 214(12), pp. 2834–2842. doi: 10.1016/j.jmatprotec.2014.06.022.

6. Lewandowski, A., Wilczynski, K. (2018) 'General model of polymer melting in extrusion process', *Polimery*, 63(06), pp. 444–452. doi: 10.14314/polimery.2018.6.5.

7. Pachner, S., Löw-Baselli, B., Affenzeller, M., Miethlinger, J. (2017) 'A generalized 2D output model of polymer melt flow in single-screw extrusion', *International Polymer Processing*, 32(2), pp. 209–216. doi: 10.3139/217.3326.

8. Marschik, C., Roland, W., Löw-Baselli, B., Miethlinger, J. (2017) 'A heuristic method for modeling three-dimensional non-Newtonian flows of polymer melts in single-screw extruders', *J. of Non-Newtonian Fluid Mechanics*, 248, pp. 27–39. doi: 10.1016/j.jnnfm.2017.08.007.

9. Durin, A., De Micheli, P., Nguyen, H.-C., David, C., Valette, R., Vergnes, B. (2014) 'Comparison between 1D and 3D approaches for twin-screw extrusion simulation', *Int. Polymer Processing*, 29(5), pp. 641–648. doi: 10.3139/217.2951.

10. *QForm Extrusion*, available at: https://qform3d.co.uk/products/qfextrusion (accessed April 20, 2023).

11. *Design for Efficient Metal Extrusion. Inspire Extrude*, available at: https://altair.com/inspire-extrude-metal (accessed April 20, 2023).

12. Lewandowski, A., Wilczyński, K. (2020) 'Global modeling for single screw extrusion of viscoplastics', *Int. Polymer Processing*, 35(1), pp. 26–36. doi: 10.3139/217.3866.

13. Zubkova, T.M., Mustyukov, N.A., Tokareva, M.A. (2016) 'Single screw extruder engineering with artificial intelligence elements', *Software & Systems*, 29(4), pp. 176–182 (in Russ.). doi: 10.15827/0236-235X.116.176-182.

14. Kowalski, R.J., Pietrysiak, E., Ganjyal, G.M. (2021) 'Optimizing screw profiles for twin-screw food extrusion processing through genetic algorithms and neural networks', *J. of Food Eng.*, 303, art. 110589. doi: 10.1016/j.jfoodeng.2021.110589.

15. Shilko, D.A. (2022) 'Construction of a 3D model of an extruder screw based on the Navier–Stokes equations', *Proc. of the TSU. Tech. Sci.*, (2), pp. 577–581 (in Russ.).

16. Goriachkin, E., Popov, G., Baturin, O., Matveev, V., Volkov, A. (2020) 'Method for Improving Multi-stage Compressor Parameters using the IOSO Optimizer', *Proc. 10th Int. Conf. SIMULTECH*, pp. 163–170. doi: 10.5220/0009769601630170.

Авторы
Попов Юрий Германович [1], к.т.н., доцент, popovyug@ystu.ru
Малов Глеб Сергеевич [1], аспирант, yargleb@mail.ru

[1] Ярославский государственный технический университет, г. Ярославль, 150999, Россия

Authors
Yury G. Popov [1], Cand. of Sci. (Engineering), Associate Professor, popovyug@ystu.ru
Gleb S. Malov [1], Postgraduate Student, yargleb@mail.ru

[1] Yaroslavl State Technical University, Yaroslavl, 150999, Russian Federation

УДК 519.688 doi: 10.15827/0236-235X.142.686-694 2023. Т. 36. № 4. С. 686–694

Программный расчет комплексного гармонического колебания с флуктуацией фазы для беспроводных систем связи

Ф.А. Цветков
В.В. Васильев
Вит.А. Михайловский
Вал.А. Михайловский
А.В. Ломакина

Ссылка для цитирования
Цветков Ф.А., Васильев В.В., Михайловский Вит.А., Михайловский Вал.А., Ломакина А.В. Программный расчет комплексного гармонического колебания с флуктуацией фазы для беспроводных систем связи // Программные продукты и системы. 2023. Т. 36. № 4. С. 686–694. doi: 10.15827/0236-235X.142.686-694
Информация о статье
Поступила в редакцию: 19.09.2023 После доработки: 10.10.2023 Принята к публикации: 19.10.2023

Аннотация. В статье описывается программа генерирования последовательности временных отсчетов комплексного гармонического колебания с флуктуацией фазы в виде фликкер-шума с заданной спектральной плотностью. Предлагается решение задачи оценки устойчивости современных каналов передачи цифровых данных к помехам и шумам, что на первом этапе проектирования канала выясняется с помощью моделирования. В моделирующей программе исходной характеристикой генерируемого колебания является желаемая спектральная плотность флуктуаций его начальной фазы. На ее основании рассчитывается амплитудно-частотная характеристика формирующего цифрового фильтра. На вход фильтра подается некоррелированная нормальная псевдослучайная последовательность, на выходе получается последовательность временных отсчетов флуктуирующей начальной фазы комплексного гармонического колебания с некоторой средней частотой (в частности, равной нулю). Для каждого из этих отсчетов вычисляются вещественная и мнимая компоненты выходного колебания генератора. Их совокупность и является искомым комплексным гармоническим колебанием с флуктуацией фазы. Программа написана на языке графического программирования NI LabVIEW, что позволяет видеть все этапы ее работы. Для оценки качества функционирования разработанной программы выполнено тестовое моделирование колебания высокостабильного генератора E5052B Signal Source Analyzer. Последующий численный анализ параметров полученного колебания и сравнение их с известными параметрами генератора подтвердили корректность работы программы. Предложенный в статье способ моделирования влияния фазовых шумов на сигналы в канале связи может быть полезен для специалистов в области исследования и проектирования различных систем беспроводной связи, а также в процессе изучения основ передачи радиосигналов в рамках образовательного процесса в высших учебных заведениях.
Ключевые слова: программа, фазовый шум, программная модель, спектральная плотность мощности фазового шума, колебание с фазовым шумом, LabVIEW
Благодарности. Работа поддержана Южным федеральным университетом, проект № ВнГр/23-01-РТ

Введение. В современных системах связи и навигации широко применяются высокостабильные гармонические колебания. Для таких систем важным параметром является оценка флуктуаций их мгновенной частоты. Согласно рекомендациям IEEE, для характеристики этих флуктуаций используются спектральная плотность мощности начальной фазы и фазовых шумов в одиночной боковой полосе и другие подобные параметры. В данном исследовании речь идет о таком параметре фазового шума, как спектральная плотность мощности фликкер-шума.

Причиной возникновения фазового шума является внезапное отклонение фазы несущей волны, что обусловливает существенное изменение спектральной плотности мощности выходного сигнала. В конечном итоге это может привести к падению общей производительности системы из-за частичной потери согласованности между параметрами передаваемого и принятого сигналов в процессе передачи данных, что и дает толчок к более тщательному изучению фазового шума гетеродинов и созданию моделей, способных дать глубокое понимание влияния таких шумов на разные системы связи [1–3].

Основным фактором, влияющим на снижение производительности системы беспроводной связи, является несовершенство генераторов, которые используются для преобразования немодулированного сигнала в сигнал в заданной полосе частот на стороне передатчика и для обратного преобразования сигнала на стороне приемника. Идеальные генераторы производят синусоидальные волны со стабильной частотой и опорной фазой. На практике они сочетают в себе реальные пассивные и активные элементы, поэтому синусоида, создаваемая реальным генератором, имеет случайные колебания фазы.

Проектирование канала передачи цифровых данных начинается с его предварительного численного моделирования. Обычно в качестве дестабилизирующего фактора используют аддитивный белый гауссов шум, реже вводят в модель генератора фазовый шум [4, 5]. Однако в этих моделях практически не используется в качестве исходного параметра спектральная плотность мощности для фазового шума. В настоящей работе сделана попытка создать вариант программной модели генератора комплексного гармонического напряжения со случайной начальной фазой, исходным параметром которой является спектральная плотность мощности начальной фазы.

Современные исследования по проблеме моделирования фазового шума в различных системах связи

Фазовый шум – очень важный и достаточно сложный объект изучения для разработчиков любых видов систем связи. Хотя причины появления таких шумов и способы их подавления довольно подробно изучены, по-прежнему нет полного понимания их происхождения и твердой математической основы для систем с фазовым шумом [6, 7].

Уровень фазового шума является важным параметром во многих системах, поскольку может существенно ограничивать их производительность и приводить к неправильному приему информации. Особенно важно создавать подобные модели для исследования систем с ортогональным частотным уплотнением (Orthogonal Frequency Division Modulation, OFDM), так как, несмотря на высокую эффективность использования спектра и устойчивость к частотной избирательности канала, одним из существенных недостатков OFDM является чувствительность к фазовому шуму. При этом даже небольшой шум приводит к значительной потере производительности. В работе [3] проанализировано влияние фазового шума на системы OFDM и представлена модель, позволяющая реализовать методы для его уменьшения.

Фазовый шум можно измерить в различных точках системы или определить с помощью численного моделирования и математических моделей. При численном моделировании каналов связи возникает необходимость ввести в программную модель канала связи некую случайную нестабильность, обусловленную нестабильностью частоты (начальной фазы) опорных генераторов. Поэтому актуальным является моделирование генераторов с нестабильной начальной фазой, статистические характеристики которой могут быть заданы. Так, например, в программной среде Matlab (Simulink) имеется модель ex_simrf_phase_noise, позволяющая сгенерировать колебание с фазовым шумом. Аналогичная модель существует и в среде программирования LabVIEW – MT Add Phase Noise, с ее помощью можно добавить шумовую составляющую к фазе вектора входного комплексного сигнала. Эти модели работают только в соответствующих средах и, как правило, не дают понимания тонкостей реализованных в них алгоритмов. Поэтому при получении неожиданных результатов работы дополнительно приходится тщательно исследовать свойства этих моделей, чтобы убедиться, что причина не в них.

В результате данного исследования создана программная модель формирователя комплексного колебания с флуктуацией фазы в виде фликкер-шума. Она использует простейшие функциональные узлы языка программирования LabVIEW, поддающиеся легкой проверке. Модель открытая, при необходимости в нее можно вносить изменения. Выбор языка LabVIEW обусловлен еще и тем, что на нем разрабатывалась программная модель канала передачи цифровых данных, публикация сведений о которой предполагается в ближайшее время.

Разработка программы расчета отсчетов искомого колебания

Генератор комплексного колебания с флуктуацией фазы представляет собой программу на языке LabVIEW, рассчитывающую последовательность комплексных отсчетов искомого колебания для конечного количества равноотстоящих моментов времени. Блок-схема программы показана на рисунке 1.

Сначала по задаваемым параметрам желаемой спектральной плотности фазового шума (вида $1/f^{\alpha}$) рассчитывается необходимый модуль *амплитудно-частотной характеристики* (АЧХ) формирующего вещественного фильтра в виде вещественной четной функции частоты (с нулевым сдвигом фаз).

Дальнейшие вычисления выполняются в цикле в виде последовательности операций:

– создается последовательность псевдослучайных отсчетов белого нормального шума и вычисляется ее спектр;

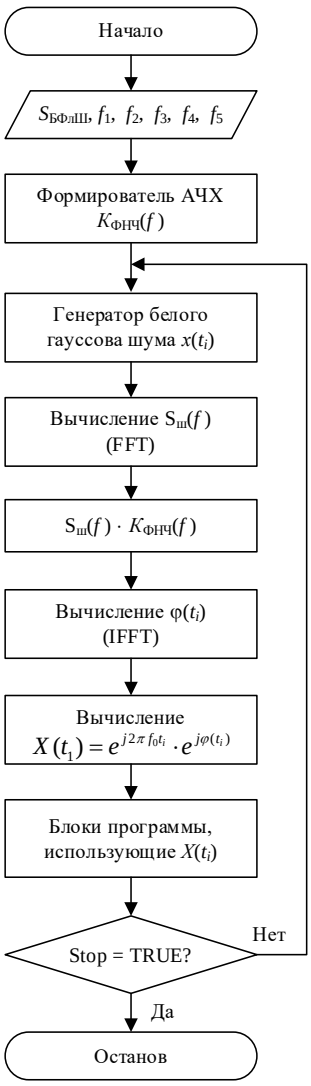


Рис. 1. Блок-схема программы

Fig. 1. Program block diagram

– спектр шума умножается на АЧХ формирующего фильтра, то есть определяется спектр формируемого фликкер-шума;

– от полученного спектра фликкер-шума вычисляется обратное преобразование Фурье с получением вещественной функции времени – мгновенных значений начальной фазы (фликкер-шума фазы) генерируемого комплексного колебания;

– для каждого из моментов времени интервала существования генерируемого комплексного гармонического колебания вычисляется его комплексный отсчет;

– полученная последовательность отсчетов генерируемого колебания со случайной начальной фазой передается в блоки программы, использующей ее.

Этот процесс повторяется необходимое для получения статистически устойчивого результата моделирования количество раз, то есть до остановки оператором или до выполнения заданного заранее условия. Рассмотрим подробнее действия в каждом из модулей алгоритма.

При определении АЧХ формирующего фильтра используется модель спектральной плотности мощности фазовых шумов из [8], показанная на рисунке 2 (обе оси в логарифмических масштабах).

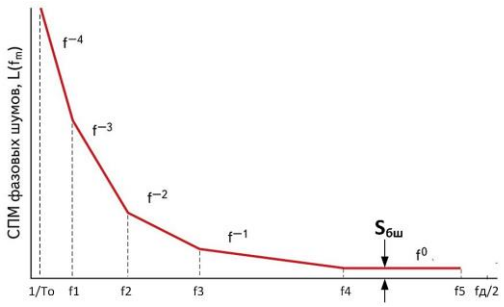


Рис. 2. Параметры модели спектральной плотности мощности фазовых шумов

Fig. 2. Parameters of the power spectral density model of phase noise

На рисунке отмечены частоты границ участков, в пределах которых характер частотной зависимости неизменен. Используются обозначения: T_o – длительность отрезка формируемого комплексного гармонического колебания; $f_д$ – частота дискретизации; $S_{бш}$ – уровень спектральной плотности мощности на участке «белый шум» (от частоты f_4 до частоты f_5). Кроме того, на частоте 0 и в полосе частот от f_5 до $f_д/2$ уровень спектральной плотности мощности принимается равным 0.

Вся программа разбивается на две подпрограммы: «АЧХ фильтра.vi» – от начала и до завершения модуля «Формирователь АЧХ» и «Генер с фаз шумом.vi», включающую остальные модули, являющиеся телом цикла. Функциональная диаграмма подпрограммы «АЧХ фильтра.vi» расчета АЧХ формирующего фильтра показана на рисунке 3.

Входные параметры подпрограммы: $T_{вх}$, с – длительность отрезка формируемого комплексного гармонического колебания; $f_д$, Гц – частота дискретизации; Уровень БШ, дБ – уровень спектральной плотности мощности на участке «белый шум»; f_i, Гц – массив частот границ участков f_1, f_2, f_3, f_4, f_5.

Выходные данные подпрограммы: N – количество отсчетов формируемого комплексного

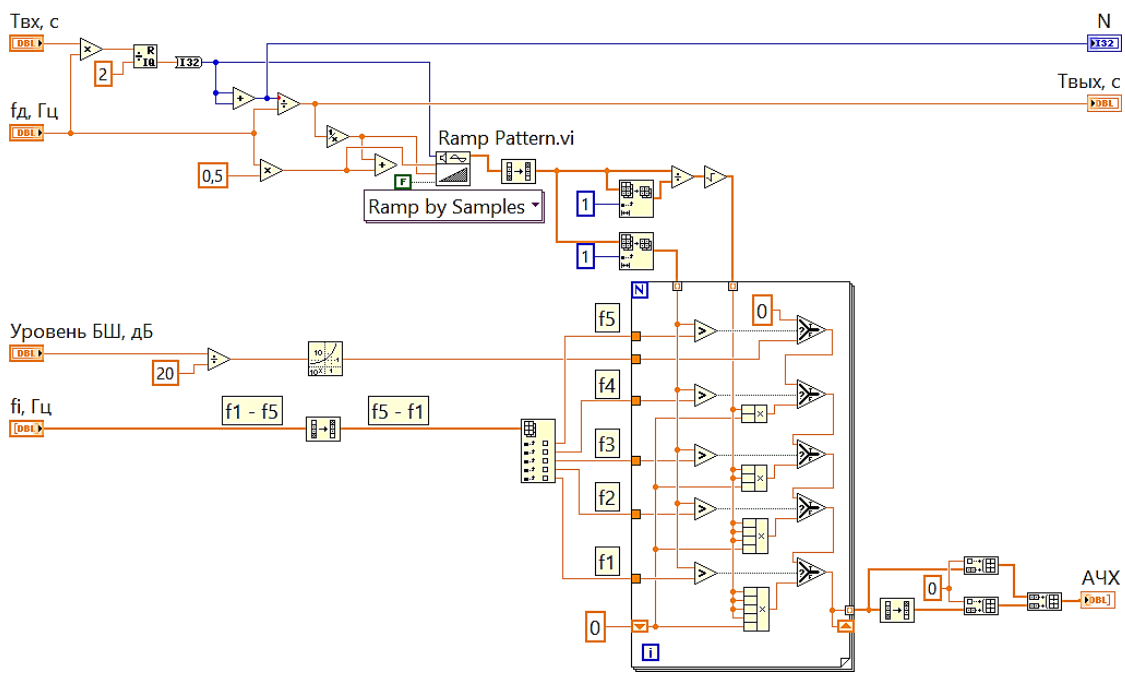


Рис. 3. Функциональная диаграмма подпрограммы расчета АЧХ формирующего фильтра

Fig. 3. Functional diagram of the subroutine for calculating a forming filter frequency response

гармонического колебания (оно же – количество частотных точек всех его спектров); $T_{вых}$, с = = T_o – фактическая длительность отрезка формируемого комплексного гармонического колебания; АЧХ – массив вещественных отсчетов коэффициента передачи формирующего фильтра в функции частоты в полосе $[-f_д/2, +f_д/2)$ с шагом $f_д/N$.

При запуске подпрограммы вычисляется произведение $T_{вх} \times f_д$, которое может быть и с дробной частью. Результат делится нацело на 2; удвоенная целая часть является целым четным числом, которое выдается как N. Четность N необходима для корректного применения быстрого преобразования Фурье. По найденному корректному значению N вычисляется фактическая длительность $T_{вых}$, с отрезка моделируемого комплексного гармонического колебания.

Так как АЧХ в диапазоне частот $[-f_д/2, +f_д/2)$ является четной функцией относительно частоты 0, вычисляется только половина ее значений – в диапазоне частот от $1/T_{вых}$ до $f_д/2$. Массив равноотстоящих частот формируется узлом Ramp pattern.vi. При вычислении значений АЧХ используется корень квадратный из отношения значений соседних частот (корень квадратный потому, что АЧХ – это отношение выходного и входного напряжений, а не мощностей). Расчет выполняется от большей частоты к меньшей, для чего массив расчетных частот реверсируется, реверсируется также и порядок частот в массиве f_i, Гц. Значения АЧХ вычисляются в цикле, при этом текущее значение частоты сравнивается со значениями частот f_i, и по мере его уменьшения, то есть перехода к участкам АЧХ, более близким к частоте 0, растет крутизна изменения АЧХ. После цикла по вычисленным значениям АЧХ для положительных частот конструируется АЧХ для всего диапазона частот $[-f_д/2, +f_д/2)$, причем на частоте 0 значение АЧХ задается равным нулю.

На лицевой панели подпрограммы «АЧХ фильтра.vi» отражены заданные входные параметры и полученные результаты расчетов для массива значений АЧХ, сами значения показаны только в окрестности частоты 0 (http://www.swsys.ru/uploaded/image/2023-4/7.jpg).

Рассчитанная АЧХ (по вертикали и горизонтали масштабы логарифмические) позволяет проследить характер изменения АЧХ на разных участках частот и соответствие параметров АЧХ заданным значениям (http://www.swsys.ru/uploaded/image/2023-4/8.jpg).

В подпрограмме «Генер с фаз шумом.vi» выполняется расчет массива отсчетов комплексного гармонического колебания с флуктуирующей начальной фазой. Функциональная диаграмма подпрограммы показана на рисунке 4.

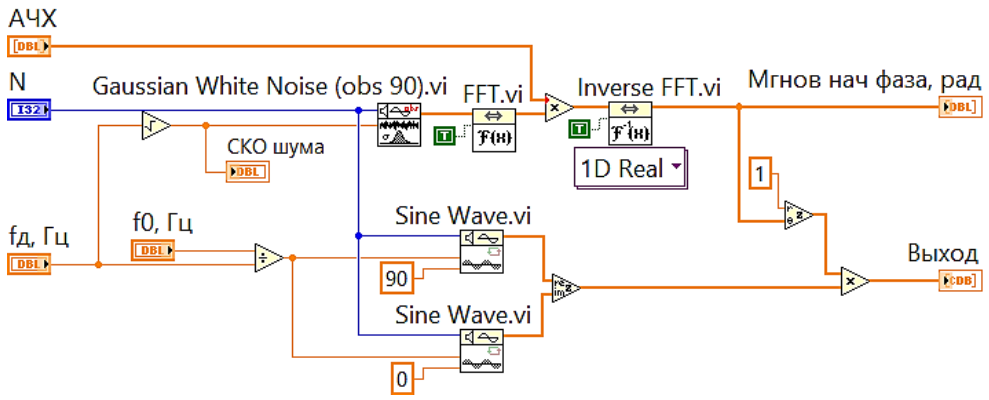


Рис. 4. Функциональная диаграмма подпрограммы «Генер с фаз шумом.vi»

Fig. 4. Functional diagram of Generator with phase noise.vi subroutine

Узел Gaussian White Noise (obs 90).vi формирует реализацию псевдослучайного шума в виде последовательности N вещественных отсчетов белого гауссова шума со среднеквадратическим отклонением $\sigma = \sqrt{f_д}$. Такое значение σ обеспечивает спектральную плотность мощности шума, равную 1 рад^2/Гц (0 дБрад^2/Гц) во всей полосе частот $[-f_д/2, +f_д/2)$.

Полученная реализация шума подается на узел FFT.vi, работающий с параметром shift? = = TRUE и выдающий массив комплексных значений спектральных составляющих в формате математического спектра (в середине массива элемент соответствует частоте 0).

Вычисленный комплексный спектр умножается на значения АЧХ, рассчитанные подпрограммой «АЧХ фильтра.vi», LabVIEW автоматически преобразует вещественные значения АЧХ в комплексные (с мнимой частью 0). С выхода перемножителя массив чисел поступает на узел Inverse FFT.vi, работающий также с параметром shift? = TRUE, то есть принимающий спектр в математическом формате. На выходе этого узла получается временная последовательность отсчетов, рассматриваемая как мгновенные значения случайной начальной фазы генерируемого комплексного гармонического колебания. Эта последовательность используется как фазы вспомогательного массива комплексных чисел, модули которых равны 1.

Вспомогательный массив поэлементно умножается на массив комплексного гармонического колебания с нулевой начальной фазой и частотой f_0, формируемого двумя узлами Sine Wave.vi. На выходе перемножителя получается искомый массив отсчетов комплексного гармонического колебания со случайной начальной фазой.

На лицевой панели подпрограммы «Генер с фаз шумом.vi» представлены входные параметры и расчетные показатели (http://www.swsys.ru/uploaded/image/2023-4/9.jpg).

Пример результата вычисления возможной реализации комплексного гармонического колебания (реальная и мнимая компоненты) иллюстрирует рисунок (http://www.swsys.ru/uploaded/image/2023-4/10.jpg). Для построения использованы параметры, отраженные на лицевых панелях «АЧХ фильтра.vi» и «Генер с фаз шумом.vi».

Анализ полученных результатов

Для оценки корректности функционирования разработанной программы проведена тестовая проверка. Диаграмма тестовой программы, являющаяся примером использования подпрограмм «АЧХ фильтра.vi» и «Генер с фаз шумом.vi», показана на рисунке 5.

В программе можно выделить три части.

Первая часть содержит подпрограмму «АЧХ фазов шума.vi», в которую вводятся параметры моделируемого колебания (за исключением f_0). Результатом ее работы является массив отсчетов АЧХ формирующего фильтра. Эта часть отрабатывает один раз при запуске тестовой программы.

Вторая часть является циклом с не заданным заранее количеством повторений (останов цикла происходит по нажатии оператором кнопки «stop» на лицевой панели). Номер текущего цикла выдается на индикатор $N_{тек}$, что позволяет оператору следить за работой программы. Для ускорения вычислений никакие графики в теле цикла не выдаются. В цикле запускается подпрограмма «Генер с фаз шум.vi». Используя массив отсчетов АЧХ формирую-

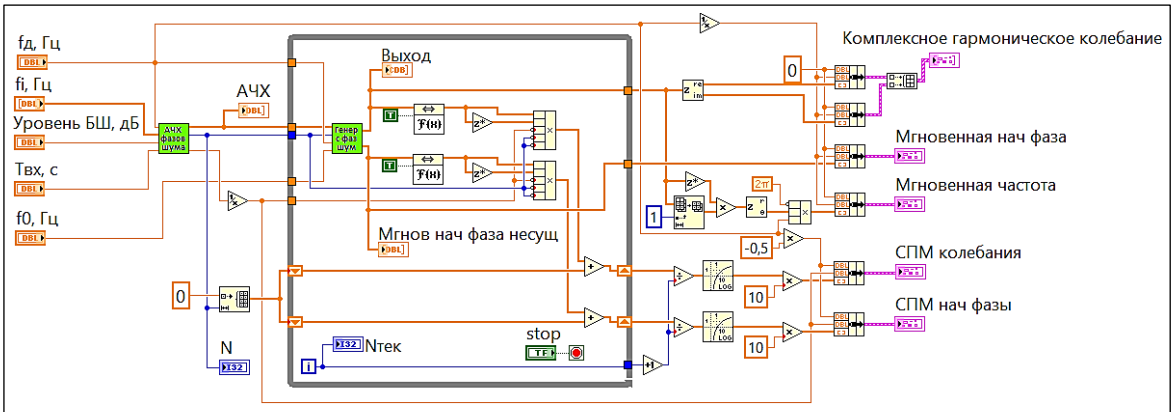


Рис. 5. Функциональная диаграмма тестовой программы

Fig. 5. Test program functional diagram

щего фильтра, она рассчитывает отсчеты комплексного гармонического колебания с фазовым шумом и с центральной частотой f_0. Рассчитанные в каждой итерации цикла массив вещественных отсчетов «Мгнов нач фаза несущ» (отклонения полной фазы генерируемого колебания от линейной составляющей) и массив рассчитанных комплексных отсчетов «Выход» подаются на цифровые индикаторы и на границу цикла, где сохраняются только их значения, рассчитанные в последней итерации. Кроме того, эти же массивы подаются на FFT-преобразователи, вычисляющие их комплексные спектры в математическом формате (средний элемент массива соответствует частоте 0). Для каждого из этих спектров вычисляется квадрат модуля каждой спектральной составляющей, деленный на шаг по оси частот, то есть вычисляется оценка спектральной плотности мощности. Для получения статистически корректных результатов массивы оценок вычисляются многократно и накапливаются как суммы массивов с помощью регистров сдвига.

По завершении цикла отрабатывает третья часть тестовой программы, которая выдает на индикаторы лицевой панели:

– графики *Re*- и *Im*-компонентов сгенерированного комплексного сигнала («Комплексное гармоническое колебание») и график мгновенной фазы («Мгновенная нач фаза») (оба графика отражают данные по последней итерации цикла);

– график «СПМ колебания»;

– график «СПМ нач фазы» (по оси частот – отклонения от частоты f_0);

– график мгновенной частоты как функции времени, рассчитанный по отсчетам массива «Выход».

В качестве примера смоделировано колебание, генерируемое в приборе Keysight Technologies E5052B Signal Source Analyzer. В [8–10] приведен результат измерения фазовых шумов этого колебания с частотой 100 МГц на отстройках от 10 Гц до 5 МГц. Результат показан на рисунке 6 (из оригинального рисунка удалены только некоторые служебные данные прибора и изменена его цветовая палитра).

Разработанная программная модель генератора колебания с фазовыми шумами позволяет сымитировать колебание со спектральной плотностью мощности фазовых шумов, подобной изображенной на рисунке 6, но в диапазоне ее частот от 2 Гц до 150 кГц. Были подобраны параметры модели (http://www.swsys.ru/uploaded/image/2023-4/11.jpg).

Построенный график (http://www.swsys.ru/uploaded/image/2023-4/12.jpg) демонстрирует спектральную плотность мощности колебания, сгенерированного программной моделью, при этом выполнено усреднение по 2 000 реализаций колебаний с фазовым шумом. Точками обозначены значения, указанные цифрами в верхней части рисунка 6. Очевидно хорошее совпадение результатов моделирования с ними.

График зависимости усредненных значений частоты от времени (http://www.swsys.ru/uploaded/image/2023-4/13.jpg), рассчитанный по мгновенным значениям частоты при времени усреднения 100 мс, иллюстрирует, что эти значения частоты флуктуируют в пределах ±0,001 Гц ($\pm 1 \cdot 10^{-11}$ по отношению к частоте 100 МГц), что характеризует колебание в E5052B Signal Source Analyzer как высокостабильное.

Результаты теста программной модели позволяют сделать заключение о корректности ее работы и возможности использования в со-

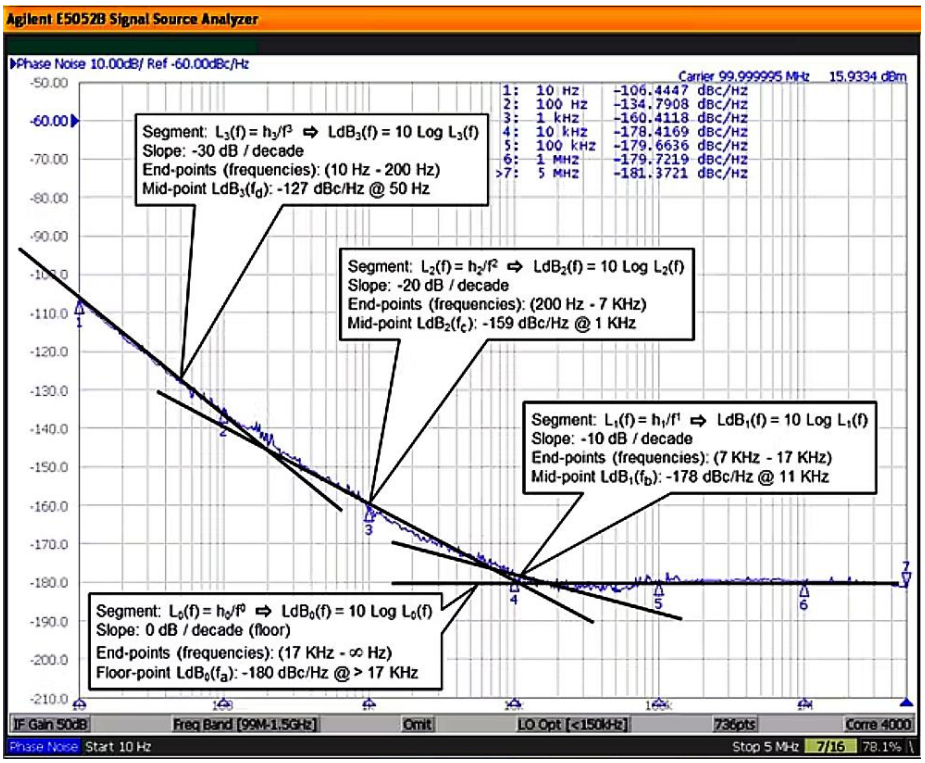


Рис. 6. Результат измерения фазовых шумов колебания в E5052B Signal Source Analyzer

Fig. 6. Result of measuring fluctuation phase noise in E5052B Signal Source Analyzer

ставе программ, моделирующих каналы передачи данных.

Заключение

Понимание и моделирование фазовых шумов крайне важны для разработки высокопроизводительных систем связи. Чтобы как можно лучше изучить влияние фазового шума, необходимо иметь возможность анализировать поведение фазового шума в различных частях системы связи. Созданная программа формирует комплексное колебание с флуктуацией фазы и рассчитывает последовательность его комплексных отсчетов для конечного количества равноотстоящих моментов времени. Разработанная программа была протестирована путем сравнения с результатами, полученными в ходе эксперимента с помощью реального прибора Keysight Technologies E5052B Signal Source Analyzer. Результаты тестирования не противоречат фундаментальным положениям теории колебаний со случайной начальной фазой типа фликкер-шум.

Список литературы

1. Aminu M.U., Lehtomaki J., Juntti M. Beamforming and transceiver optimization with phase noise for mmWave and THz bands. Proc. ISWCS, 2019, pp. 692–696. doi: 10.1109/ISWCS.2019.8877230.

2. Chen X., Fang C., Zou Y., Wolfgang A., Svensson T. Beamforming MIMO-OFDM systems in the presence of phase noises at millimeter-wave frequencies. Proc. WCNCW, 2017, pp. 1–6. doi: 10.1109/WCNCW.2017.7919036.

3. Chen X., Wang H., Fan W., Zou Y., Wolfgang A., Svensson T., Luo J. Phase noise effect on MIMO-OFDM systems with common and independent oscillators. Wireless Communications and Mobile Computing, 2017, art. 8238234. doi: 10.1155/2017/8238234.

4. Delos P. System-level LO phase noise model for phased arrays with distributed phase-locked loops. RFPD, 2018. URL: https://www.analog.com/en/technical-articles/system level-lo-phase-noise-model-for-phased-arrays-with-distributed-phase-locked-loops.html (дата обращения: 20.05.2023).

5. Качармина Е.Г., Толкачев П.А., Шахтарин Б.И. Полиномиальная модель расчета уровня шумов в синтезаторах частот. Вестн. МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2018. № 2. doi: 10.18698/0236-3933-2018-2-41-47.

6. Free Ch.E., Aitchison C.S. Oscillators. In: RF and Microwave Circuit Design: Theory and Applications, Wiley Publ., 2021, pp. 301–358. doi: 10.1002/9781119332237.ch11.

7. Fang Y., Qiu L., Liang X., Ren C. Cell-Free massive MIMO systems with oscillator phase noise: Performance analysis and power control. IEEE Transactions on Vehicular Technology, 2021, vol. 70, no. 10, pp. 10048–10064. doi: 10.1109/TVT.2021.3100862.

8. Weist F. Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 1. Microwaves & RF, 2023. URL: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-1 (дата обращения: 20.05.2023).

9. Weist F. Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 2. Microwaves & RF, 2023. URL: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-2 (дата обращения: 20.05.2023).

10. Weist F. Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 3. Microwaves & RF, 2023. URL: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-3 (дата обращения: 20.05.2023).

Software & Systems doi: 10.15827/0236-235X.142.686-694 2023, vol. 36, no. 4, pp. 686–694

Software calculating complex harmonic oscillation with phase fluctuation for wireless communication systems

Fedor A. Tsvetkov
Vitaly A. Mikhaylovsky
Vasily V. Vasilyev
Valery A. Mikhaylovsky
Anna V. Lomakina

For citation
Tsvetkov, F.A., Mikhaylovsky, Vit.A., Vasilyev, V.V., Mikhaylovsky, Val.A., Lomakina, A.V. (2023) 'Software calculating complex harmonic oscillation with phase fluctuation for wireless communication systems', *Software & Systems*, 36(4), pp. 686–694 (in Russ.). doi: 10.15827/0236-235X.142.686-694
Article info
Received: 19.09.2023 After revision: 10.10.2023 Accepted: 19.10.2023

Abstract. The paper describes a program for generating a sequence of complex harmonic oscillation time samples with phase fluctuation in the form of flicker noise with a given spectral density. One of the reasons of failures in data transmission channels are synchronization errors in a receiver caused by frequency instability (initial phase) of reference generators in both a transmitter and a receiver since signals transmitted in wireless communication systems are quite sensitive to random distortions of the phase structure of a signal received by a receiver. This leads to errors in received data and, therefore, to a decrease in a data transfer rate. Thus, it is important to solve the problem of assessing stability of modern digital data transmission channels to interference and noise, which is found out at the first stage of channel design using modeling. In a modeling program, the initial characteristic of a generated oscillation is a desired spectral density of fluctuations in its initial phase. Ie becomes a basis for calculating an amplitude-frequency response of a forming digital filter. Filter input is an uncorrelated normal pseudo-random sequence; filter output is a sequence of time samples of a fluctuating initial phase of a complex harmonic oscillation with some average frequency (in particular, equal to zero). A real and imaginary components of generator output oscillation are calculated for each of these samples. Their totality is a desired complex harmonic oscillation with phase fluctuation. The program is written in NI LabVIEW graphical programming language, which shows all work stages. To assess the quality of the developed program, the authors performed a test simulation of the oscillation of a highly stable E5052B Signal Source Analyzer generator. The subsequent numerical analysis of the obtained oscillation parameters and their comparison with the known generator parameters confirmed the correctness of the program. The proposed method for modeling a phase noise effect on signals in a communication channel can be useful for specialists in the field of research and design of various wireless communication systems, as well as in studying radio signal transmission basics within an educational process in higher educational institutions.

Keywords: phase noise, spectral power density of phase noise, software model, oscillation with phase noise, LabVIEW
Acknowledgements. The paper was supported by the Southern Federal University, project No. VnGr/23-01-RT "Research of software-defined radio systems"

References

1. Aminu, M.U., Lehtomaki, J., Juntti, M. (2019) 'Beamforming and transceiver optimization with phase noise for mmWave and THz bands', *Proc. ISWCS*, pp. 692–696. doi: 10.1109/ISWCS.2019.8877230.

2. Chen, X., Fang, C., Zou, Y., Wolfgang, A., Svensson, T. (2017) 'Beamforming MIMO-OFDM systems in the presence of phase noises at millimeter-wave frequencies', *Proc. WCNCW*, pp. 1–6. doi: 10.1109/WCNCW.2017.7919036.

3. Chen, X., Wang, H., Fan, W., Zou, Y., Wolfgang, A., Svensson, T., Luo, J. (2017) 'Phase noise effect on MIMO-OFDM systems with common and independent oscillators', *Wireless Communications and Mobile Computing*, art. 8238234. doi: 10.1155/2017/8238234.

4. Delos, P. (2018) 'System-level LO phase noise model for phased arrays with distributed phase-locked loops', *RFPD*, available at: https://www.analog.com/en/technical-articles/system level-lo-phase-noise-model-for-phased-arrays-with-distributed-phase-locked-loops.html (accessed May, 20, 2023).

5. Kacharmina, E.G., Tolkachev, P.A., Shakhtarin, B.I. (2018) 'Polynomial model for computing noise levels in frequency synthesisers', *Herald of the Bauman MSTU. Ser. Instrument Engineering*, (2). doi: 10.18698/0236-3933-2018-2-41-47.

6. Free, Ch.E., Aitchison, C.S. (2021) 'Oscillators', in *RF and Microwave Circuit Design: Theory and Applications*, Wiley Publ., pp. 301–358. doi: 10.1002/9781119332237.ch11.

7. Fang, Y., Qiu, L., Liang, X., Ren, C. (2021) 'Cell-Free massive MIMO systems with oscillator phase noise: Performance analysis and power control', *IEEE Transactions on Vehicular Technology*, 70(10), pp. 10048–10064. doi: 10.1109/TVT.2021.3100862.

8. Weist, F. (2023) 'Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 1', *Microwaves & RF*, available at: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-1 (accessed May, 20, 2023).

9. Weist, F. (2023) 'Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 2', *Microwaves & RF*, available at: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-2 (accessed May, 20, 2023).

10. Weist, F. (2023) 'Phase-noise modeling, simulation, and propagation in phase-locked loops: Part 3', *Microwaves & RF*, available at: https://www.mwrf.com/technologies/embedded/systems/article/21259311/phasenoise-modeling-simulation-and-propagation-in-phaselocked-loops-part-3 (accessed May, 20, 2023).

Авторы

Цветков Федор Алексеевич [1], к.т.н., доцент, ведущий научный сотрудник, facvetkov@sfedu.ru
Васильев Василий Васильевич [1], младший научный сотрудник, vasily@sunsdr.com
Михайловский Виталий Алексеевич [1], младший научный сотрудник, vitaly@sunsdr.com
Михайловский Валерий Алексеевич [2], директор, valera.mikhaylovsry@yandex.ru
Ломакина Анна Владимировна [2], инженер-программист, lomakinaav90@gmail.com

[1] Южный федеральный университет, г. Таганрог, 347900, Россия
[2] ООО «Ратем», г. Таганрог, 347900, Россия

Authors

Fedor A. Tsvetkov [1], Cand. of Sci. (Engineering), Associate Professor, Leading Researcher, facvetkov@sfedu.ru
Vasily V. Vasilyev[1], Junior Researcher, vasily@sunsdr.com
Vitaly A. Mikhaylovsky [1], Junior Researcher, vitaly@sunsdr.com
Valery A. Mikhaylovsky [2], Director, valera.mikhaylovsry@yandex.ru
Anna V. Lomakina [2], Engineer-Programmer, lomakinaav90@gmail.com

[1] Southern Federal University, Taganrog, 347900, Russian Federation
[2] Limited Liability Company «Ratem», Taganrog, 347900, Russian Federation

УДК 004.056.5 doi: 10.15827/0236-235X.142.695-702 2023. Т. 36. № 4. С. 695–702

О проекте межуниверситетской квантовой сети

А.П. Овсянников
Б.М. Шабанов

Ссылка для цитирования
Овсянников А.П., Шабанов Б.М. О проекте межуниверситетской квантовой сети // Программные продукты и системы. 2023. Т. 36. № 4. С. 695–702. doi: 10.15827/0236-235X.142.695-702
Информация о статье
Поступила в редакцию: 21.08.2023 После доработки: 18.09.2023 Принята к публикации: 13.10.2023

Аннотация. Работа посвящена вопросам создания межуниверситетской квантовой сети – инфраструктурного полигона для проверки решений построения сетей с квантовым распределением ключей и расширения возможностей подготовки кадров в области квантовых коммуникаций. В настоящее время интерес к созданию сетей с квантовым распределением ключей проявляют все передовые страны. В России разработано собственное оборудование квантового распределения ключей и строится магистральная квантовая сеть. Для ускорения практического внедрения квантовых коммуникаций и решения связанных с ним множества научных, технических, нормативных и кадровых вопросов разработана Концепция создания, развития и эксплуатации межуниверситетской квантовой сети национальной исследовательской компьютерной сети. В рамках реализации Концепции создан пилотный проект межуниверситетской квантовой сети, которая объединяет ряд университетов и научных организаций, имеющих собственную инфраструктуру квантовых коммуникаций, обладающих компетенциями в области квантовых технологий, ведущих исследования и разработки в этой области и осуществляющих подготовку специалистов данного направления. Описана топология межуниверситетской квантовой сети, приведены схемы распространения ключа и обмена данными. При рассмотрении различных схем распределения ключей предпочтение отдается схемам с минимальными требованиями к защите доверенных промежуточных узлов. Предложено исследовать применение в межуниверситетской сети протокола MDI квантового распределения ключей через недоверенные узлы. Приведено краткое описание российского оборудования квантового распределения ключей, планируемого для применения в межуниверситетской квантовой сети. Использование в этой сети оборудования всех трех российских производителей квантового распределения ключа позволит продемонстрировать возможности масштабирования квантовых сетей на каждом типе оборудования, исследовать возможности совместной работы оборудования разных производителей и адаптировать его к совместному использованию. Пилотный проект создания межуниверситетской квантовой сети рассчитан на реализацию в 2023–2024 гг.

Ключевые слова: квантовая сеть, квантовое распределение ключей, КРК, квантовые коммуникации, информационная безопасность, национальная исследовательская компьютерная сеть

Введение. Шифрование данных необходимо для безопасности всех современных телекоммуникаций, используемых для банковских операций, управления производственными процессами, применения беспилотных автомобилей, имплантированных медицинских устройств и т.д. Для многих широко используемых криптографических систем серьезную угрозу представляет развитие квантовых компьютеров: доказано, например, что алгоритм Шора [1, 2] на квантовом компьютере позволяет в течение нескольких часов или суток подобрать ключи шифрования для асимметричных шифров, таких, например, как схема Диффи–Хелмана. В [3] показано, что симметричные шифры остаются устойчивыми к взлому квантовым компьютером, хотя и требуют увеличения длины ключа. Однако использование симметричных шифров требует наличия у удаленных абонентов одинакового секретного ключа. Перспективным подходом к замене доверенной доставки ключей курьером является применение систем *квантового распределения ключа* (КРК). Это позволит обеспечить достаточную скорость создания ключей (хотя значительно медленнее, чем уязвимые схемы Диффи–Хелмана) и исключит человеческий фактор из процесса их распределения.

Технология КРК уже вышла из стен лабораторий, в мире активно растет число пилотных квантовых сетей, создаются инфраструктуры КРК разного масштаба. В Китае с 2017 года разрабатывается защищенная сеть передачи данных в интересах государственных и финансовых организаций. В 2021 году была создана сеть КРК, соединившая абонентов на расстоянии 4 600 км с помощью как волоконно-оптических, так и спутниковых каналов КРК [4]. К 2025 году планируется увеличить их протяженность до 35 000 км [5]. При этом наряду с волоконно-оптическими линиями связи реализуются атмосферные и спутниковые каналы распределения ключа шифрования. К сети подключено свыше 200 организаций – государственных, научных, финансовых. В Южной Корее строительство квантово-защищенной инфраструктуры ведется с 2020 года, первая очередь включала 800 км для подключения

около 50 государственных учреждений, вторая – свыше 1 000 км.

В рамках Евросоюза реализуется проект европейской квантовой инфраструктуры EuroQCI [6]. Проект предполагает переход от создания экспериментальных научных квантовых сетей к национальным квантовым инфраструктурам отдельных стран и их объединению в общеевропейскую квантовую инфраструктуру. В проекте предусматриваются использование серийного оборудования разных производителей, развитие наряду с волоконно-оптическими линиями связи спутниковых каналов КРК, разработка сценариев использования квантовых ключей как для передачи данных, так и для защиты сохраняемой информации, а также интеграции в квантовую инфраструктуру квантовых вычислителей и квантовых сенсоров. В значительной мере сеть передачи данных проекта использует инфраструктуру научно-образовательной телекоммуникационной сети GÉANT. Проект планируется завершить к 2027 году, фаза создания национальных квантовых сетей начата в 2023 году, с 2024 года начнется объединение национальных квантовых сетей в единую европейскую квантовую сеть.

Отдельные пилотные проекты по созданию сетей КРК реализованы и в США, однако программа исследований направлена на создание квантовых сетей передачи данных, в которых квантовые линии связи используются не для генерации и передачи ключей, а непосредственно для передачи данных. Это существенно более сложная задача, чем КРК, и технологии для ее решения еще далеки от готовности. В рамках программы создания квантовой инфраструктуры в США планируется квантовая сеть между национальными лабораториями.

Проекты создания сетей с КРК реализуются в России с 2019 года. В МГУ им. М.В. Ломоносова (Москва) и в ТУСУР (Томск) создана университетская квантовая сеть на оборудовании компании «ИнфоТеКС», на оборудовании компании «КуРэйт» реализована университетская сеть между МИСИС и МТУСИ. В университете ИТМО (Санкт-Петербург) создана университетская сеть на оборудовании КРК компании «СМАРС-Кванттелеком».

В соответствии с дорожной картой по развитию высокотехнологичной отрасли «Квантовые коммуникации» компания РЖД создает магистральную сеть с КРК. Создан участок между Санкт-Петербургом, Москвой и Нижним Новгородом. До конца 2023 года будут введены в строй участки Нижний Новгород–Арзамас–Казань и Москва–Тула–Воронеж–Ростов-на-Дону, общая протяженность *магистральной квантовой сети* (МКС) РЖД составит 2 500 км. В 2024 году планируется охватить сетью Краснодар, Сочи, Ульяновск, Самару, Саратов, Волгоград, Ижевск, Пермь, Уфу, Екатеринбург, Челябинск. Некоторые участки этой сети строятся на оборудовании «СМАРТС-Кванттелеком», какие-то – на оборудовании «ИнфоТеКС».

В России разработано оборудование для создания отечественных систем КРК, проведены тестовые испытания этих систем, строится МКС, идет сертификация оборудования КРК (часть линейки оборудования «ИнфоТеКС» уже сертифицирована). На повестке дня – переход к внедрению и широкому применению действующей квантовой коммуникационной инфраструктуры.

Проект межуниверситетской квантовой сети

Основой инфраструктуры КРК страны должна стать МКС РЖД. При этом задачу «последней мили» предоставления доступа пользователям к услугам шифрования квантовым ключом должны решать операторы связи (например, «Ростелеком», «МТС» и др.), создавая в городах присутствия МКС РЖД городские сети с КРК на основе собственной волоконно-оптической инфраструктуры. Таким образом, для практического применения КРК необходима мультидоменная сеть, образованная совокупностью взаимодействующих друг с другом сетей КРК: магистральной сети РЖД и городских сетей «последней мили». Конечные пользователи, которым нужен защищенный обмен информацией, могут быть клиентами разных операторов защищенной квантовой связи, в свою очередь, взаимодействующих между собой через цепочку других операторов. Полноценное функционирование мультидоменной сети с КРК требует решения множества научных, технических, нормативных, кадровых вопросов. Необходима проверка предлагаемых решений на практике, а значит, нужен инфраструктурный полигон, в качестве которого предлагается использовать *межуниверситетскую квантовую сеть* (МУКС).

Создание межуниверситетской сети с КРК предусмотрено дорожной картой по развитию высокотехнологичного направления «Квантовые коммуникации». Рабочей группой, в которую вошли представители университетов и

организаций – разработчиков систем КРК, разработана Концепция создания, развития и эксплуатации МУКС *национальной исследовательской компьютерной сети* (НИКС) на 2023–2030 гг.

Согласно Концепции, целью создания МУКС в составе НИКС является предоставление научным и образовательным организациям Российской Федерации дополнительных возможностей для выполнения исследований и разработок по приоритетным направлениям научно-технологического развития на основе квантовых технологий, а также участия в крупных российских и международных научных и образовательных проектах, базирующихся на использовании отвечающей современным требованиям квантовой сети, интегрированной в инфраструктуру НИКС.

Концепция предполагает использование МУКС для решения следующих задач:

– подготовка инженерных и научных кадров для развития и эксплуатации квантовых сетей: как сетей КРК, так и квантовых сетей передачи данных;

– проведение НИР и ОКР для разработки и совершенствования оборудования для создания квантовых сетей: как для дальнейшего развития технологии КРК, так и для создания научных и инженерных заделов для реализации квантового Интернета;

– создание возможностей для появления и развития технологических компаний-стартапов в области квантовых коммуникаций, квантовых вычислений и квантовых сенсоров;

– создание условий для реализации пилотных проектов компаниями, занимающимися внедрением квантовых коммуникаций.

Возможные направления НИР и ОКР на базе МУКС:

– взаимодействие внутригородских, корпоративных сетей и магистральных сетей с КРК;

– сценарии интеграции квантовой сети с сетью передачи данных;

– управление ключами и мониторинг квантовой сети;

– возможность использования оборудования, в том числе квантового, от разных производителей;

– увеличение дистанции и скорости распределения ключей;

– мультиплексированное распределение ключей в волоконно-оптических линиях связи;

– разработка доверенных и недоверенных узлов (квантовых репитеров);

– распределение ключей в атмосфере, между объектами в космическом пространстве и на поверхности Земли;

– разработка и испытание различных стандартов для квантовых коммуникаций в рамках комитета по стандартизации ТК-26;

– разработка моделей угроз для сетей КРК.

В соответствии с Концепцией МУКС на начальном этапе (2023–2024 гг.) будет создаваться как «надстроенная» на НИКС (сеть передачи данных) сеть КРК.

НИКС – национальная сеть науки и образования (NREN), единственная в России научно-образовательная телекоммуникационная сеть федерального масштаба, обладающая протяженной высокоскоростной магистральной инфраструктурой и международными каналами, обеспечивающими интеграцию с зарубежными NREN и Интернетом [7]. Одной из важнейших функций НИКС является предоставление своим пользователям – организациям науки и образования страны устойчивой и отвечающей современным требованиям (в том числе в части информационной безопасности [8]) телекоммуникационной сети для выполнения исследований и разработок. Как всякая сеть науки и образования, НИКС – не только сеть передачи данных, но и полигон для тестирования и распространения новых информационных технологий. Развертывание МУКС в составе НИКС и на ее основе представляется логичным шагом, обеспечивающим наибольшие возможности для ее функционирования в режиме как научной сети, так и продуктовой, обеспечивающей защиту передаваемых данных в соответствии со всеми требованиями законодательства.

В рамках реализации Концепции разработан пилотный проект МУКС. Предусматривается объединение университетов, имеющих собственную инфраструктуру квантовых коммуникаций (университетские квантовые сети, лабораторные стенды), обладающих компетенциями в области квантовых технологий, ведущих в ней исследования и разработки и осуществляющих подготовку специалистов данного направления. Участники проекта: университет ИТМО (Санкт-Петербург), МГУ им. М.В. Ломоносова, квантовая сеть МИСИС–МТУСИ, МСЦ РАН (Москва), ННГУ им. Н.И. Лобачевского (Нижний Новгород), ИКТИБ ЮФУ (Таганрог), КНИТУ КАИ им. А.Н. Туполева и КФТИ КазНЦ (Казань), Самарский университет им. С.П. Королева и ПГУТИ (Самара).

Схема МУКС в рамках пилотного проекта 2023–2024 гг. приведена на рисунке 1.

Базовой конфигурацией системы КРК является сегмент «точка-точка», ключ создается оборудованием, устанавливаемым на концах волоконно-оптической линии связи, – распределяется между связываемыми ею узлами. Длины сегментов КРК имеют практические ограничения: скорость выработки ключей зависит от оптических потерь [9, 10], необходимо отсутствие активных оптических и электрооптических компонентов, в том числе усилителей сигнала, необратимо разрушающих передаваемые квантовые состояния. Так как защищенный обмен данными требует наличия общего ключа между любой парой узлов, возникает необходимость передачи квантового ключа через несколько сегментов (промежуточных доверенных узлов) [11, 12]. Китайская национальная квантовая сеть также построена на основе доверенных промежуточных узлов [10].

Передача квантового ключа между городами реализуется с использованием МКС РЖД. В городах присутствия МКС создаются узлы МУКС на базе узлов связи НИКС в университетах и научных организациях. Узлы МУКС непосредственно подключаются к узлам МКС, остальные участники пилотного проекта подключаются к узлам МУКС. Генерация квантовых ключей осуществляется на оборудовании КРК, устанавливаемом на концах волоконно-оптических линий связи, обмен данными реализуется по каналам связи НИКС с использованием совместимых с оборудованием КРК *средств криптографической защиты информации* (СКЗИ), получающих квантовые ключи от оборудования КРК. МКС рассматривается как некий единый сегмент – закрытый от МУКС доверенный сервис передачи квантовых ключей между СКЗИ МУКС, размещаемыми в защищенных зонах доверенных узлов МКС РЖД.

Типовая схема сегмента между двумя университетами приведена на рисунке 2, между университетом и узлом МКС РЖД – на рисунке 3. Различие схем обусловлено тем, что зашифрованные квантовыми ключами данные и передаваемые конечным абонентам ключи шифрования в МУКС используют разные пути передачи.

Существенным недостатком доверенных узлов является появление ключа в открытом виде на каждом доверенном узле, что делает необходимой защиту аппаратуры от нарушителя, поскольку на доверенном узле имеются квантовые ключи от соседних сегментов сети, соединенных с данным узлом. Если для МКС РЖД требование защиты доверенных узлов выполняется безусловно, то для узлов МУКС в университетах этого хотелось бы избежать как для снижения стоимости ее создания и эксплуатации, так и для обеспечения максимальной гибкости ее использования в качестве экспериментального стенда для исследовательских работ и подготовки специалистов. Поэтому при рассмотрении различных схем распределения ключей [13] предпочтение отдается схемам с минимальными требованиями к защите доверенных промежуточных узлов. Рассматривается также применение в МУКС протокола MDI (Measurement-Device-Independent) КРК через недоверенные узлы, доказательства стойкости которого приведены в [14].

МУКС использует оборудование всех российских разработчиков оборудования КРК. Для сегментов в Санкт-Петербурге, Нижнем Новгороде, Казани, Самаре используются оборудование КРК от «СМАРТС-Кванттелеком» и СКЗИ «ФПСУ-IP» от «Амикон». Сегменты между ИКТИБ ЮФУ (Таганрог) и узлом МКС (Ростов-на-Дону), между МГУ им. М.В. Ломоносова и МСЦ РАН (Москва) строятся на оборудовании «Инфотекс». Сегмент МИСИС–МСЦ РАН использует оборудование КРК «КуРэйт» и СКЗИ «Код безопасности».

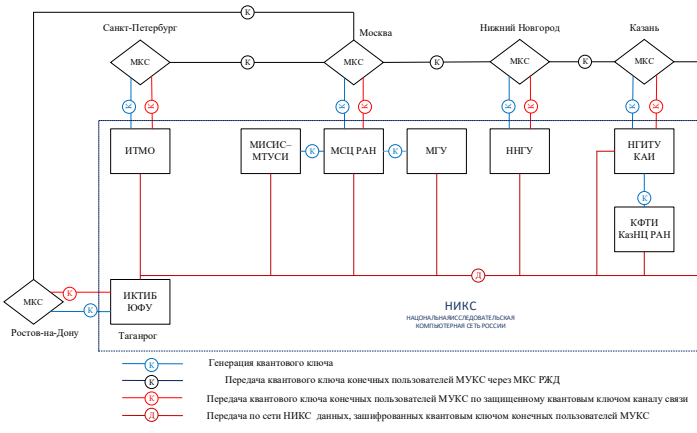


Рис. 1. Схема МУКС. Пилотный проект

Fig. 1. Interuniversity quantum network scheme. Pilot project

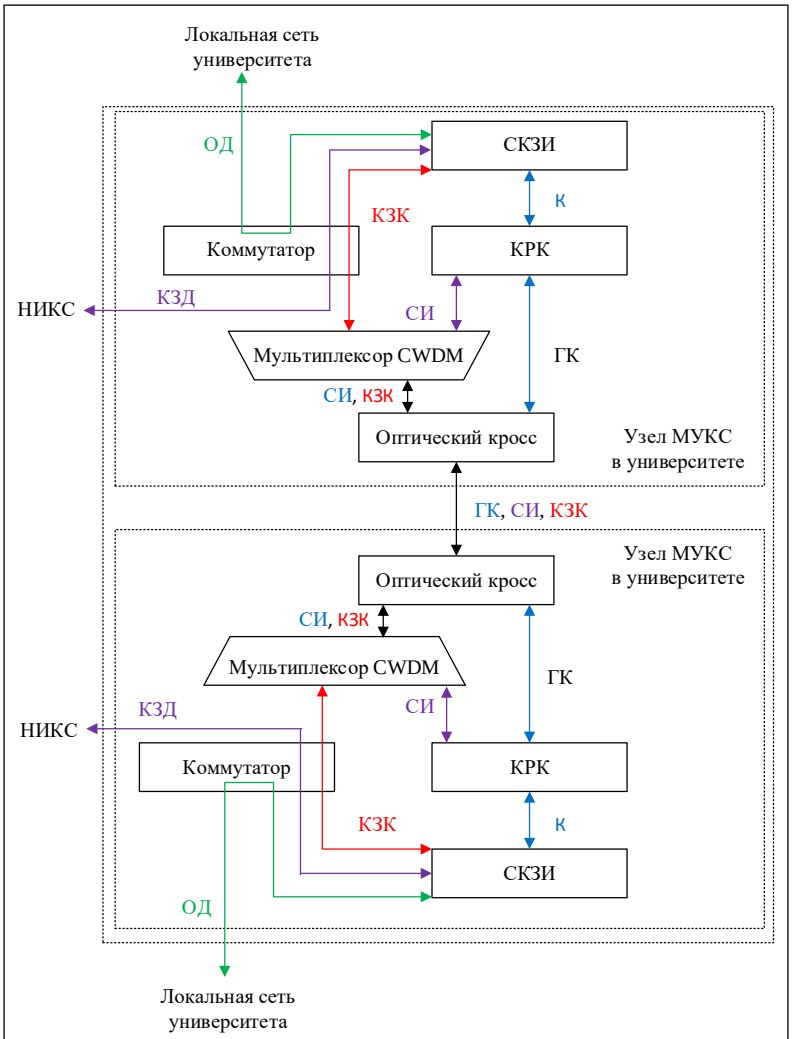


Рис. 2. Типовая схема сегмента между университетами:
ГК – генерация квантового ключа;
К – просеянный квантовый ключ; СИ – служебная информация;
КЗК – криптозащищенный ключ конечного пользователя;
КЗД – криптозащищенные данные; ОД – нешифрованные данные

Fig. 2. Typical scheme of a segment between universities:
QK – quantum key generation;
K – sifted quantum key; PI – proprietary information;
CPK – end-user crypto-protected key;
CPD – crypto-protected data; UD – unencrypted data

Оборудование «Инфотекс» использует основанный на геометрически однородных состояниях оригинальный протокол [15], разработанный в МГУ им. М.В. Ломоносова. Оно предназначено для защиты государственной критической информационной инфраструктуры государства и крупных госкомпаний, обеспечивает высокую производительность по генерации и обновлению ключа, максимальную скорость передачи. Оборудование производится серийно, более 75 % стоимости изделия приходится на российские компоненты. Ожидается завершение его сертификации.

Оборудование «СМАРТС-Квантелеком» реализует уникальный российский протокол генерации квантового ключа на боковых частотах, созданный учеными ИТМО (Санкт-Петербург) [16, 17]. Комплект предназначен для защиты критической информационной инфраструктуры крупных и средних российских компаний, обладает уникальным сочетанием производительности, возможностей и стоимости. Первое поколение этого оборудования уже применяется в крупных российских компаниях. Находится в процессе прохождения сертификации. Использование его в рамках МУКС позволит существенно поднять качество работы и его характеристики, откроет возможность защиты критической информационной инфраструктуры широкого круга российских компаний.

Оборудование «КуРэйт» реализует метод обманных состояний (decoy states) протокола BB84 [18], используемый в мировом сообществе, в том числе в зарубежном оборудовании. Его особенностями являются применение однофотонных детекторов и односторонней (поляризационной) оптической схемы, а также повышенная эффективность исправления ошибок. Оборудование проходит процедуру сертификации в испытательной лаборатории «Код Безопасности» и интегрировано с многофункциональным межсетевым экраном «Континент», что обеспечивает подключение к МУКС организаций, использующих решения компании «Код Безопасности».

Использование в МУКС оборудования всех трех российских производителей КРК позволит продемонстрировать возможности масштабирования квантовых сетей на каждом типе оборудования, исследовать возможности сов-

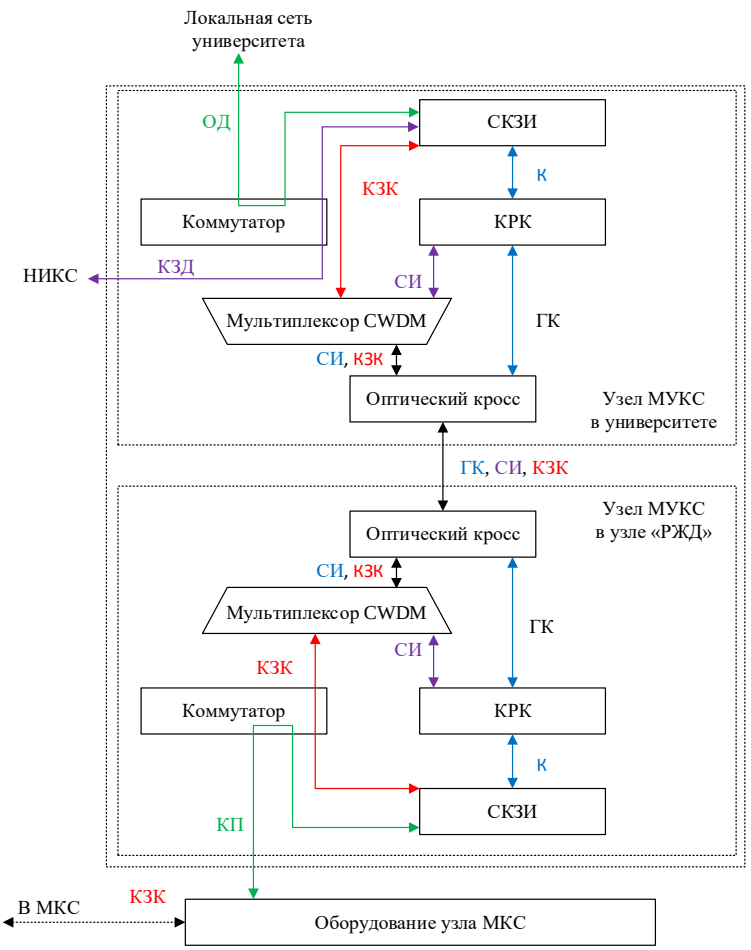


Рис. 3. Типовая схема сегмента между узлом МУКС в университете и узлом МКС: ГК – генерация квантового ключа; К – просеянный квантовый ключ; СИ – служебная информация; КЗК – криптозащищенный ключ конечного пользователя; КП – ключ конечного пользователя (внутри доверенного сегмента); КЗД – криптозащищенные данные; ОД – нешифрованные данные

Fig. 3. Typical segment diagram between an interuniversity quantum network node and a backbone quantum network node: QK – quantum key generation; K – sifted quantum key; PI – proprietary information; CPK – end-user crypto-protected key; EUK – end-user key (within a trusted segment); CPD – crypto-protected data; UD – unencrypted data

местной работы оборудования разных производителей и адаптировать его к совместному использованию в мультидоменной сети с КРК. По мере сертификации оборудования КРК может быть организован защищенный доступ с использованием квантовых сетей на оборудовании разных производителей к суперкомпьютерным ресурсам МГУ им. М.В. Ломоносова и МСЦ РАН – ведущих суперкомпьютерных центров страны.

Заключение

Описанный пилотный проект создания МУКС рассчитан на реализацию в 2023–2024 гг. В 2023 году на Форуме будущих технологий «Вычисления и связь. Квантовый мир» на базе опытного сегмента МУКС был проведен тестовый сеанс квантовой связи между МГУ им. М.В. Ломоносова и ННГУ им. Н.И. Лобачевского.

В дальнейшем планируются опытная эксплуатация пилотного сегмента МУКС с целью разработки сервисов (в рамках НИКС) по передаче информации, защищенной квантовым ключом, а также масштабирование сети, в том числе с использованием КРК через спутник. Предполагается также развитие МУКС как научной квантовой сети для повышения ее функциональности в области передачи квантовых состояний по линиям связи для объединения квантовых объектов – квантовых вычислителей и сенсоров.

Список литературы

1. Shor P.W. Algorithms for quantum computation: Discrete logarithms and factoring. Proc. 35th Annual Symposium on Foundations of Comput. Sci., 1994, pp. 124–134. doi: 10.1109/SFCS.1994.365700.
2. Shor W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. on Computing, 1997, vol. 26, no. 5, pp. 1484–1509. doi: 10.1137/S0097539795293172.
3. Bernstein D.J., Lange T. Post-quantum cryptography. Nature, 2017, no. 549, pp. 188–194. doi: 10.1038/nature23461.
4. Chen Y.A., Zhang Q., Chen T.Y. et al. An integrated space-to-ground quantum communication network over 4,600 kilometres. Nature, 2021, no. 589, pp. 214–219. doi: 10.1038/s41586-020-03093-8.

5. Qin H. Towards large-scale quantum key distribution network and its applications. Proc. ITU QIT4N Workshop, 2019. URL: https://www.itu.int/en/ITU-T/Workshops-and-Seminars/2019060507/Documents/Hao_Qin_Presentation.pdf (дата обращения: 25.08.2023).

6. The European Quantum Communication Infrastructure (EuroQCI) Initiative. URL: https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci (дата обращения: 01.09.2023).

7. Абрамов А.Г., Гончар А.А., Евсеев А.В., Шабанов Б.М. Национальная исследовательская компьютерная сеть нового поколения: текущее состояние и концепция развития // Информационные технологии. 2021. Т. 27. № 3. С. 115–124. doi: 10.17587/it.27.115-124.

8. Абрамов А.Г. Защита от DDoS-атак своими руками: оперативные разработка и внедрение сервиса в Национальной исследовательской компьютерной сети России // Программные продукты и системы. 2022. Т. 35. № 4. С. 572–582. doi: 10.15827/0236-235X.140.572-582.

9. Scarani V., Bechmann-Pasquinucci H., Cerf N.J., Dušek M., Lütkenhaus N., Peev M. The security of practical quantum key distribution. Rev. Mod. Phys., 2009, vol. 81, pp. 1301–1350. doi: 10.1103/RevModPhys.81.1301.

10. Zhang Q., Xu F., Chen Y.-A., Peng C.-Zh., Pan J.-W. Large scale quantum key distribution: Challenges and solutions [Invited]. Opt. Express, 2018, vol. 26, no. 18, pp. 24260–24273. doi: 10.1364/OE.26.024260.

11. Elliot C. Building the quantum network. New J. of Phys., 2002, vol. 4, pp. 46.1–46.12. doi: 10.1088/1367-2630/4/1/346.

12. Lucamarini M., Yuan Z.L., Dynes J.F., Shields A.J. Overcoming the rate-distance limit of quantum key distribution without quantum repeaters. Nature, 2018, no. 557, pp. 400–403. doi: 10.1038/s41586-018-0066-6.

13. Жиляев А.Е. Классификация схем выработки и распределения ключей в сетях квантового распределения ключей произвольной топологии // Докл. ТУСУР. 2021. Т. 24. № 4. С. 33–39. doi: 10.21293/1818-0442-2021-24-4-33-39.

14. Кулик С.П., Молотков С.Н. MDI – Measurement Device Independent квантового распределения ключей // Письма в ЖЭТФ. 2023. Т. 118. № 1-2. С. 62–70.

15. Молотков С.Н. О геометрически однородных когерентных состояниях в квантовой криптографии // Письма в ЖЭТФ. 2012. Т. 95. № 6. С. 361–366. doi: 10.1134/S0021364012060070.

16. Kiselev F., Goncharov R., Veselkova N., Samsonov E., Kiselev A.D., Egorov V. Performance of subcarrier-wave quantum key distribution in the presence of spontaneous Raman scattering noise generated by classical DWDM channels. JOSA B, 2021, vol. 38, no. 2, pp. 595–601. doi: 10.1364/JOSAB.412289.

17. Samsonov E., Goncharov R., Gaidash A. et al. Subcarrier wave continuous variable quantum key distribution with discrete modulation: Mathematical model and finite-key analysis. Sci. Rep., 2020, no. 10, art. 10034. doi: 10.1038/s41598-020-66948-0.

18. Трушечкин А.С., Киктенко Е.О., Кронберг Д.А., Федоров А.К. Стойкость метода обманных состояний в квантовой криптографии // УФН. 2021. Т. 191. № 1. С. 93–109. doi: 10.3367/UFNr.2020.11.038882.

Software & Systems doi: 10.15827/0236-235X.142.695-702 2023, vol. 36, no. 4, pp. 695–702

On an interuniversity quantum network project

Aleksey P. Ovsyannikov
Boris M. Shabanov

For citation
Ovsyannikov, A.P., Shabanov, B.M. (2023) 'On an interuniversity quantum network project', *Software & Systems*, 36(4), pp. 695–702 (in Russ.). doi: 10.15827/0236-235X.142.695-702
Article info
Received: 21.08.2023 After revision: 18.09.2023 Accepted: 13.10.2023

Abstract. The paper discusses the issues of creating an interuniversity quantum network, a so-called national infrastructure testbed for quantum key distribution (QKD). It is designed for testing solutions for building networks with quantum key distribution, equipment checking and benchmarking in practice, education and personnel training. Currently all advanced countries are showing interest in creating QKD networks. Russia developed its own QKD equipment and is building a backbone QKD quantum network. A Concept for creating, developing and operating an Interuniversity Quantum Network (IUQN) based on a National Research Computer Network (NICS) was developed in order to accelerate practical implementation of quantum communications and to solve a variety of related research, technical, regulatory and personnel issues. A pilot project of the IUQN was developed as a part of implementing the concept. IUQN connects universities and scientific organizations that have their own quantum communication infrastructure, have competencies in quantum technologies, conduct research and development and train specialists in this field. The paper describes IUQN topology, gives QKD and data exchange schemes. When considering various QKD schemes, minimal requirements for trusted intermediate nodes protection are preferred. The paper proposes to consider using MDI-QKD protocol through untrusted nodes. The authors briefly describe Russian QKD equipment planned to use in IUQN. The use of equipment from all three Russian manufacturers in IUQN will allow demonstrating the possibilities of scaling quantum networks on each equipment type, explore

the possibilities of different manufacturers' equipment collaboration and adapt it to joint use. It is planned to implement the pilot project of creating IUQN in 2023-2024.

Keywords: quantum key distribution, QKD, quantum communications, information security, national research and educational network

Acknowledgements. The work was carried out at the JSC RAS in terms of FNEF-2022-0014 state assignment

Благодарности. Работа выполнена в МСЦ РАН в рамках государственного задания по теме FNEF- 2022-0014

References

1. Shor, P.W. (1994) 'Algorithms for quantum computation: Discrete logarithms and factoring', *Proc. 35th Annual Symposium on Foundations of Comput. Sci.*, pp. 124–134. doi: 10.1109/SFCS.1994.365700.

2. Shor, W. (1997) 'Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer', *SIAM J. on Computing*, 26(5), pp. 1484–1509. doi: 10.1137/S0097539795293172.

3. Bernstein, D.J., Lange, T. (2017) 'Post-quantum cryptography', *Nature*, (549), pp. 188–194. doi: 10.1038/nature23461.

4. Chen, Y.A., Zhang, Q., Chen, T.Y. et al. (2021) 'An integrated space-to-ground quantum communication network over 4,600 kilometres', *Nature*, (589), pp. 214–219. doi: 10.1038/s41586-020-03093-8.

5. Qin, H. (2019) 'Towards large-scale quantum key distribution network and its applications', *Proc. ITU QIT4N Workshop*, available at: https://www.itu.int/en/ITU-T/Workshops-and-Seminars/2019060507/Documents/Hao_Qin_Presentation.pdf (accessed August 25, 2023).

6. *The European Quantum Communication Infrastructure (EuroQCI) Initiative*, available at: https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci (accessed September 01, 2023).

7. Abramov, A.G., Gonchar, A.A., Evseev, A.V., Shabanov, B.M. (2021) 'The new generation national research computer network: Current status and concept for the development', *Information Technologies*, 27(3), pp. 115–124 (in Russ.). doi: 10.17587/it.27.115-124.

8. Abramov, A.G. (2022) 'DIY DDoS Protection: Operational development and implementation of the service in the National Research Computer Network of Russia', *Software & Systems*, 35(4), pp. 572–582 (in Russ.). doi: 10.15827/0236-235X.140.572-582

9. Scarani, V., Bechmann-Pasquinucci, H., Cerf, N.J., Dušek, M., Lütkenhaus, N., Peev, M. (2009) 'The security of practical quantum key distribution', *Rev. Mod. Phys*., 81, pp. 1301–1350. doi: 10.1103/RevModPhys.81.1301.

10. Zhang, Q., Xu, F., Chen, Y.-A., Peng, C.-Zh., Pan, J.-W. (2018) 'Large scale quantum key distribution: Challenges and solutions [Invited]', *Opt. Express*, 26(18), pp. 24260–24273. doi: 10.1364/OE.26.024260.

11. Elliot, C. (2002) 'Building the quantum network', *New J. of Phys*., 4, pp. 46.1–46.12. doi: 10.1088/1367-2630/4/1/346.

12. Lucamarini, M., Yuan, Z.L., Dynes, J.F., Shields, A.J. (2018) 'Overcoming the rate-distance limit of quantum key distribution without quantum repeaters', *Nature*, (557), pp. 400–403. doi: 10.1038/s41586-018-0066-6.

13. Zhilyaev, A.E. (2021) 'Key generation and distribution schemes classification for quantum key distribution networks of arbitrary topology', *Proc. of TUSUR University*, 24(4), pp. 33–39 (in Russ.). doi: 10.21293/1818-0442-2021-24-4-33-39.

14. Kulik, S.P., Molotkov, S.N. (2023) 'MDI – Measurement Device Independent quantum key distribution', *JETP Letters*, 118(1-2), pp. 62–70 (in Russ.).

15. Molotkov, S.N. (2012) 'On geometrically uniform states in quantum cryptography', *JETP Letters*, 95(6), pp. 361–366. doi: 10.1134/S0021364012060070 (in Russ.).

16. Kiselev, F., Goncharov, R., Veselkova, N., Samsonov, E., Kiselev, A.D., Egorov, V. (2021) 'Performance of subcarrier-wave quantum key distribution in the presence of spontaneous Raman scattering noise generated by classical DWDM channels', *JOSA B*, 38(2), pp. 595–601. doi: 10.1364/JOSAB.412289.

17. Samsonov, E., Goncharov, R., Gaidash, A. et al. (2020) 'Subcarrier wave continuous variable quantum key distribution with discrete modulation: Mathematical model and finite-key analysis', *Sci. Rep*., (10), art. 10034. doi: 10.1038/s41598-020-66948-0.

18. Trushechkin, A.S., Kiktenko, E.O., Kronberg, D.A., Fedorov, A.K. (2021) 'Security of the decoy state method for quantum key distribution', *Physics-Uspekhi*, 191(1), pp. 93–109 (in Russ.). doi: 10.3367/UFNr.2020.11.038882.

Авторы

Овсянников Алексей Павлович [1],
ведущий научный сотрудник,
ovsyannikov@jscc.ru

Шабанов Борис Михайлович [1],
д.т.н., чл.-корр. РАН, директор,
jscc@jscc.ru

[1] Межведомственный суперкомпьютерный центр РАН, г. Москва, 119991, Россия

Authors

Aleksey P. Ovsyannikov [1],
Leading Researcher,
ovsyannikov@jscc.ru

Boris M. Shabanov [1], Dr.Sc. (Engineering),
Corresponding Member of the RAS,
Director, jscc@jscc.ru

[1] Joint Supercomputer Center of RAS,
Moscow, 119991, Russian Federation

С О Д Е Р Ж А Н И Е

Бибило П.Н., Кириенко Н.А., Романов В.И.
Выделение из многоуровневого представления системы булевых функций подсистем для совместной логической минимизации 509
Ulyanov, S.V., Ulyanov, V.S.
Quantum algorithmic gate-based computing: Grover quantum search algorithm design in quantum software engineering 523
Иванов А.В., Матвеев Ю.Н.
Поиск решения задачи целочисленного программирования с помощью итеративного округления координат 539
Достовалова А.М., Шерминская А.А.
Автоматическая локализация ошибок времени выполнения в программном обеспечении с помощью нейронных сетей 551
Рыбаков А.А., Швиндт А.Н.
Создание инструментария для векторизации тела плоского цикла с помощью векторных инструкций AVX-512 561
Гурьев В.В., Шабалина О.А., Садовникова Н.П., Воронина А.А.
Разработка модели жизненного цикла мобильных приложений для людей с ограниченными интеллектуальными возможностями 573
Страшнов Е.В., Мироненко И.Н.
Моделирование динамики гидроприводов в системах виртуального окружения 582
Тумбинская М.В., Галиев Р.А.
Идентификация фейк-новостей с помощью веб-ресурса на основе нейронных сетей 590
Еремеев А.П., Сергеев М.Д., Петров В.С.
Интеграция методов обучения с подкреплением и нечеткой логики для интеллектуальных систем реального времени 600
Ай Мин Тайк, Лупин С.А., Федяшин Д.А.
Использование библиотеки MPI для параллельной реализации алгоритма полного перебора вариантов 607
Баранов А.В., Корепанов П.М., Кузнецов Е.Е.
Обеспечение информационной безопасности научного суперкомпьютерного центра 615
Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Брызгалов А.А.
Разработка моделей производственных и бизнес-процессов сетевых предприятий на основе многоагентных систем 632
Абу-Абед Ф.Н., Жиронкин С.А.
Моделирование структуры импортозамещения на базе модели системы оптимального распределения 644
Горбич Л.Г.
Поиск оптимального набора букв для стилевой классификации художественных текстов методом статистических индексов 654
Читалов Д.И.
Разработка программного модуля для решателя stressFoam платформы OpenFOAM 661
Лабынцев А.В., Игнатьев В.В., Васильев В.В., Михайловский Вит.А., Харланов Д.В.
Проектирование фильтра из шпилечных резонаторов с металлизированными отверстиями с помощью программной системы 668
Попов Ю.Г., Малов Г.С.
Построение параметризованной 3D-модели рабочей зоны шнека экструдера 678
Цветков Ф.А., Васильев В.В., Михайловский Вит.А., Михайловский Вал.А., Ломакина А.В.
Программный расчет комплексного гармонического колебания с флуктуацией фазы для беспроводных систем связи 686
Овсянников А.П., Шабанов Б.М.
О проекте межуниверситетской квантовой сети 695

C O N T E N T

Bibilo, P.N., Kirienko, N.A., Romanov, V.I.
Extracting subsystems from a multilevel representation of a Boolean function system for joint logical minimization 509

Ulyanov, S.V., Ulyanov, V.S.
Quantum algorithmic gate-based computing: Grover quantum search algorithm design in quantum software engineering 523

Ivanov, A.V., Matveev, Yu.N.
An integer programming solution: Iterative rounding of coordinates 539

Dostovalova, A.M., Sherminskaya, A.A.
Automatic runtime error localization for software debugging using neural networks 551

Rybakov, A.A., Shvindt, A.N.
Tools for vectorizing a flat loop body using AVX-512 vector instructions 561

Guryev, V.V., Shabalina, O.A., Sadovnikova, N.P., Voronina, A.A.
Developing a mobile application life cycle model for people with intellectual disabilities 573

Strashnov, E.V., Mironenko, I.N.
Simulation of hydraulic actuator dynamics in virtual environment systems 582

Tumbinskaya, M.V., Galiev, R.A.
Neural network-based approach for identifying fake news 590

Eremeev, A.P., Sergeev, M.D., Petrov, V.S.
Integration of reinforcement learning methods and fuzzy logic for intelligent real-time systems 600

Aye Min Thike, Lupin, S.A., Fedyashin, D.A.
Using MPI library for parallel implementation of a brute-force algorithm 607

Baranov, A.V., Korepanov, P.M., Kuznetsov, E.E.
Information security of a supercomputer center 615

Telnov, Yu.F., Kazakov, V.A., Danilov, A.V., Bryzgalov, A.A.
Network enterprises: Production and business process models based on multi-agent systems 632

Abu-Abed, F.N., Zhironkin, S.A.
Russian import-substitution structure based on an optimal distribution system model 644

Gorbich, L.G.
Finding an optimal letter set for stylistic classification of literary texts by a statistical index method 654

Chitalov, D.I.
Software module for stressFoam solver in OpenFOAM platform 661

Labyntsev, A.V., Ignatyev, V.V., Vasilyev, V.V., Mikhaylovsky, Vit.A., Kharlanov, D.V.
Designing a filter from metallized-hole hairpin resonators using a software system 668

Popov, Yu.G., Malov, G.S.
Constructing a parametric 3D model of an extruder screw working area 678

Tsvetkov, F.A., Mikhaylovsky, Vit.A., Vasilyev, V.V., Mikhaylovsky, Val.A., Lomakina, A.V.
Software calculating complex harmonic oscillation with phase fluctuation for wireless communication systems 686

Ovsyannikov, A.P., Shabanov, B.M.
On an interuniversity quantum network project 695

Вниманию подписчиков

Научный журнал
«Программные продукты и системы»
распространяется по подписке
через подписное агентство «УП УРАЛ-ПРЕСС»
Индекс издания
в электронном каталоге агентства 70799

Телефон редакции: (4822) 39-91-49
Электронная почта: red@cps.tver.ru
Сайт журнала: www.swsys.ru