

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ИНФОРМАТИКА

INFORMATICS

TOM **19**
VOL.

2 | **2022**

ОТ РЕДАКЦИИ

В журнале «Информатика» публикуются оригинальные и обзорные статьи, описывающие результаты фундаментальных и прикладных исследований специалистов академического и вузовского профиля в области информатики и информационных технологий.

Основной целью журнала является публикация наиболее значимых новых результатов в указанной области. Приветствуются статьи, описывающие заключительные результаты научных проектов и диссертационных исследований, открывающие новые направления исследований, которые находятся на стыке информатики и других наук.

Журнал рассчитан на широкий круг специалистов в области информатики и информационных технологий.

Основные разделы журнала:

- биоинформатика;
- математическое моделирование;
- защита информации и надежность систем;
- информационные технологии;
- логическое проектирование;
- обработка сигналов, изображений, речи, текста и распознавание образов;
- автоматизация проектирования;
- интеллектуальные системы.

Префикс DOI: 10.37661

Условия распространения материалов:

контент доступен под лицензией Creative Commons Attribution 4.0 License

Индексирование:

Высшей аттестационной комиссией Республики Беларусь журнал «Информатика» был включен в список научных изданий для опубликования результатов диссертационных исследований.

В декабре 2017 г. включен в базу данных Российского индекса научного цитирования (РИНЦ). С помощью инструментов и сервисов, доступных на платформе eLIBRARY (раздел «Личный кабинет»), можно самостоятельно корректировать список своих публикаций и цитирований в РИНЦ.

В июле 2017 г. включен в базу журналов открытого доступа Directory of Open Access Journals (DOAJ).

С помощью поисковых систем Google Scholar, WorldCat, Соционет можно получить свободный доступ к полному тексту научных публикаций журнала.

Адрес редакции:

ул. Сурганова, 6, к. 305, г. Минск, 220012, Беларусь
Тел. +375 (017) 351 26 22

Editorial address:

Surganova str., 6, of. 305, Minsk, 220012, Belarus
Phone +375 (017) 351 26 22

E-mail: rio@newman.bas-net.by

<https://inf.grid.by/jour>

THE EDITOR'S NOTE

The journal «Informatics» is a scientific publication in computer sciences and information technologies which reviews the results in basic and applied research of scientists from the universities and scientific centers.

The journal focuses on the most significant and modern papers of research projects results and PhD/DSc thesis in computer sciences.

The journal is edited for the specialists in IT and computer sciences research and application.

The main sections of the journal:

- bioinformatics;
- mathematical modeling;
- information protection and system reliability;
- information technology;
- logical design;
- signal, image, speech, text processing and pattern recognition;
- computer-aided design;
- artificial intelligence methods.

DOI Prefix: 10.37661

Distribution:

content is distributed under Creative Commons Attribution 4.0 License

Indexation:

the journal «Informatics» is in the list of scientific publications recommended by the Higher Attestation Commission of the Republic of Belarus for scientists to publish the results of PhD/DSc research.

In December 2017 the journal was included in the database of the Russian Science Citation Index (RISC) and provides free access to reviewed electronic scientific paper, improving scientific information traffic and also raising quotation of works of the authors (please use <https://elibrary.ru> or section https://elibrary.ru_author_tools).

In July 2017 included in the database of open access journals Directory of Open Access Journals (DOAJ).

Using the Google Scholar, WorldCat, Соционет search engine, you can get free access to full text of scientific publications of magazine.

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

Informatika

Том 19, № 2, апрель-июнь 2022

Ежеквартальный научный журнал

Издается с января 2004 г.

Учредитель и издатель – Объединенный институт проблем информатики
Национальной академии наук Беларуси (ОИПИ НАН Беларуси)

Г л а в н ы й р е д а к т о р

Тузиков Александр Васильевич, д-р физ.-мат. наук, проф., чл.-корр. НАН Беларуси,
ОИПИ НАН Беларуси (Минск, Беларусь)

З а м е с т и т е л ь г л а в н о г о р е д а к т о р а

Ковалев Михаил Яковлевич, д-р физ.-мат. наук, проф., чл.-корр. НАН Беларуси,
ОИПИ НАН Беларуси (Минск, Беларусь)

Р е д а к ц и о н н а я к о л л е г и я

Абламейко Сергей Владимирович, д-р техн. наук, проф., академик НАН Беларуси, БГУ (Минск, Беларусь)

Анищенко Владимир Викторович, канд. техн. наук, доцент, ООО «СофтКлуб» (Минск, Беларусь)

Бибило Петр Николаевич, д-р техн. наук, проф., ОИПИ НАН Беларуси (Минск, Беларусь)

Бобов Михаил Никитич, д-р техн. наук, проф., БГУИР (Минск, Беларусь)

Долгий Александр Борисович, д-р техн. наук, проф., Высшая инженерная школа Бретани (Нант, Франция)

Дудин Александр Николаевич, д-р физ.-мат. наук, проф., БГУ (Минск, Беларусь)

Карпов Алексей Анатольевич, д-р техн. наук, доцент, СПИИРАН (Санкт-Петербург, Россия)

Килин Сергей Яковлевич, д-р физ.-мат. наук, проф., академик НАН Беларуси, Президиум НАН
Беларуси (Минск, Беларусь)

Краснопрошин Виктор Владимирович, д-р техн. наук, проф., БГУ (Минск, Беларусь)

Крот Александр Михайлович, д-р техн. наук, проф., ОИПИ НАН Беларуси (Минск, Беларусь)

Кругликов Сергей Владимирович, д-р воен. наук, канд. техн. наук, доцент, ОИПИ НАН Беларуси
(Минск, Беларусь)

Лиходед Николай Александрович, д-р физ.-мат. наук, проф., БГУ (Минск, Беларусь)

Матус Петр Павлович, д-р физ.-мат. наук, проф., Институт математики НАН Беларуси (Минск, Беларусь)

Скляров Валерий Анатольевич, д-р техн. наук, проф., Университет Авейру (Авейру, Португалия)

Сотсков Юрий Назарович, д-р физ.-мат. наук, проф., ОИПИ НАН Беларуси (Минск, Беларусь)

Стемпковский Александр Леонидович, д-р техн. наук, проф., академик РАН, ИПИМ РАН (Москва, Россия)

Харин Юрий Семенович, д-р физ.-мат. наук, проф., академик НАН Беларуси, НИИ ППМИ БГУ
(Минск, Беларусь)

Чернявский Александр Федорович, д-р техн. наук, проф., академик НАН Беларуси, НИИ ПФП
им. А. Н. Севченко БГУ (Минск, Беларусь)

Ярмолик Вячеслав Николаевич, д-р техн. наук, проф., БГУИР (Минск, Беларусь)

Редакционный совет

Ефанов Дмитрий Викторович, Российский университет транспорта (Московский институт инженеров транспорта) (Москва, Россия)

Кумари Мадху, Университетский центр исследований и разработок, Университет Чандигарха (Мохали, Пенджаб, Индия)

Лазарев Александр Алексеевич, Институт проблем управления им. В. А. Трапезникова РАН (Москва, Россия)

Лай Цунг-Чьян, Азиатский университет в Тайчжуне (Китайская Народная Республика, Тайвань)

Марина Нинослав, Университет информационных наук и технологий им. Св. апостола Павла (Охрид, Македония)

Меликян Вазген Шаваршович, Национальный политехнический университет Армении (Ереван, Армения)

Пеш Эрвин, Зигенский университет (Зиген, Германия)

Сингх Таджиндер, Институт инженерии и технологий Сант Лонговал (Лонговал, Пенджаб, Индия)

Ходаченко Максим Леонидович, Институт космических исследований Австрийской академии наук (Грац, Австрия)

Чиулла Карло, Университет Эпока (Тирана, Албания)

Штейнберг Борис Яковлевич, Институт математики, механики и компьютерных наук Южного федерального университета (Ростов-на-Дону, Россия)

ИНФОРМАТИКА

Том 19, № 2, апрель-июнь 2022

Ответственный за выпуск *Мойсейчик Светлана Сергеевна*

Редактор *Гончаренко Галина Борисовна*

Корректор *Михайлова Анна Антоновна*

Компьютерная верстка *Бутевич Ольга Борисовна*

Сдано в набор 31.05.2022. Подписано в печать 20.06.2022. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография. Усл. печ. л. 13,5. Уч.-изд. л. 13,2. Тираж 40 экз. Заказ 2.

Государственное научное учреждение «Объединенный институт проблем информатики
Национальной академии наук Беларуси».

Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/274 от 04.04.2014. ЛП № 02330/444 от 18.12.13. Ул. Сурганова, 6, 220012, Минск, Беларусь.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

THE UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

INFORMATICS

Vol. 19, no. 2, April-June 2022

Published quarterly

Issued since January 2004

Founder and publisher – the United Institute of Informatics Problems
of the National Academy of Sciences of Belarus (UIIP NASB)

Editor-in-Chief

Alexander V. Tuzikov, D. Sc. (Phys.-Math.), Prof., Corr. Member of NASB,
UIIP NASB (Minsk, Belarus)

Deputy Editor-in-Chief

Mikhail Y. Kovalyov, D. Sc. (Phys.-Math.), Prof., Corr. Member of NASB,
UIIP NASB (Minsk, Belarus)

Editorial Board

Sergey V. Ablameyko, D. Sc. (Eng.), Prof., Academician of NASB, BSU (Minsk, Belarus)

Uladimir V. Anishchanka, Ph. D. (Eng.), Assoc. Prof., SoftClub Ltd. (Minsk, Belarus)

Petr N. Bibilo, D. Sc. (Eng.), Prof., UIIP NASB (Minsk, Belarus)

Mikhail N. Bobov, D. Sc. (Eng.), Prof., BSUIR (Minsk, Belarus)

Alexandre B. Dolgui, D. Sc. (Eng.), Prof., IMT Atlantique (Nantes, France)

Alexander N. Dudin, D. Sc. (Phys.-Math.), Prof., BSU (Minsk, Belarus)

Alexey A. Karpov, D. Sc. (Eng.), Assoc. Prof., SPII RAS (Saint Petersburg, Russia)

Sergey Ya. Kilin, D. Sc. (Phys.-Math.), Prof., Academician of NASB, Presidium of NASB (Minsk, Belarus)

Viktor V. Krasnoproshin, D. Sc. (Eng.), Prof., BSU (Minsk, Belarus)

Alexander M. Krot, D. Sc. (Eng.), Prof., UIIP NASB (Minsk, Belarus)

Sergey V. Kruglikov, D. Sc. (Milit.), Ph. D. (Eng.), Assoc. Prof., UIIP NASB (Minsk, Belarus)

Nikolai A. Likhoded, D. Sc. (Phys.-Math.), Prof., BSU (Minsk, Belarus)

Petr P. Matus, D. Sc. (Phys.-Math.), Prof., Institute of Mathematics of NASB (Minsk, Belarus)

Valery A. Sklyarov, D. Sc. (Eng.), Prof., University of Aveiro (Aveiro, Portugal)

Yuri N. Sotskov, D. Sc. (Phys.-Math.), Prof., UIIP NASB (Minsk, Belarus)

Alexander L. Stempkovsky, D. Sc. (Eng.), Prof., Academician of RAS, IPPM RAS (Moscow, Russia)

Yuriy S. Kharin, D. Sc. (Phys.-Math.), Prof., Academician of NASB, RI APMI BSU (Minsk, Belarus)

Alexander F. Cherniavsky, D. Sc. (Eng.), Prof., Academician of NASB, A. N. Sevchenko IAPP BSU (Minsk, Belarus)

Vyacheslav N. Yarmolik, D. Sc. (Eng.), Prof., BSUIR (Minsk, Belarus)

Editorial Council

Dmitry V. Efanov, Russian University of Transport (Moscow Institute of Transport Engineers) (Moscow, Russia)

Madhu Kumari, University Center for Research & Development, Chandigarh University (Mohali, Punjab, India)

Alexander A. Lazarev, V. A. Trapeznikov Institute of Control Sciences of the RAS (Moscow, Russia)

Tsung-Chyan Lai, Asia University at Taichung (The People's Republic of China, Taiwan)

Ninoslav Marina, St. Paul the Apostle University of Information Sciences and Technology (Ohrid, Macedonia)

Vazgen Sh. Melikyan, National Polytechnic University of Armenia (Yerevan, Armenia)

Erwin Pesch, University of Siegen (Siegen, Germany)

Tajinder Singh, Sant Longowal Institute of Engineering & Technology (Longowal, Punjab, India)

Maxim L. Khodachenko, Space Research Institute, Austrian Academy of Sciences (Graz, Austria)

Carlo Ciulla, Epoka University (Tirana, Albania)

Boris Steinberg, Institute of Mathematics, Mechanics and Computer Science Southern Federal University (Rostov-on-Don, Russia)

INFORMATICS

Vol. 19, no. 2, April-June 2022

Issue Head *Sviatlana S. Maiseichyk*

Editor *Halina B. Hancharenka*

Corrector *Hanna A. Mikhailava*

Computer Imposition *Volha B. Butsevich*

Sent for press 31.05.2022. Output 20.06.2022. Format 60×84 1/8. Offset paper. Headset Times. Riesography. Printed sheets 13,5. Publisher's signatures 13,2. Circulation 40 copies. Order 2.

State Scientific Institution "The United Institute of Informatics Problems of the National Academy of Sciences of Belarus".

Certificate on the state registration of the publisher, manufacturer, distributor of printing editions no. 1/274 dated 04.04.2014. License for the press no. 02330/444 dated 18.12.13.

6, Surganov Str., 220012, Minsk, Belarus.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

СОДЕРЖАНИЕ

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ, РЕЧИ, ТЕКСТА И РАСПОЗНАВАНИЕ ОБРАЗОВ

Рудиков С. И., Цветков В. Ю., Шкадаревич А. П. Уменьшение динамического диапазона инфракрасных изображений на основе блочно-приоритетного выравнивания и сжатия гистограмм..... 7

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Бибило П. Н., Логинова И. П. Экспериментальное сравнение эффективности программ минимизации систем булевых функций в классе дизъюнктивных нормальных форм26

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Клименок В. И. Система массового обслуживания с разделением процессора, повторными вызовами и нетерпеливостью запросов56

Розин Б. М., Шатерник И. А. Об оптимизации смешанной зарядной инфраструктуры электробусов для городских маршрутов 68

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Бухаров М. Н. Информационная система для онтологического моделирования предметных областей85

Андрушевич А. А., Войтешенко И. С., Емельянова О. Ю. О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов 100

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

CONTENTS

SIGNAL, IMAGE, SPEECH, TEXT PROCESSING AND PATTERN RECOGNITION

Rudikov S. I., Tsviatkou V. Yu., Shkadarevich A. P. Reducing the dynamic range of infrared images based on block-priority equalization and compression of histograms 7

LOGICAL DESIGN

Bibilo P. N., Loginova I. P. Experimental comparison of the effectiveness of programs for minimizing systems of Boolean functions in the class of disjunctive normal forms 26

MATHEMATICAL MODELING

Klimenok V. I. A retrial queueing system with processor sharing and impatient customers 56

Rozin B. M., Shaternik I. A. On optimization of the mixed charging infrastructure of electric buses for urban routes 68

INFORMATION TECHNOLOGIES

Bukharov M. N. Information system for ontological modelling the subject areas 85

Andrushevich A. A., Vojteshenko I. S., Emelianova O. Ju. About simulation environment for modeling the Internet of Things components and applications using visual flow-based programming tools and cloud services 100

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ, РЕЧИ, ТЕКСТА И РАСПОЗНАВАНИЕ ОБРАЗОВ

SIGNAL, IMAGE, SPEECH, TEXT PROCESSING AND PATTERN RECOGNITION



УДК 004.932
<https://doi.org/10.37661/1816-0301-2022-19-2-7-25>

Оригинальная статья
Original Paper

Уменьшение динамического диапазона инфракрасных изображений на основе блочно-приоритетного выравнивания и сжатия гистограмм

С. И. Рудиков^{1✉}, В. Ю. Цветков², А. П. Шкадаревич¹

¹Научно-технический центр «ЛЭМТ» БелОМО,
ул. Макаенка, 23, корп.1, Минск, 220114, Беларусь
✉E-mail: stanislav.rudikov@gmail.com

²Белорусский государственный университет
информатики и радиоэлектроники,
ул. П. Бровки, 6, Минск, 220013, Беларусь

Аннотация

Цели. Рассматривается задача уменьшения динамического диапазона инфракрасных изображений для их воспроизведения на устройствах отображения с узким динамическим диапазоном. Исследуется метод адаптивного выравнивания гистограммы изображения на основе интегральной функции распределения яркости. Для преобразования яркости пиксела этот метод использует аппроксимацию локальных значений выравнивания ближайших блоков пикселей, на которые делится исходное изображение. Это повышает локальный контраст изображения, но приводит к высокой вычислительной сложности, которая растет с уменьшением размера блока. Целью работы является снижение вычислительной сложности адаптивного выравнивания и сжатия гистограмм инфракрасных изображений при уменьшении их динамического диапазона.

Методы. Используются методы обработки изображений.

Результаты. Для уменьшения вычислительной сложности преобразования динамического диапазона инфракрасных изображений предложена блочно-приоритетная модификация метода адаптивного выравнивания гистограммы. Модификация основана на разделении множества блоков изображения на подмножества высокоприоритетных и низкоприоритетных блоков в зависимости от их яркостных статистических свойств. При интерполяции значений пикселей для высокоприоритетных блоков используются локальные значения выравнивания, а для низкоприоритетных блоков – общие значения выравнивания. В результате общее число векторов выравнивания сокращается пропорционально соотношению размеров подмножеств и снижается вычислительная сложность преобразования динамического диапазона.

Заключение. При изменении отношения количества высокоприоритетных блоков пикселей инфракрасного изображения к количеству всех блоков в диапазоне 0,25–0,75 предложенный алгоритм более эффективен по сравнению с алгоритмами глобального и адаптивного выравнивания гистограммы.

Ключевые слова: динамический диапазон, инфракрасные изображения, выравнивание гистограммы, сжатие гистограммы, адаптивное выравнивание

Для цитирования. Рудиков, С. И. Уменьшение динамического диапазона инфракрасных изображений на основе блочно-приоритетного выравнивания и сжатия гистограмм / С. И. Рудиков, В. Ю. Цветков, А. П. Шкадаревич // Информатика. – 2022. – Т. 19, № 2. – С. 7–25.

<https://doi.org/10.37661/1816-0301-2022-19-2-7-25>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 29.03.2022

Подписана в печать | Accepted 11.04.2022

Опубликована | Published 29.06.2022

Reducing the dynamic range of infrared images based on block-priority equalization and compression of histograms

Stanislav I. Rudikov^{1✉}, Viktor Yu. Tsviatkou², Alexey P. Shkadarevich¹

¹Scientific and Technical Center LEMT of the BelOMO,
st. Makayonok 23/1, Minsk, 220114, Belarus

✉E-mail: stanislav.rudikov@gmail.com

²Belarusian State University of Informatics and Radioelectronics,
st. P. Brovki, 6, Minsk, 220013, Belarus

Abstract

Objectives. The problem of reducing the dynamic range of infrared images for their reproduction on display devices with a narrow dynamic range is considered. The method of local image histogram equalization based on the integral distribution function of brightness is investigated. To transform the brightness of a pixel, this method uses an approximation of the local alignment values of the nearest blocks of pixels of original image. This increases the local contrast of the image, but leads to high computational complexity, which is increasing while block size decreases. The aim of the work is to reduce the computational complexity of adaptive equalization and compression of infrared image histograms while reducing their dynamic range.

Methods. Image processing methods are used.

Results. To reduce the computational complexity of transforming the dynamic range of infrared images, a block-priority modification of the adaptive histogram equalization method is proposed. The modification is based on the division of the set of image blocks into two subsets of high-priority and low-priority blocks depending on their brightness statistical properties. When interpolating pixel values, high-priority blocks use local alignment values, and low-priority blocks use global alignment values. As a result, the total number of alignment vectors is reduced in proportion to the ratio of subsets sizes and the computational complexity of the dynamic range transformation is reduced.

Conclusion. When changing the ratio of the number of high-priority blocks of infrared image pixels to the number of all blocks in the range of 0.25–0.75, the proposed algorithm is more efficient than global and adaptive histogram equalization algorithms.

Keywords: dynamic range, infrared images, histogram equalization, histogram compression, adaptive equalization

For citation. Rudikov S. I., Tsviatkou V. Yu., Shkadarevich A. P. *Reducing the dynamic range of infrared images based on block-priority equalization and compression of histograms*. Informatika [Informatics], 2022, vol. 19, no. 2, pp. 7–25 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-2-7-25>

Conflict of interest. The authors declare of no conflict of interest.

Введение. Для воспроизведения инфракрасных (ИК) изображений, имеющих широкий динамический диапазон (от 0 до $2^{16}-1$ уровней яркости), на стандартных устройствах отображения с более узким динамическим диапазоном (как правило, от 0 до 2^8-1 уровней яркости) осуществляется сжатие динамического диапазона ИК-изображений. Такое преобразование

выполняется также для медицинских, спутниковых оптических и радарных изображений и называется тоновым отображением [1–3]. Исследования в данной области направлены на минимизацию потерь информации в результате сжатия динамического диапазона и уменьшение вычислительной сложности преобразования. Низкую вычислительную сложность и высокую равномерность гистограммы результирующего изображения обеспечивают глобальные преобразования. Хорошо сохраняя семантику исходного изображения, они, однако, теряют локальный контраст и мелкие детали. Локальные преобразования имеют более высокую вычислительную сложность и не позволяют выровнять гистограмму всего изображения, но повышают локальный контраст и разборчивость мелких деталей. Большинство методов тонового отображения являются гибридными и используют нелинейное квантование [3–6], выравнивание и ограничение гистограммы [6–8], представление изображений в нескольких разрешениях с помощью гауссовой и билатеральной фильтрации, карт прозрачности [5, 8–13], вейвлет-преобразование [12], семантическую сегментацию [14], кластеризацию [15], линейную интерполяцию [13], локальные экстремумы [11], вероятностные модели [16], модели человеческого восприятия [17], искусственные нейронные сети [18].

Для ИК-изображений, применяемых в системах оптического наведения, важно сохранение детализации и локального контраста при сжатии динамического диапазона, а также улучшение видимости в темных и светлых областях даже за счет нарушения семантики восприятия. Дополнительным требованием к преобразованию в таких системах является низкая вычислительная сложность. Поэтому методы тонового отображения ИК-изображений используют преимущественно выравнивание гистограммы [19, 20], представление изображений в нескольких разрешениях с помощью гауссовой и билатеральной фильтрации [19–22] и нелинейное квантование [22], имеющие относительно невысокую вычислительную сложность и уменьшающие потери в локальном контрасте при сжатии динамического диапазона. Высокий локальный контраст и сохранение детализации в темных и светлых областях обеспечивает метод адаптивного выравнивания гистограммы (Adaptive Histogram Equalization, АНЕ) [23]. Как и метод глобального выравнивания гистограммы (Histogram Equalization, HE) [24], метод АНЕ применяет интегральную функцию распределения яркостей для формирования значений выравнивания. Отличие этих методов заключается в том, что для преобразования яркости пиксела метод HE использует глобальное значение выравнивания, формируемое на основе яркости всего изображения, а метод АНЕ – аппроксимацию локальных значений выравнивания ближайших блоков пикселей, на которые делится исходное изображение. Это позволяет повысить локальный контраст изображения, но приводит к высокой вычислительной сложности, которая растет с уменьшением размера блока и, соответственно, увеличением количества блоков. Данный недостаток характерен для большинства модификаций метода АНЕ.

Целью исследования является снижение вычислительной сложности адаптивного выравнивания и сжатия гистограмм ИК-изображений при уменьшении их динамического диапазона.

Постановка задачи. В алгоритме HE глобального выравнивания гистограммы исходное изображение $I_{\text{HDR}} = \|i_{\text{HDR}}(y, x)\|_{(y=0, \overline{Y-1}, x=0, \overline{X-1})}$ размером $Y \times X$ пикселей с динамическим диапазоном $[0, L_{\text{HDR}} - 1]$ (L_{HDR} – количество уровней квантования исходного изображения I_{HDR}) преобразуется в изображение $I_{\text{HE}}(L_{\text{LDR}}) = \|i_{\text{HE}}(L_{\text{LDR}}, y, x)\|_{(y=0, \overline{Y-1}, x=0, \overline{X-1})}$ с динамическим диапазоном $[0, L_{\text{LDR}} - 1]$ (L_{LDR} – количество уровней квантования преобразованного изображения $I_{\text{HE}}(L_{\text{LDR}})$) в результате выборки значений вектора $H_{\text{E}}(L_{\text{LDR}}) = \|h_{\text{E}}(L_{\text{LDR}}, l)\|_{(l=0, \overline{L_{\text{HDR}}-1})}$ выравнивания по значениям пикселей исходного изображения I_{HDR} (рис. 1) с помощью выражения

$$i_{\text{HE}}(L_{\text{LDR}}, y, x) = h_{\text{E}}(L_{\text{LDR}}, i_{\text{HDR}}(y, x)) \quad (1)$$

при $y = \overline{0, Y-1}, x = \overline{0, X-1}$.

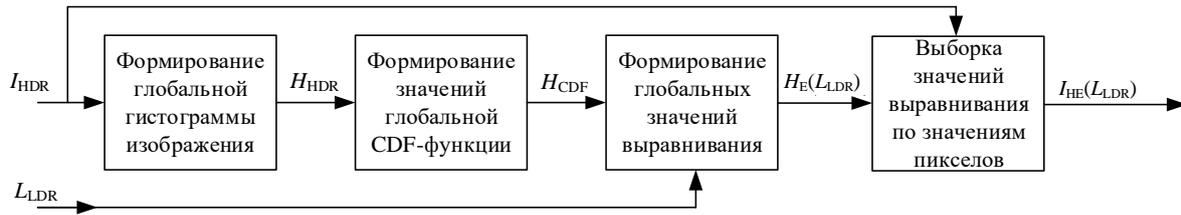


Рис. 1. Преобразование динамического диапазона изображения на основе глобального выравнивания гистограммы HE

Fig. 1. Image dynamic range converting based on global histogram equalization HE

Значения вектора $H_E(L_{LDR})$ выравнивания вычисляются на основе вектора $H_{CDF} = \|h_{CDF}(l)\|_{(l=0, L_{HDR}-1)}$ значений интегральной функции распределения яркостей (Scattering Distribution Function, CDF) с помощью уравнения

$$h_E(L_{LDR}, l) = \left\lfloor \frac{(L_{LDR} - 1) \cdot (h_{CDF}(l) - \min(H_{CDF}))}{(YX - \min(H_{CDF}))} \right\rfloor \quad (2)$$

при $l = \overline{0, L_{HDR} - 1}$, где $h_{CDF}(0) = h_{HDR}(0)$; $h_{CDF}(l+1) = h_{HDR}(l+1) + h_{CDF}(l)$ при $l = \overline{0, L_{HDR} - 2}$; $h_{HDR}(l)$ – элемент гистограммы $H_{HDR} = \|h_{HDR}(l)\|_{(l=0, L_{HDR}-1)}$ исходного изображения I_{HDR} ; $\lfloor \rfloor$ – операция округления до целого с недостатком.

Для уменьшения динамического диапазона исходного изображения I_{HDR} количество уровней квантования L_{HDR} и L_{LDR} ограничено условием $L_{HDR} > L_{LDR}$.

Алгоритм HE имеет значительно меньшую вычислительную сложность по сравнению с другими алгоритмами выравнивания гистограммы.

Временная сложность C_T^{HE} алгоритма HE определяется количеством операций обращения к памяти при загрузке исходного изображения I_{HDR} (YX операций); при формировании глобальной гистограммы H_{HDR} (YX операций), значений глобальной интегральной функции H_{CDF} распределения яркостей (L_{HDR} операций), глобальных значений $H_E(L_{LDR})$ выравнивания (L_{HDR} операций); при выборке значений выравнивания (YX операций) и выгрузке преобразованного изображения $I_{HE}(L_{LDR})$ (YX операций) с помощью выражения

$$C_T^{HE} = 4YX + 2L_{HDR}. \quad (3)$$

Пространственная сложность C_S^{HE} алгоритма HE определяется объемом оперативной памяти, необходимой для загрузки исходного изображения I_{HDR} ($YX \log_2(L_{HDR})$ бит); для формирования глобальной гистограммы H_{HDR} ($L_{HDR} \log_2(YX)$ бит), значений глобальной интегральной функции H_{CDF} распределения яркостей ($L_{HDR} \log_2(YX)$ бит), глобальных значений $H_E(L_{LDR})$ выравнивания ($L_{HDR} \log_2(YX)$ бит); для выборки значений выравнивания и формирования преобразованного изображения $I_{HE}(L_{LDR})$ ($YX \log_2(L_{LDR})$ бит), с помощью равенства

$$C_S^{HE} = 3L_{HDR} \log_2(YX) + YX (\log_2(L_{HDR}) + \log_2(L_{LDR})). \quad (4)$$

Выравнивание гистограммы с помощью алгоритма HE позволяет получить достаточно близкие вероятности яркостей пикселей после преобразования, но не учитывает локальные особенности распределения яркостей пикселей, что приводит к низкому локальному контрасту преобразованного изображения $I_{HE}(L_{LDR})$ (рис. 2).

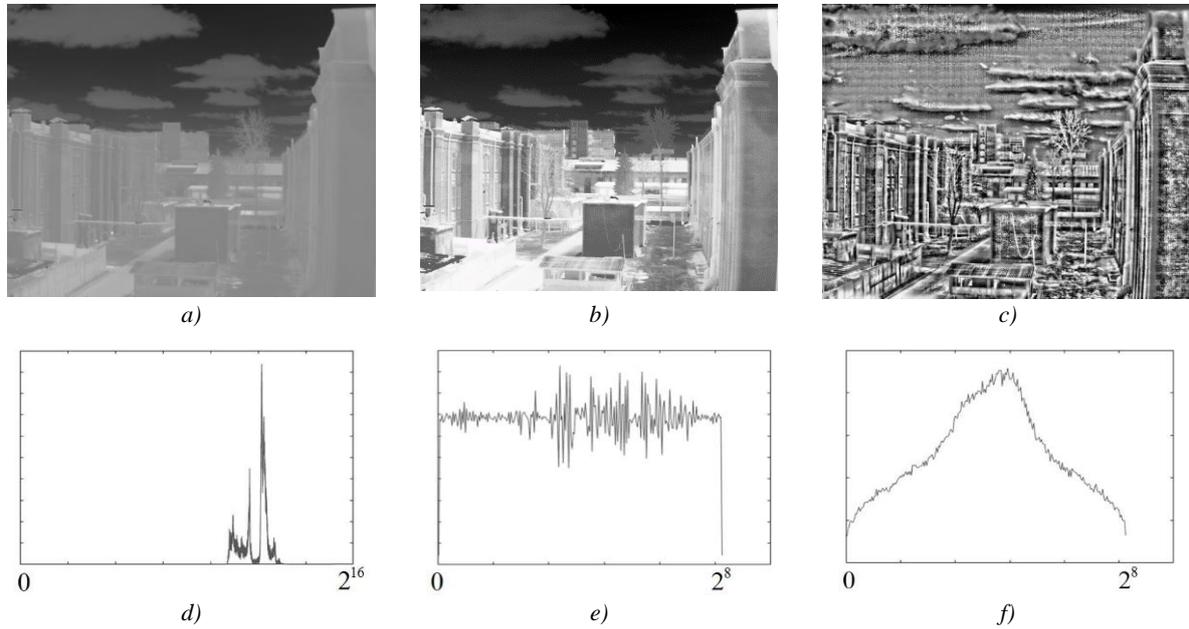


Рис. 2. ИК-изображения и их гистограммы: *a)* исходное изображение I_{HDR} ; *b)* преобразованное изображение $I_{HE}(L_{LDR})$; *c)* преобразованное изображение $I_{AHE}(L_{LDR})$ при $Y_B = 16$; *d)* гистограмма изображения I_{HDR} ; *e)* гистограмма изображения $I_{HE}(L_{LDR})$; *f)* гистограмма изображения $I_{AHE}(L_{LDR})$

Fig. 2. IR-images and histograms: *a)* original image I_{HDR} ; *b)* modified image $I_{HE}(L_{LDR})$; *c)* modified image $I_{AHE}(L_{LDR})$ for $Y_B = 16$; *d)* image histogram I_{HDR} ; *e)* image histogram $I_{HE}(L_{LDR})$; *f)* image histogram $I_{AHE}(L_{LDR})$

В алгоритме АНЕ адаптивного (блочного) выравнивания гистограммы исходное изображение I_{HDR} делится на $B_Y \times B_X$ блоков по вертикали и горизонтали размером $Y_B \times X_B$ пикселей по вертикали и горизонтали (рис. 3), для центров которых с помощью выражения (2) формируются векторы $H_E(y_B, x_B, L_{LDR}) = \|h_E(y_B, x_B, L_{LDR}, l)\|_{(l=0, L_{LDR}-1)}$ выравнивания в пределах блока, где y_B, x_B – координаты блока, $y_B = \overline{0, B_Y - 1}$, $x_B = \overline{0, B_X - 1}$. Затем на основе одного, двух или четырех (в зависимости от положения пиксела на изображении) значений векторов выравнивания смежных блоков интерполируются выравненные значения $i_{AHE}(L_{LDR}, y, x)$ пикселей изображения $I_{AHE}(L_{LDR})$ с динамическим диапазоном $[0, L_{LDR} - 1]$, что позволяет снизить вычислительную сложность преобразования. В центральной части изображения выравненные значения пикселей интерполируются на основе векторов выравнивания четырех смежных блоков с помощью выражения

$$i_{AHE}(L_{LDR}, y, x) = c_{00}h_E(y_B, x_B, L_{LDR}, i_{HDR}(y, x)) + c_{10}h_E(y_B + 1, x_B, L_{LDR}, i_{HDR}(y, x)) + c_{01}h_E(y_B, x_B + 1, L_{LDR}, i_{HDR}(y, x)) + c_{11}h_E(y_B + 1, x_B + 1, L_{LDR}, i_{HDR}(y, x)) \quad (5)$$

при $y = \overline{y_B Y_B, (y_B + 1) Y_B - 1}$, $x = \overline{x_B X_B, (x_B + 1) X_B - 1}$, $y_B = \overline{1, B_Y - 2}$, $x_B = \overline{1, B_X - 2}$, где $c_{00}, c_{10}, c_{01}, c_{11}$ – весовые коэффициенты ($c_{00} + c_{10} + c_{01} + c_{11} = 1$), определяющие вклад значения выравнивания каждого блока в выравненное значение пиксела $i_{AHE}(L_{LDR}, y, x)$.

Разделение изображения на блоки в АНЕ позволяет адаптироваться к структуре изображения и повысить локальный контраст, хотя не обеспечивает равномерность глобальной гистограммы (см. рис. 2, *c, f*).

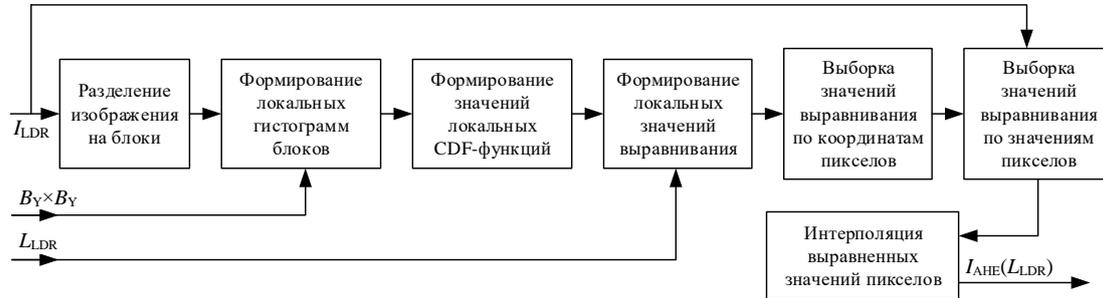


Рис. 3. Преобразование динамического диапазона изображения на основе адаптивного выравнивания гистограммы АНЕ

Fig. 3. Image dynamic range converting based on adaptive histogram equalization AHE

Для оценки эффективности алгоритмов тонового отображения широко используются показатели статистической естественности (Statistical Naturalness, SN) [25], структурной точности (Structural Fidelity, SF) [26], качества тональной карты (Tone Mapped image Quality Index, TMQI) [27]. Для изображения на рис. 2, *a* алгоритм HE обеспечивает $SN = 0,0514$, $SF = 0,7873$, $TMQI = 0,7693$. Алгоритм АНЕ позволяет улучшить показатели SN и TMQI. Для изображения на рис. 2, *a* алгоритм АНЕ обеспечивает $SN = 0,6134$, $SF = 0,6243$, $TMQI = 0,8347$ при размере блока обработки 16×16 пикселей и $SN = 0,6988$, $SF = 0,4689$, $TMQI = 0,7903$ при размере блока обработки 10×10 пикселей, т. е. алгоритм АНЕ, повышая локальный контраст и сохраняя детализацию в темных и светлых областях, выигрывает в естественности и интегральной оценке качества по сравнению с алгоритмом HE, проигрывает ему в структурной точности. Показатель TMQI является интегральным, поскольку учитывает SN и SF. Поэтому при уменьшении размера блока в АНЕ показатель TMQI уменьшается из-за более резкого уменьшения SF по сравнению с ростом SN. При акцентировании внимания на детализации более однозначную оценку качества тонового отображения ИК-изображений дают контраст D_{ST} , средний градиент G_A и энтропия E_1 . Такие оценки могут быть получены усреднением по достаточно большим блокам, например 64×64 пикселей. Для изображения на рис. 2, *a* алгоритм HE обеспечивает $D_{ST} = 32,85$, $G_A = 4,64$, $E_1 = 6,16$. Алгоритм АНЕ обеспечивает $D_{ST} = 58,29$, $G_A = 19,33$, $E_1 = 7,70$ при размере блока обработки 16×16 пикселей и $D_{ST} = 59,10$, $G_A = 24,09$, $E_1 = 7,75$ при размере блока обработки 10×10 пикселей.

Качество тонового отображения ИК-изображений с помощью алгоритма АНЕ, оцениваемое локальным контрастом и детализацией в темных и светлых областях, возрастает при уменьшении размера блока. Для систем оптического наведения достаточно высокое качество обеспечивается при размерах блоков в диапазоне $8 \times 8 - 16 \times 16$ пикселей. При этом существенно увеличивается вычислительная сложность по сравнению с алгоритмом HE.

Временная сложность C_T^{AHE} алгоритма АНЕ определяется количеством операций обращения к памяти при загрузке исходного изображения I_{HDR} (YX операций); формировании локальных гистограмм $H_{HDR}(y_B, x_B)$ (YX операций), значений локальных интегральных функций $H_{CDF}(y_B, x_B)$ распределения яркостей ($L_{HDR} B_Y B_X$ операций), локальных векторов $H_E(y_B, x_B, L_{LDR})$ выравнивания ($L_{HDR} B_Y B_X$ операций); при выборке значений выравнивания (YX операций); интерполяции выравненных значений пикселей (YX операций) и выгрузке преобразованного изображения $I_{AHE}(L_{LDR})$ (YX операций) с помощью выражения

$$C_T^{AHE} = 5YX + 2L_{HDR} B_Y B_X. \quad (6)$$

Пространственная сложность $C_S^{\text{АНЕ}}$ алгоритма АНЕ определяется объемом оперативной памяти, необходимой для загрузки исходного изображения I_{HDR} ($YX \log_2(L_{\text{HDR}})$ бит); формирования локальных гистограмм $H_{\text{HDR}}(y_B, x_B)$ ($L_{\text{HDR}} \log_2(Y_B X_B)$ бит), локальных интегральных функций $H_{\text{CDF}}(y_B, x_B)$ распределения яркостей ($L_{\text{HDR}} \log_2(Y_B X_B)$ бит), локальных значений $H_E(y_B, x_B, L_{\text{LDR}})$ выравнивания ($L_{\text{HDR}} B_Y B_X \log_2(Y_B X_B)$ бит); выборки значений выравнивания; интерполяции значений пикселей и формирования преобразованного изображения $I_{\text{АНЕ}}(L_{\text{LDR}})$ ($YX \log_2(L_{\text{LDR}})$ бит), с помощью выражения

$$C_S^{\text{АНЕ}} = L_{\text{HDR}} (2 + B_Y B_X) \log_2(Y_B X_B) + YX (\log_2(L_{\text{HDR}}) + \log_2(L_{\text{LDR}})). \quad (7)$$

В табл. 1 приведены значения вычислительной сложности алгоритмов НЕ и АНЕ для ИК-изображений размером 480×640 пикселей при $L_{\text{HDR}} = 2^{16}$, $L_{\text{LDR}} = 2^8$ и $Y_B = \{10, 16\}$.

Таблица 1

Вычислительная сложность алгоритмов НЕ и АНЕ в зависимости от размера блока по вертикали Y_B

Table 1

Computational complexity of HE and AHE algorithms in dependence with vertical block size Y_B

Y_B	Временная сложность алгоритмов <i>Temporary algorithm complexity</i>		Пространственная сложность алгоритмов <i>Spatial algorithm complexity</i>	
	АНЕ	НЕ	АНЕ	НЕ
10	672 624 640	1 359 872	2 237 551 831	10 956 732
16	213 872 640		857 767 936	

Как видно из табл. 1, вычислительная сложность алгоритма АНЕ значительно выше вычислительной сложности алгоритма НЕ, что обусловлено необходимостью хранения значений векторов $H_E(y_B, x_B, L_{\text{LDR}})$ выравнивания для $B_Y B_X$ блоков. При размере блока 16×16 пикселей алгоритм АНЕ проигрывает алгоритму НЕ во временной сложности в 157 раз, в пространственной сложности – в 78 раз. С уменьшением размера блока вычислительная сложность алгоритма АНЕ быстро растет, в то время как вычислительная сложность алгоритма НЕ не меняется. При размере блока 10×10 пикселей алгоритм АНЕ проигрывает алгоритму НЕ во временной сложности в 495 раз, а в пространственной сложности – в 204 раза. При такой высокой вычислительной сложности реализация алгоритма АНЕ в реальном времени возможна только на графическом процессоре или ПЛИС. В связи с этим актуальной является задача разработки алгоритма уменьшения динамического диапазона ИК-изображений на основе локального выравнивания и сжатия гистограмм, обеспечивающего более высокий локальный контраст после преобразования по сравнению с алгоритмом НЕ, но имеющего меньшую вычислительную сложность по сравнению с алгоритмом АНЕ.

Математическая модель, структура и алгоритм блочно-приоритетного выравнивания и сжатия гистограмм ИК-изображений. Для повышения эффективности преобразования динамического диапазона ИК-изображений предлагается блочно-приоритетное выравнивание и сжатие гистограмм (Block-Priority Histogram Equalization, ВРНЕ), основанное на разделении множества всех блоков изображения на два подмножества высокоприоритетных и низкоприоритетных блоков в зависимости от их яркостных статистических свойств и использовании при интерполяции значений пикселей формируемого изображения с узким динамическим диапазоном локальных векторов $H_E(y_B, x_B, L_{\text{LDR}})$ выравнивания для $B_Y B_X K_B$ высокоприоритетных блоков и общего вектора $H_E(L_{\text{LDR}})$ выравнивания для $B_Y B_X (1 - K_B)$ низкоприоритетных блоков, где K_B – коэффициент, определяющий часть блоков, для которых реализуется локальное выравнивание гистограмм. В результате общее число векторов выравнивания сокращается

и снижается вычислительная сложность преобразования динамического диапазона пропорционально отношению количества высокоприоритетных блоков к количеству всех блоков.

При блочно-приоритетном выравнивании и сжатии гистограмм осуществляется упорядочивание $B_Y \times B_X$ блоков исходного изображения I_{HDR} , имеющих размер $Y_B \times X_B$ пикселей по вертикали и горизонтали, по убыванию приоритета. В качестве показателя приоритета могут применяться контраст (стандартное отклонение) $D_{\text{ST}}(y_B, x_B)$, энтропия $E_1(y_B, x_B)$ и другие оценки качества. При использовании контраста или энтропии в качестве высокоприоритетных определяются первые $B_Y B_X K_B$ низкоконтрастных (низкоэнтропийных) блоков. Остальные $B_Y B_X (1 - K_B)$ высококонтрастных (высокоэнтропийных) блоков считаются низкоприоритетными. Для вычисления контраста $D_{\text{ST}}(y_B, x_B)$ и энтропии $E_1(y_B, x_B)$ блоков используются равенства

$$D_{\text{ST}}(y_B, x_B) = \sqrt{\frac{1}{Y_B X_B} \sum_{y=y_B Y_B}^{(y_B+1)Y_B-1} \sum_{x=x_B X_B}^{(x_B+1)X_B-1} (i_{\text{HDR}}(y, x) - V_M(y_B, x_B))^2}, \quad (8)$$

$$E_1(y_B, x_B) = - \sum_{l=0}^{L_{\text{HDR}}-1} P(i_{\text{HDR}}(l)) \log_2 P(i_{\text{HDR}}(l)), \quad (9)$$

где $V_M(y_B, x_B)$ – средняя яркость блока, $V_M(y_B, x_B) = \frac{1}{Y_B X_B} \sum_{y=y_B Y_B}^{(y_B+1)Y_B-1} \sum_{x=x_B X_B}^{(x_B+1)X_B-1} i_{\text{HDR}}(y, x)$;

$P(i_{\text{HDR}}(l))$ – вероятность пиксела со значением l .

Для упорядочивания блоков по убыванию приоритета формируются множества $Y_B^{\text{D}} = \left\| y_B^{\text{D}}(i) \right\|_{(i=0, \overline{B_Y B_X - 1})}$ и $X_B^{\text{D}} = \left\| x_B^{\text{D}}(i) \right\|_{(i=0, \overline{B_Y B_X - 1})}$ координат блоков, для элементов которых справедливы условия

$$(j > i) \Rightarrow \left(D_{\text{ST}}(y_B^{\text{D}}(i), x_B^{\text{D}}(i)) \geq D_{\text{ST}}(y_B^{\text{D}}(j), x_B^{\text{D}}(j)) \right), \quad (10)$$

$$(j > i) \Rightarrow \left(E_1(y_B^{\text{D}}(i), x_B^{\text{D}}(i)) \geq E_1(y_B^{\text{D}}(j), x_B^{\text{D}}(j)) \right) \quad (11)$$

при $i = \overline{0, B_Y B_X - 1}$, $j = \overline{0, B_Y B_X - 1}$.

Множество координат $\{Y_B^{\text{D}}, X_B^{\text{D}}\}$ блоков разделяется на два подмножества высокоприоритетных $\left\{ Y_B^{\text{DL}} = \left\| y_B^{\text{DL}}(i) \right\|_{(i=0, \overline{[B_Y B_X K_B] - 1})}, X_B^{\text{DL}} = \left\| x_B^{\text{DL}}(i) \right\|_{(i=0, \overline{[B_Y B_X K_B] - 1})} \right\}$ и низкоприоритетных $\left\{ Y_B^{\text{DH}} = \left\| y_B^{\text{DH}}(i) \right\|_{(i=0, \overline{B_Y B_X - [B_Y B_X K_B] - 1})}, X_B^{\text{DH}} = \left\| x_B^{\text{DH}}(i) \right\|_{(i=0, \overline{B_Y B_X - [B_Y B_X K_B] - 1})} \right\}$ блоков с помощью выражений

$$(i < [B_Y B_X K_B]) \Rightarrow \left((y_B^{\text{DL}}(i) \leftarrow y_B^{\text{D}}(i)) \wedge (x_B^{\text{DL}}(i) \leftarrow x_B^{\text{D}}(i)) \right), \quad (12)$$

$$(i \geq [B_Y B_X K_B]) \Rightarrow \left((y_B^{\text{DH}}(i - [B_Y B_X K_B]) \leftarrow y_B^{\text{D}}(i)) \wedge (x_B^{\text{DH}}(i - [B_Y B_X K_B]) \leftarrow x_B^{\text{D}}(i)) \right) \quad (13)$$

при $i = \overline{0, B_Y B_X - 1}$, где $[]$ – операция округления до ближайшего целого.

Принадлежность блоков подмножествам $\{Y_B^{\text{DL}}, X_B^{\text{DL}}\}$ и $\{Y_B^{\text{DH}}, X_B^{\text{DH}}\}$ отражается также посредством маски $M_{\text{GB}} = \|m_{\text{GB}}(y_B, x_B)\|_{(y_B=\overline{0, Y_B-1}, x_B=\overline{0, X_B-1})}$ типов блоков, каждый элемент $m_{\text{GB}}(y_B, x_B)$ которой соответствует блоку (y_B, x_B) . Если этот блок высокоприоритетный, то $m_{\text{GB}}(y_B, x_B) = 1$; если блок низкоприоритетный, то $m_{\text{GB}}(y_B, x_B) = 0$. Значения элементов маски M_{GB} вычисляются по формуле

$$m_{\text{GB}}(y_B, x_B) = \begin{cases} 1, & (y_B = y_B^{\text{D}}(i)) \wedge (x_B = x_B^{\text{D}}(i)) \wedge (i < [B_Y B_X K_B]), \\ 0, & (y_B = y_B^{\text{D}}(i)) \wedge (x_B = x_B^{\text{D}}(i)) \wedge (i \geq [B_Y B_X K_B]) \end{cases} \quad (14)$$

при $i = \overline{0, B_Y B_X - 1}$.

Для высокоприоритетных блоков на основе соответствующих векторов $H_{\text{CDF}}(y_B, x_B) = \|h_{\text{CDF}}(y_B, x_B, l)\|_{(l=\overline{0, L_{\text{HDR}}-1})}$ значений интегральных функций распределения яркостей формируются векторы $H_{\text{E}}(y_B, x_B, L_{\text{LDR}})$ выравнивания, значения элементов которых вычисляются с помощью выражения, аналогичного выражению (2):

$$h_{\text{E}}(y_B, x_B, L_{\text{LDR}}, l) = \left\lfloor \frac{(L_{\text{LDR}} - 1) \cdot (h_{\text{CDF}}(y_B, x_B, l) - \min(H_{\text{CDF}}(y_B, x_B)))}{(Y_B X_B - \min(H_{\text{CDF}}(y_B, x_B)))} \right\rfloor \quad (15)$$

при $y_B = \overline{0, B_Y - 1}$, $x_B = \overline{0, B_X - 1}$, $m_{\text{GB}}(y_B, x_B) = 1$, $l = \overline{0, L_{\text{HDR}} - 1}$, где $h_{\text{CDF}}(y_B, x_B, 0) = h_{\text{HDR}}(y_B, x_B, 0)$; $h_{\text{CDF}}(y_B, x_B, l+1) = h_{\text{HDR}}(y_B, x_B, l+1) + h_{\text{CDF}}(y_B, x_B, l)$ при $l = \overline{0, L_{\text{HDR}} - 2}$; $h_{\text{HDR}}(y_B, x_B, l)$ – элемент гистограммы $H_{\text{HDR}}(y_B, x_B) = \|h_{\text{HDR}}(y_B, x_B, l)\|_{(l=\overline{0, L_{\text{HDR}}-1})}$ блока (y_B, x_B) .

Для низкоприоритетных блоков на основе их общего вектора $H_{\text{CDF}} = \|h_{\text{CDF}}(l)\|_{(l=\overline{0, L_{\text{HDR}}-1})}$ значений интегральной функции распределения яркостей формируется общий вектор $H_{\text{E}}(L_{\text{LDR}})$ выравнивания с помощью равенства

$$h_{\text{E}}(L_{\text{LDR}}, l) = \left\lfloor \frac{(L_{\text{LDR}} - 1) \cdot (h_{\text{CDF}}(l) - \min(H_{\text{CDF}}))}{(Y_B X_B (B_Y B_X - [B_Y B_X K_B]) - \min(H_{\text{CDF}}))} \right\rfloor \quad (16)$$

при $l = \overline{0, L_{\text{HDR}} - 1}$, где $h_{\text{CDF}}(0) = h_{\text{HDR}}^{\text{G}}(0)$; $h_{\text{CDF}}(l+1) = h_{\text{HDR}}^{\text{G}}(l+1) + h_{\text{CDF}}(l)$ при $l = \overline{0, L_{\text{HDR}} - 2}$; $h_{\text{HDR}}^{\text{G}}(l)$ – элемент общей гистограммы $H_{\text{HDR}}^{\text{G}} = \|h_{\text{HDR}}^{\text{G}}(l)\|_{(l=\overline{0, L_{\text{HDR}}-1})}$ всех низкоприоритетных блоков.

Гистограмма H_{HDR}^G может быть сформирована на основе гистограммы H_{HDR} исходного изображения и гистограмм $H_{\text{HDR}}(y_B, x_B)$ высокоприоритетных блоков с помощью выражения

$$H_{\text{HDR}}^G = H_{\text{HDR}} - \sum_{y_B=0}^{Y_B-1} \sum_{x_B=0}^{X_B-1} H_{\text{HDR}}(y_B, x_B). \quad (17)$$

При блочно-приоритетном выравнивании и сжатии гистограмм в центральной части изображения интерполяция выравненного значения пиксела осуществляется на основе векторов выравнивания четырех смежных блоков с помощью равенства

$$i_{\text{ВРНЕ}}(L_{\text{LDR}}, y, x) = \sum_{j=0}^1 \sum_{i=0}^1 c(j, i) h(L_{\text{LDR}}, y, x, j, i) \quad (18)$$

при $y = \overline{y_B Y_B, (y_B + 1) Y_B - 1}$, $x = \overline{x_B X_B, (x_B + 1) X_B - 1}$, $y_B = \overline{1, B_Y - 2}$, $x_B = \overline{1, B_X - 2}$, где $c(j, i)$ – весовые коэффициенты, определяющие вклад значения выравнивания каждого блока, $\sum_{j=0}^1 \sum_{i=0}^1 c(j, i) = 1$; $h(L_{\text{LDR}}, y, x, j, i)$ – локальное или общее значение выравнивания в зависимости от значения $m_{\text{GB}}(y_B + j, x_B + i) = 1$, которое находится с помощью выражения

$$h(L_{\text{LDR}}, y, x, j, i) = \begin{cases} h_E(y_B + j, x_B + i, L_{\text{LDR}}, i_{\text{HDR}}(y, x)) & \text{при } m_{\text{GB}}(y_B + j, x_B + i) = 1, \\ h_E(L_{\text{LDR}}, i_{\text{HDR}}(y, x)) & \text{при } m_{\text{GB}}(y_B + j, x_B + i) = 0. \end{cases}$$

На рис. 4 изображена структурная схема ВРНЕ блочно-приоритетного выравнивания и сжатия гистограмм, разработанная на основе математической модели (8)–(18).

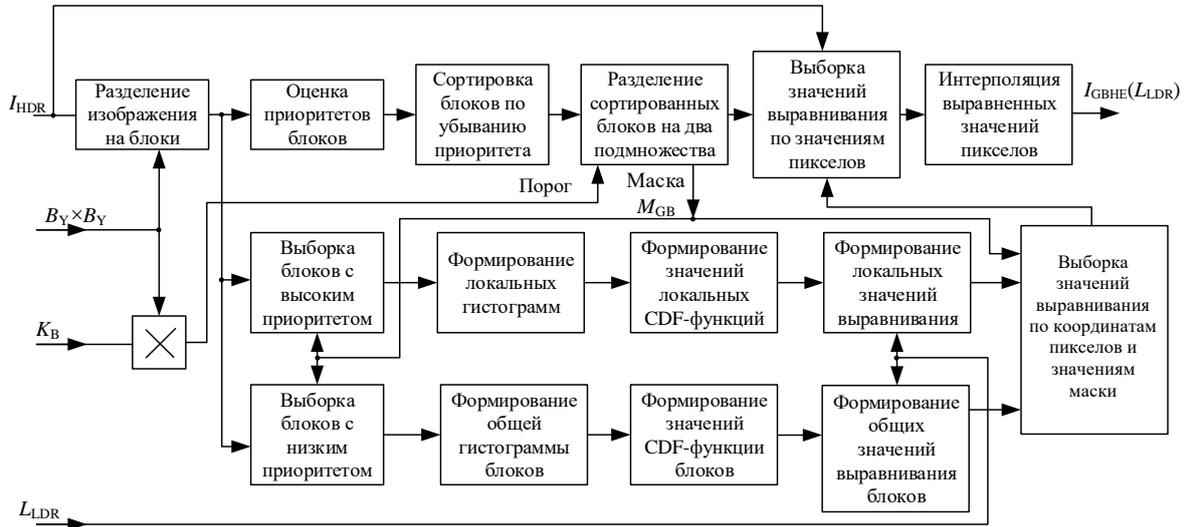


Рис. 4. Структура преобразования динамического диапазона изображения на основе блочно-приоритетного выравнивания и сжатия гистограмм

Fig. 4. Structure of image dynamic range converting based on block-structured-priority equalization and histogram compression

В приведенной схеме исходное изображение I_{HDR} с широким динамическим диапазоном разделяется на $B_Y B_X$ блоков с возможным перекрытием. Для каждого блока вычисляется значение приоритета на основе контраста, энтропии (согласно выражениям (8), (9)) или других показателей качества изображений. Блоки сортируются по убыванию приоритета в соответствии с выражения-

ми (10) или (11). Упорядоченное множество блоков разделяется на подмножества высоко- и низкоприоритетных блоков в зависимости от значения коэффициента K_B с помощью выражений (12) и (13) и формируется маска M_{GB} типов блоков с помощью выражения (14). Значение порога определяется как $B_Y B_X K_B$. Посредством маски M_{GB} блоки изображения разделяются на два подмножества. Для высокоприоритетных блоков формируются локальные гистограммы $H_{HDR}(y_B, x_B)$, интегральные функции $H_{CDF}(y_B, x_B)$ распределения яркости и векторы $H_E(y_B, x_B, L_{LDR})$ выравнивания с помощью выражения (15). Для низкоприоритетных блоков формируются общие гистограмма H_{HDR}^G , интегральная функция H_{CDF} распределения яркости и вектор $H_E(L_{LDR})$ выравнивания с помощью выражения (16). Маска M_{GB} используется также для выборки локальных $H_E(y_B, x_B, L_{LDR})$ и общего $H_E(L_{LDR})$ значений выравнивания при интерполяции выравненных значений пикселей изображения $I_{BPHE}(L_{LDR})$ с помощью выражения (18). При $L_{LDR} < L_{HDR}$ изображение $I_{BPHE}(L_{LDR})$ имеет меньший динамический диапазон по сравнению с исходным изображением I_{HDR} .

Алгоритм BPHE состоит из следующих основных шагов.

Вход: I_{HDR} , $B_Y B_X$, K_B , L_{LDR} .

1. Разделение изображения I_{HDR} на $B_Y B_X$ блоков.
2. Оценка приоритетов блоков с помощью выражений (8) или (9), сортировка блоков по убыванию приоритетов с помощью выражений (10) или (11), разделение блоков на два подмножества с помощью выражений (12) и (13), формирование маски M_{GB} с помощью выражения (14).
3. Выборка блоков с высоким приоритетом на основе K_B ; формирование локальной гистограммы, CDF-функции, значений выравнивания для каждого высокоприоритетного блока; буферизация значений выравнивания всех высокоприоритетных блоков.
4. Выборка блоков с низким приоритетом на основе K_B ; формирование общих гистограммы, CDF-функции, значений выравнивания для всех низкоприоритетных блоков.
5. Для каждого пиксела: выборка значений выравнивания смежных блоков на основе M_{GB} , интерполяция значений пикселей преобразованного изображения $I_{BPHE}(L_{LDR})$ с помощью выражения (18).

Выход: $I_{BPHE}(L_{LDR})$.

Временная сложность C_T^{BPHE} алгоритма BPHE определяется количеством операций обращения к памяти при загрузке исходного ИК-изображения I_{HDR} (YX операций); при оценке блочного контраста (энтропии) изображения I_{HDR} (YX операций); сортировке блоков ($(B_Y B_X)^2$ операций); формировании маски M_{GB} блоков (YX операций); выборке блоков (YX операций), формировании гистограмм $H_{HDR}(y_B, x_B)$ блоков (YX операций), значений глобальной H_{CDF} и блочных $H_{CDF}(y_B, x_B)$ интегральных функций распределения яркостей ($L_{HDR}(1 + B_Y B_X K_B)$ операций), значений глобального $H_E(L_{LDR})$ и блочных $H_E(y_B, x_B, L_{LDR})$ векторов выравнивания ($L_{HDR}(1 + B_Y B_X K_B)$ операций), значений выравнивания (YX операций); при интерполяции значений пикселей (YX операций) и выгрузке преобразованного изображения $I_{BPHE}(L_{LDR})$ (YX операций) с помощью выражения

$$C_T^{BPHE} = 8YX + B_Y B_X (B_Y B_X + 2L_{HDR} K_B) + 2L_{HDR}. \quad (19)$$

Пространственная сложность C_S^{BPHE} алгоритма BPHE определяется объемом оперативной памяти (в битах), необходимой для загрузки исходного ИК-изображения I_{HDR} ($YX \log_2(L_{\text{HDR}})$ бит); оценки локального контраста (энтропии) изображения I_{HDR} ($B_Y B_X \log_2(L_{\text{HDR}})$ бит); сортировки блоков ($B_Y B_X \log_2(L_{\text{HDR}})$ бит); формирования маски M_{GB} блоков (YX бит); выборки блоков ($YX \log_2(L_{\text{HDR}})$ бит); формирования локальных гистограмм $H_{\text{HDR}}(y_B, x_B)$ ($L_{\text{HDR}} \log_2(Y_B X_B) + L_{\text{HDR}} \log_2(YX - Y_B X_B B_Y B_X K_B)$ бит), общей H_{CDF} и локальных $H_{\text{CDF}}(y_B, x_B)$ интегральных функций распределения яркостей ($L_{\text{HDR}} \log_2(Y_B X_B) + L_{\text{HDR}} \log_2(YX - Y_B X_B B_Y B_X K_B)$ бит), общих $H_E(L_{\text{LDR}})$ и локальных $H_E(y_B, x_B, L_{\text{LDR}})$ значений выравнивания ($L_{\text{HDR}} \log_2(Y_B X_B) + L_{\text{HDR}} \log_2(YX - Y_B X_B B_Y B_X K_B)$ бит); выборки значений выравнивания; интерполяции значений пикселей и формирования преобразованного изображения $I_{\text{BPHE}}(L_{\text{LDR}})$ ($YX \log_2(L_{\text{LDR}})$ бит), с помощью выражения

$$C_S^{\text{BPHE}} = L_{\text{HDR}} (2 + B_Y B_X K_B) \log_2(Y_B X_B) + 3L_{\text{HDR}} \log_2(YX - Y_B X_B B_Y B_X K_B) + 2B_Y B_X \log_2(L_{\text{HDR}}) + YX (1 + 2\log_2(L_{\text{HDR}}) + \log_2(L_{\text{LDR}})). \quad (20)$$

В табл. 2 приведены значения вычислительной сложности алгоритма BPHE для ИК-изображений размерами 480×640 пикселей при $L_{\text{HDR}} = 2^{16}$, $L_{\text{LDR}} = 2^8$ и $Y_B = \{10, 16\}$.

Таблица 2

Вычислительная сложность алгоритма BPHE в зависимости от размера блока по вертикали Y_B и доли блоков, для которых реализуется локальное выравнивание гистограмм, K_B

Table 2

Computational complexity of BPHE algorithm in dependence with vertical block size Y_B and fraction of the blocks for which local equalization is performed, K_B

Y_B	Временная сложность алгоритмов Temporary algorithm complexity			Пространственная сложность алгоритмов Spatial algorithm complexity		
	$K_B = 0,75$	$K_B = 0,5$	$K_B = 0,25$	$K_B = 0,75$	$K_B = 0,5$	$K_B = 0,25$
10	532 119 552	364 347 392	196 575 232	1 688 801 735	1 131 671 291	574 459 247
16	164 465 552	111 381 392	58 297 232	653 896 252	441 756 220	229 534 588

Из табл. 1 и 2 следует, что алгоритм BPHE обеспечивает уменьшение вычислительной сложности по сравнению с алгоритмом АНЕ пропорционально коэффициенту K_B , определяющему отношение количества высокоприоритетных блоков к количеству всех блоков. Временная сложность уменьшается в 1,26–3,42 раза при размерах блока 10×10 пикселей и в 1,30–3,67 раза при размерах блока 16×16 пикселей. Пространственная сложность уменьшается в 1,33–3,90 раза при размерах блока 10×10 пикселей и в 1,31–3,74 раза при размерах блока 16×16 пикселей. Алгоритм BPHE проигрывает алгоритму HE при размере блока 16×16 пикселей во временной сложности в 43–121 раз, в пространственной сложности – в 21–60 раз; при размере блока 10×10 пикселей проигрывает во временной сложности в 145–391 раз, в пространственной сложности – в 52–154 раза.

Оценка эффективности алгоритмов выравнивания и сжатия гистограммы ИК-изображений. На рис. 5–8 показаны ИК-изображения с узким динамическим диапазоном ($L_{\text{LDR}} = 2^8$), полученные из исходного ИК-изображения с широким динамическим диапазоном ($L_{\text{HDR}} = 2^{16}$) (см. рис. 2, а), при использовании контраста $D_{\text{ST}}(y_B, x_B)$ и энтропии $E_I(y_B, x_B)$ для определения приоритетности блоков при различных значениях Y_B и K_B .

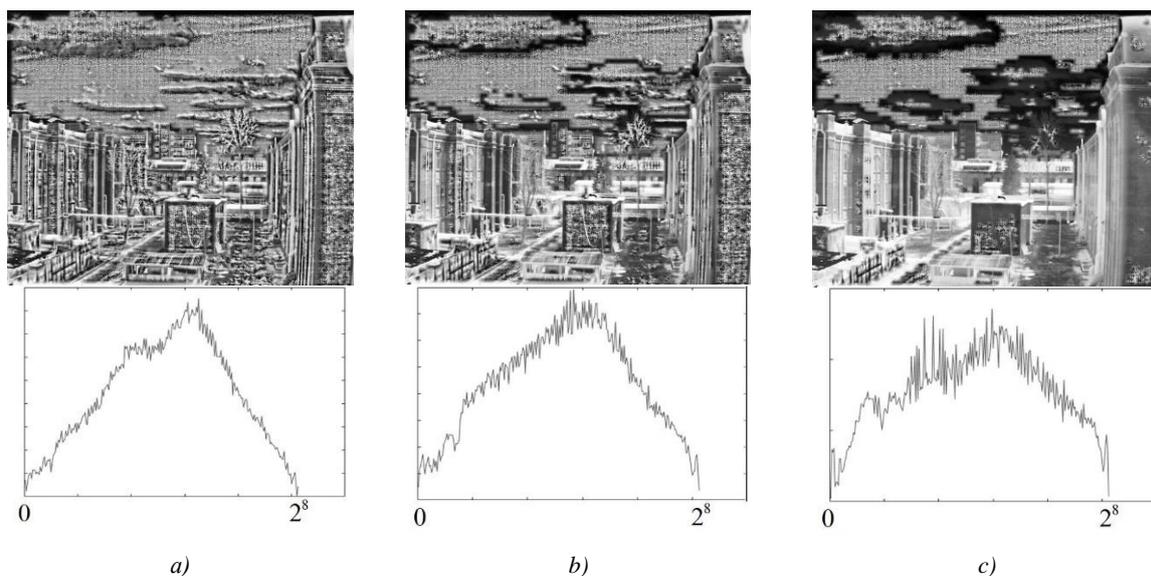


Рис. 5. ИК-изображения с $L_{\text{LDR}} = 2^8$, полученные при $Y_b = 10$ и использовании контраста для определения приоритетности блоков: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

Fig. 5. IR-images and histograms for $L_{\text{LDR}} = 2^8$ received in case of $Y_b = 10$ and contrast using for block priority determination: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

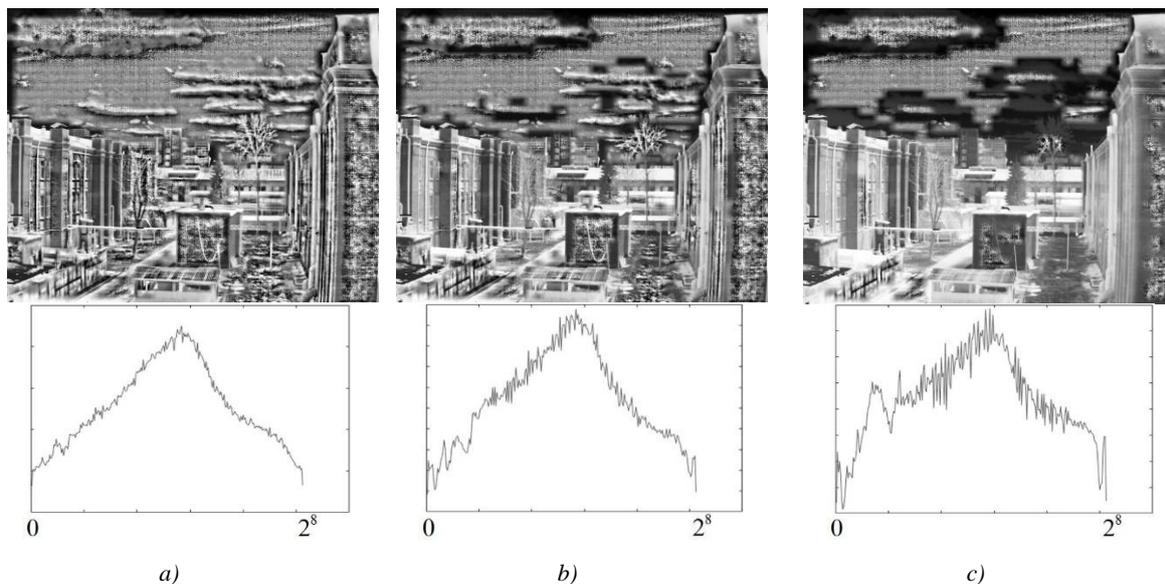


Рис. 6. ИК-изображения с $L_{\text{LDR}} = 2^8$, полученные при $Y_b = 16$ и использовании контраста для определения приоритетности блоков: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

Fig. 6. IR-images and histograms for $L_{\text{LDR}} = 2^8$ received in case of $Y_b = 16$ and contrast using for block priority determination: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

Как следует из рис. 5 и 6, результаты преобразования динамического диапазона изображений при использовании контраста $D_{\text{ST}}(y_B, x_B)$ для определения приоритетности блоков отличаются от аналогичных результатов при использовании энтропии $E_I(y_B, x_B)$ усилением блочного эффекта на однородных по яркости участках изображения.

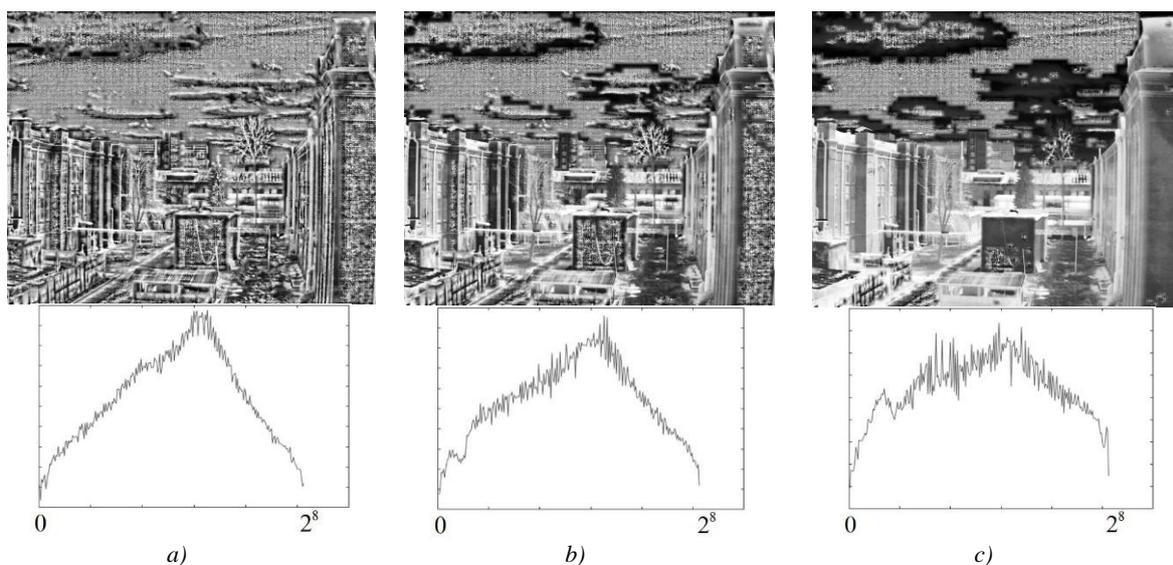


Рис. 7. ИК-изображения с $L_{\text{LDR}} = 2^8$, полученные при $Y_b = 10$ и использовании энтропии для определения приоритетности блоков: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

Fig. 7. IR-images and histograms for $L_{\text{LDR}} = 2^8$ received in case of $Y_b = 10$ and entropy using for block priority determination: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

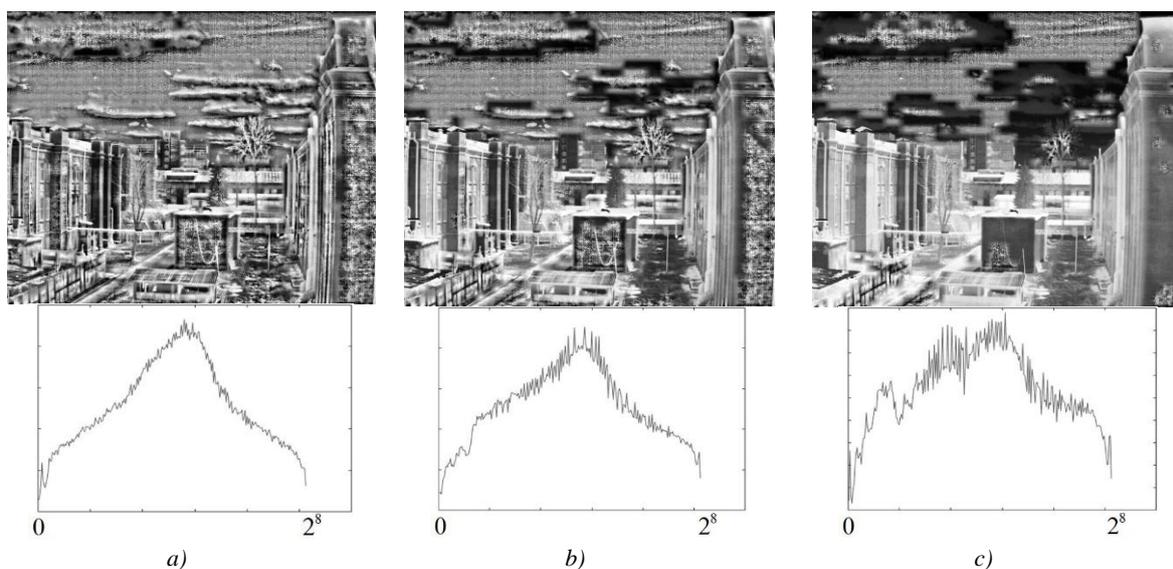


Рис. 8. ИК-изображения с $L_{\text{LDR}} = 2^8$, полученные при $Y_b = 16$ и использовании энтропии для определения приоритетности блоков: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

Fig. 8. IR-images and histograms for $L_{\text{LDR}} = 2^8$ received in case of $Y_b = 16$ and entropy using for block priority determination: а) $K_b = 0,75$; б) $K_b = 0,50$; в) $K_b = 0,25$

В табл. 3 приведены значения показателей SN, SF и TMQI для изображений на рис. 5–8. Видно, что с уменьшением количества высокоприоритетных блоков показатели SN и TMQI уменьшаются. Это свидетельствует об ухудшении статистической естественности ИК-изображений после тонового преобразования. Показатель SF, характеризующий структурную точность, нестабилен. Вместе с тем показатели для изображений, полученных с использованием контраста при определении приоритетности блоков, лучше, чем показатели для изображений, полученных с использованием энтропии, несмотря на более сильное проявление блочного эффекта. Это го-

ворит о низкой точности показателей SF и TMQI для оценки качества тонового преобразования ИК-изображений при акцентировании внимания на улучшении детализации изображений в светлых и темных областях.

Таблица 3

Значения показателей SN, SF и TMQI в зависимости от критерия сортировки и размера блока по вертикали Y_B и доли блоков, для которых реализуется локальное выравнивание гистограмм, K_B

Table 3

Values of SN, SF, TMQI in dependence with block sort criteria, vertical block size Y_B and fraction of the blocks for which local equalization is performed, K_B

Y_B	Критерий сортировки блоков <i>Block sort criteria</i>	K_B	SN	SF	TMQI
10	Контраст <i>Contrast</i>	0,75	0,6450	0,5094	0,7981
		0,50	0,5413	0,4918	0,7741
		0,25	0,3032	0,5531	0,7543
	Энтропия <i>Entropy</i>	0,75	0,6567	0,4707	0,7844
		0,50	0,5410	0,4694	0,7649
		0,25	0,2956	0,5396	0,7477
16	Контраст <i>Contrast</i>	0,75	0,5432	0,6351	0,8267
		0,50	0,4294	0,5956	0,7934
		0,25	0,2385	0,6191	0,7643
	Энтропия <i>Entropy</i>	0,75	0,5485	0,6147	0,8207
		0,50	0,4235	0,5566	0,7784
		0,25	0,2239	0,6076	0,7572

В табл. 4 приведены показатели выигрышей по контрасту D_{ST} , среднему градиенту G_A и энтропии E_I для алгоритма АНЕ по сравнению с алгоритмом ВРНЕ и для алгоритма ВРНЕ по сравнению с алгоритмом НЕ, усредненные по блокам 64×64 пикселей и 94 ландшафтными ИК-изображениям размером 480×640 пикселей при $Y_B = \{10, 16\}$. Для вычисления выигрышей использованы выражения

$$D_{ST} = D_{ST}^B / D_{ST}^Z, \quad (21)$$

$$G_A = G_A^B / G_A^Z, \quad (22)$$

$$E_I = E_I^B / E_I^Z, \quad (23)$$

где $B = \{АНЕ, ВРНЕ\}$, $Z = \{ВРНЕ, НЕ\}$.

Из табл. 4 следует, что по контрасту D_{ST} , среднему градиенту G_A и энтропии E_I для алгоритма ВРНЕ при $K_B = 0,75$ выигрыши по сравнению с алгоритмом НЕ ($D_{ST} = 1,74-1,95$, $G_A = 3,4-4,12$, $E_I = 1,16-1,24$) существенно превосходят проигрыши по сравнению с алгоритмом АНЕ (выигрыши АНЕ по сравнению с ВРНЕ: $D_{ST} = 1,03-1,10$, $G_A = 1,09-1,22$, $E_I = 1,01-1,02$), а при $K_B = 0,25$ выигрыши по сравнению с алгоритмом НЕ ($D_{ST} = 1,36-1,48$, $G_A = 1,87-2,22$, $E_I = 1,07-1,13$) соизмеримы с проигрышами по сравнению с алгоритмом АНЕ (выигрыши АНЕ по сравнению с ВРНЕ: $D_{ST} = 1,34-1,36$, $G_A = 1,91-2,11$, $E_I = 1,10$). Таким образом, при изменении значения K_B в диапазоне $0,25-0,75$, т. е. использовании 25–75 % блоков изображения для локального выравнивания гистограмм, и при соответствующем снижении вычислительной сложности (как следует из табл. 2) разработанный алгоритм ВРНЕ более эффективен для уменьшения динамического диапазона изображений по сравнению с алгоритмом АНЕ.

Таблица 4

Средние выигрыши в характеристиках ИК-изображений после преобразования динамического диапазона (Y_B – размер блока по вертикали; K_B – доля блоков, для которых реализуется локальное выравнивание гистограмм; D_{ST} – показатель выигрыша по контрасту;

G_A – показатель выигрыша по среднему градиенту; E_I – показатель выигрыша по энтропии)

Table 4

Mean improvement of IR images characteristics after dynamic range converting (Y_B – vertical block size,

K_B – fraction of the blocks for which local equalization is performed, D_{ST} – contrast scoring index,

G_A – average gradient scoring index, E_I – entropy scoring index)

Y_B	Критерий сортировки блоков <i>Block sort criteria</i>	K_B	Средние выигрыши АНЕ по сравнению с ВРНЕ <i>Average gain ANE in comparison with BPNE</i>			Средние выигрыши ВРНЕ по сравнению с НЕ <i>Average gain BPNE in comparison with HE</i>		
			D_{ST}	G_A	E_I	D_{ST}	G_A	E_I
10	Контраст <i>Contrast</i>	0,75	1,04	1,09	1,01	1,90	3,99	1,23
		0,50	1,12	1,34	1,04	1,78	3,37	1,20
		0,25	1,34	2,01	1,10	1,48	2,22	1,13
	Энтропия <i>Entropy</i>	0,75	1,03	1,10	1,01	1,95	4,12	1,24
		0,50	1,10	1,36	1,04	1,70	3,07	1,13
		0,25	1,35	2,11	1,10	1,48	2,12	1,13
16	Контраст <i>Contrast</i>	0,75	1,10	1,22	1,02	1,74	3,36	1,16
		0,50	1,17	1,40	1,05	1,61	2,79	1,13
		0,25	1,35	1,91	1,10	1,36	1,92	1,07
	Энтропия <i>Entropy</i>	0,75	1,03	1,11	1,01	1,83	3,40	1,21
		0,50	1,12	1,37	1,04	1,73	2,80	1,19
		0,25	1,36	2,06	1,10	1,42	1,87	1,12

Заключение. В работе предложены математическая модель, структура и алгоритм ВРНЕ блочно-приоритетного выравнивания и сжатия гистограмм для уменьшения динамического диапазона ИК-изображений, основанные на разделении множества всех блоков исходного изображения на подмножества высокоприоритетных и низкоприоритетных блоков в зависимости от их яркостных статистических свойств и на использовании при интерполяции значений пикселей преобразованного изображения с узким динамическим диапазоном локальных значений выравнивания для высокоприоритетных блоков и общих значений выравнивания для низкоприоритетных блоков. Выигрыши алгоритма ВРНЕ по сравнению с алгоритмом НЕ глобального выравнивания гистограммы, а также проигрыши по сравнению с алгоритмом АНЕ адаптивного выравнивания гистограммы по контрасту, среднему градиенту и энтропии изменяются пропорционально отношению количества высокоприоритетных блоков к количеству всех блоков. От этого отношения пропорционально зависят также выигрыши алгоритма ВРНЕ по сравнению с алгоритмом АНЕ, а также его проигрыши по сравнению с алгоритмом НЕ в вычислительной сложности. Эксперименты по уменьшению динамического диапазона 94 ландшафтных ИК-изображений размером 480×640 пикселей показали, что при изменении отношения количества высокоприоритетных блоков к количеству всех блоков в диапазоне 0,25–0,75 алгоритм ВРНЕ более эффективен по сравнению с алгоритмами АНЕ и НЕ. В данном диапазоне алгоритм ВРНЕ имеет в 1,3–3,5 раза меньшую временную и в 1,3–3,9 меньшую пространственную сложность по сравнению с алгоритмом АНЕ при ухудшении контраста преобразованных изображений в 1,1–1,4 раза.

Вклад авторов. С. И. Рудиков разработал математическую модель, структуру, алгоритм и программную модель выравнивания гистограммы изображений, провел эксперименты; В. Ю. Цветков осуществил постановку задачи исследования и научное редактирование статьи; А. П. Шкадаревич определил направление и цель исследования, интерпретировал полученные результаты.

Список использованных источников

1. Garcia, F. Noise removal and real-time detail enhancement of high-dynamic-range infrared images with time consistency / F. Garcia, C. Schockaert, B. Mirbach // Intern. Conf. on Quality Control by Artificial Vision, SPIE Proceedings, Le Creusot, France, 3 June 2015. – Le Creusot, 2015. – Vol. 9534. <https://doi.org/10.1117/12.2182896>
2. Yang, K.-F. A biological vision inspired framework for image enhancement in poor visibility conditions / K.-F. Yang, X.-S. Zhang, Y.-J. Li // IEEE Transactions on Image Processing. – 2020. – Vol. 29. – P. 1493–1506. <https://doi.org/10.1109/tip.2019.2938310>
3. Старовойтов, В. В. Адаптивное сжатие широкого динамического диапазона цифровых радарных спутниковых изображений / В. В. Старовойтов // Информатика. – 2018. – № 15(1). – С. 81–91.
4. Lee, J. W. Local tone mapping using K-means algorithm and automatic gamma setting / J. W. Lee, R. Park, S. Chang // IEEE Intern. Conf. on Consumer Electronics (ICCE). – Las Vegas, NV, USA, 2011. – P. 807–808. <https://doi.org/10.1109/ICCE.2011.5722876>
5. Iwahashi, M. Two layer lossless coding of HDR images / M. Iwahashi, H. Kiya // IEEE Intern. Conf. on Acoustics, Speech and Signal Processing. – Vancouver, BC, Canada, 2013. – P. 1340–1344. <https://doi.org/10.1109/ICASSP.2013.6637869>
6. Khan, I. R. Tone-mapping using perceptual-quantizer and image histogram / I. R. Khan, W. Aziz, S.-O. Shim // IEEE Access. – 2020. – Vol. 8. – P. 31350–31358. <https://doi.org/10.1109/ACCESS.2020.2973273>
7. Adaptive contrast adjustment for postprocessing of tone mapped high dynamic range images / M. Narwaria [et al.] // IEEE Intern. Symp. on Circuits and Systems (ISCAS). – Beijing, China, 2013. – P. 1103–1106. <https://doi.org/10.1109/ISCAS.2013.6572043>
8. Thai, B. C. HDR image tone mapping approach based on near optimal separable adaptive lifting scheme / B. C. Thai, A. Mokraoui, B. Matei // Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA). – Poznan, Poland, 2018. – P. 108–113. <https://doi.org/10.23919/SPA.2018.8563293>
9. Huang, P. Multi-scale bilateral grid for image tone mapping / P. Huang, Z. Su, Z. Li // Intern. Conf. on Multimedia Technology. – Hangzhou, 2011. – P. 3143–3146. <https://doi.org/10.1109/ICMT.2011.6003057>
10. HDR compression based on image matting Laplacian / C.-C. Huang [et al.] // IEEE Intern. Conf. on Consumer Electronics-Taiwan (ICCE-TW). – Nantou, Taiwan, 2016. – P. 1–2. <https://doi.org/10.1109/ICCE-TW.2016.7520957>
11. A fast multi-scale decomposition based tone mapping algorithm for High Dynamic Range images / Q. Chen [et al.] // IEEE Intern. Conf. on Systems, Man, and Cybernetics (SMC). – Budapest, 2016. – P. 001455–001460. <https://doi.org/10.1109/SMC.2016.7844442>
12. High dynamic tone mapping algorithm based on wavelet domain image fusion / W. Liu [et al.] // 13th IEEE Conf. on Industrial Electronics and Applications (ICIEA). – Wuhan, China, 2018. – P. 1945–1950. <https://doi.org/10.1109/ICIEA.2018.8398027>
13. Lin, Y. High dynamic range image composition using a linear interpolation approach / Y. Lin, M. Huang, C. Wang // IEEE/ACIS 15th Intern. Conf. on Computer and Information Science (ICIS). – Okayama, Japan, 2016. – P. 1–6. <https://doi.org/10.1109/ICIS.2016.7550796>
14. Tone mapping operators: progressing towards semantic-awareness / A. Goswami [et al.] // IEEE Intern. Conf. on Multimedia & Expo Workshops (ICMEW). – London, UK, 2020. – P. 1–6. <https://doi.org/10.1109/ICMEW46912.2020.9106057>
15. Lee, J. W. Local tone mapping using K-means algorithm and automatic gamma setting / J. W. Lee, R. Park, S. Chang // IEEE Intern. Conf. on Consumer Electronics (ICCE). – Las Vegas, NV, USA, 2011. – P. 807–808. <https://doi.org/10.1109/ICCE.2011.5722876>
16. Guangjun, Z. An improved tone mapping algorithm for High Dynamic Range images / Z. Guangjun, L. Yan // Intern. Conf. on Computer Application and System Modeling (ICCASM 2010). – Taiyuan, 2010. – P. V2-466–V2-468. <https://doi.org/10.1109/ICCASM.2010.5620562>
17. Banic, N. Puma: A high-quality retinex-based tone mapping operator / N. Banic, S. Loncaric // 24th European Signal Processing Conf. (EUSIPCO). – Budapest, Hungary, 2016. – P. 943–947. <https://doi.org/10.1109/EUSIPCO.2016.7760387>
18. Adversarial and adaptive tone mapping operator for high dynamic range images / X. Cao [et al.] // IEEE Symp. Series on Computational Intelligence (SSCI). – Canberra, Australia, 2020. – P. 1814–1821. <https://doi.org/10.1109/SSCI47803.2020.9308535>
19. Kumar, N. A. M. Real-time implementation of a novel detail enhancement algorithm for thermal imager / N. A. M. Kumar, B. S. Ravishankar, C. R. Patil // IEEE Uttar Pradesh Section Intern. Conf. on Electrical, Computer and Electronics Engineering (UPCON). – Varanasi, India, 2016. – P. 1–6. <https://doi.org/10.1109/UPCON.2016.7894614>

20. Peng, Y. Detail enhancement for infrared images based on propagated image filter / Y. Peng, Y. Yan, J. Zhao // *Mathematical Problems in Engineering*. – 2016. – Vol. 2016. – P. 1–12. <https://doi.org/10.1155/2016/9410368>
21. Display and detail enhancement for high-dynamic-range infrared images / C. Zuo [et al.] // *Optical Engineering*. – 2011. – Vol. 50(12). – P. 127401-1-10. <https://doi.org/10.1117/1.3659698>
22. Infrared image adaptive enhancement guided by energy of gradient transformation and multiscale image fusion / F. Chen [et al.] // *Applied Sciences*. – 2020. – Vol. 10. – P. 1–21. <https://doi.org/10.3390/app10186262>
23. Kim, T. K. Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering / T. K. Kim, J. K. Paik, B. S. Kang // *IEEE Transactions on Consumer Electronics*. – 1998. – Vol. 44, no. 1. – P. 82–87. <https://doi.org/10.1109/30.663733>
24. Nithyananda, C. R. Review on histogram equalization based image enhancement techniques / C. R. Nithyananda, A. C. Ramachandra // *Intern. Conf. on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. – Chennai, 2016. – P. 2512–2517. <https://doi.org/10.1109/ICEEOT.2016.7755145>
25. Independence of luminance and contrast in natural scenes and in the early visual system / V. Mante [et al.] // *Nature Neuroscience*. – 2005. – Vol. 8. – P. 1690–1697. <https://doi.org/10.1038/nn1556>
26. Wang, Z. Multiscale structural similarity for image quality assessment / Z. Wang, E. P. Simoncelli, A. C. Bovik // *The Thirty-Seventh Asilomar Conf. on Signals, Systems & Computers*. – Pacific Grove, CA, USA, 2003. – Vol. 2. – P. 1398–1402. <https://doi.org/10.1109/ACSSC.2003.1292216>
27. Yeganeh, H. Objective quality assessment of tone-mapped images / H. Yeganeh, Z. Wang // *IEEE Transactions on Image Processing*. – 2013. – Vol. 22, no. 2. – P. 657–667. <https://doi.org/10.1109/TIP.2012.2221725>

References

1. Garcia F., Schockaert C., Mirbach B. Noise removal and real-time detail enhancement of high-dynamic-range infrared images with time consistency. *International Conference on Quality Control by Artificial Vision, SPIE Proceedings, Le Creusot, France, 3 June 2015*. Le Creusot, 2015, vol. 9534. <https://doi.org/10.1117/12.2182896>
2. Yang K.-F., Zhang X.-S., Li Y.-J. A biological vision inspired framework for image enhancement in poor visibility conditions. *IEEE Transactions on Image Processing*, 2020, vol. 29, pp. 1493–1506. <https://doi.org/10.1109/tip.2019.2938310>
3. Starovoitov V. V. *Adaptive compressing of the high dynamic range of digital radar satellite images*. Informatika [Informatiks], 2018, no. 15(1), pp. 81–91 (In Russ.).
4. Lee J. W., Park R., Chang S. Local tone mapping using K-means algorithm and automatic gamma setting. *IEEE International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, USA, 2011, pp. 807–808. <https://doi.org/10.1109/ICCE.2011.5722876>
5. Iwahashi M., Kiya H. Two layer lossless coding of HDR images. *IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada, 2013, pp. 1340–1344. <https://doi.org/10.1109/ICASSP.2013.6637869>
6. Khan I. R., Aziz W., Shim S.-O. Tone-mapping using perceptual-quantizer and image histogram. *IEEE Access*, 2020, vol. 8, pp. 31350–31358. <https://doi.org/10.1109/ACCESS.2020.2973273>
7. Narwaria M., Da Silva M. P., Le Callet P., Pepion R. Adaptive contrast adjustment for postprocessing of tone mapped high dynamic range images. *IEEE International Symposium on Circuits and Systems (ISCAS)*. Beijing, China, 2013, pp. 1103–1106. <https://doi.org/10.1109/ISCAS.2013.6572043>
8. Thai B. C., Mokraoui A., Matei B. HDR image tone mapping approach based on near optimal separable adaptive lifting scheme. *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. Poznan, Poland, 2018, pp. 108–113. <https://doi.org/10.23919/SPA.2018.8563293>
9. Huang P., Su Z., Li Z. Multi-scale bilateral grid for image tone mapping. *International Conference on Multimedia Technology*. Hangzhou, 2011, pp. 3143–3146. <https://doi.org/10.1109/ICMT.2011.6003057>
10. Huang C.-C., Ismail, Cai M.-X., Vu H. T. HDR compression based on image matting Laplacian. *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. Nantou, Taiwan, 2016, pp. 1–2. <https://doi.org/10.1109/ICCE-TW.2016.7520957>
11. Chen Q., Liu X., Ran H., Dong S., Cui D., Deng X., Wang J. A fast multi-scale decomposition based tone mapping algorithm for High Dynamic Range images. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Budapest, 2016, pp. 001455–001460. <https://doi.org/10.1109/SMC.2016.7844442>
12. Liu W., Wang Q., Liu Y., Li N. High dynamic tone mapping algorithm based on wavelet domain image fusion. *13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. Wuhan, China, 2018, pp. 1945–1950. <https://doi.org/10.1109/ICIEA.2018.8398027>

13. Lin Y., Huang M., Wang C. High dynamic range image composition using a linear interpolation approach. *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. Okayama, Japan, 2016, pp. 1–6. <https://doi.org/10.1109/ICIS.2016.7550796>
14. Goswami A., Petrovich M., Hauser W., Dufaux F. Tone mapping operators: progressing towards semantic-awareness. *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. London, UK, 2020, pp. 1–6. <https://doi.org/10.1109/ICMEW46912.2020.9106057>
15. Lee J. W., Park R., Chang S. Local tone mapping using K-means algorithm and automatic gamma setting. *IEEE International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, USA, 2011, pp. 807–808. <https://doi.org/10.1109/ICCE.2011.5722876>
16. Guangjun Z., Yan L. An improved tone mapping algorithm for High Dynamic Range images. *International Conference on Computer Application and System Modeling (ICCASM 2010)*. Taiyuan, 2010, pp. V2-466–V2-468. <https://doi.org/10.1109/ICCASM.2010.5620562>
17. Banic N., Loncaric S. Puma: A high-quality retinex-based tone mapping operator. *24th European Signal Processing Conference (EUSIPCO)*. Budapest, Hungary, 2016, pp. 943–947. <https://doi.org/10.1109/EUSIPCO.2016.7760387>
18. Cao X., Lai K., Yanushkevich S. N., Smith M. R. Adversarial and Adaptive Tone Mapping Operator for High Dynamic Range Images. *IEEE Symposium Series on Computational Intelligence (SSCI)*. Canberra, Australia, 2020, pp. 1814–1821. <https://doi.org/10.1109/SSCI47803.2020.9308535>
19. Kumar N. A. M., Ravishankar B. S., Patil C. R. Real-time implementation of a novel detail enhancement algorithm for thermal imager. *IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*. Varanasi, India, 2016, pp. 1–6. <https://doi.org/10.1109/UPCON.2016.7894614>
20. Peng Y., Yan Y., Zhao J. Detail enhancement for infrared images based on propagated image filter. *Mathematical Problems in Engineering*, 2016, vol. 2016, pp. 1–12. <https://doi.org/10.1155/2016/9410368>
21. Zuo C., Chen Q., Liu N., Ren J., Sui X. Display and detail enhancement for high-dynamic-range infrared images. *Optical Engineering*, 2011, vol. 50(12), pp. 127401-1-10. <https://doi.org/10.1117/1.3659698>
22. Chen F., Zhang J., Cai J., Xu T., Lu G., Peng X. Infrared image adaptive enhancement guided by energy of gradient transformation and multiscale image fusion. *Applied Sciences*, 2020, vol. 10, pp. 1–21. <https://doi.org/10.3390/app10186262>
23. Kim T. K., Paik J. K., Kang B. S. Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering. *IEEE Transactions on Consumer Electronics*, 1998, vol. 44, no. 1, pp. 82–87. <https://doi.org/10.1109/30.663733>
24. Nithyananda C. R., Ramachandra A. C. Review on histogram equalization based image enhancement techniques. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. Chennai, 2016, pp. 2512–2517.
25. Mante V., Frazor R. A., Bonin V., Geisler W. S., Carandini M. Independence of luminance and contrast in natural scenes and in the early visual system. *Nature Neuroscience*, 2005, vol. 8, pp. 1690–1697. <https://doi.org/10.1038/nn1556>
26. Wang Z., Simoncelli E. P., Bovik A. C. Multiscale structural similarity for image quality assessment. *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*. Pacific Grove, CA, USA, 2003, vol. 2, pp. 1398–1402. <https://doi.org/10.1109/ACSSC.2003.1292216>
27. Yeganeh H., Wang Z. Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 2, pp. 657–667. <https://doi.org/10.1109/TIP.2012.2221725>

Информация об авторах

Рудиков Станислав Игоревич, магистр технических наук, заместитель директора по информационным технологиям, Научно-технический центр «ЛЭМТ» БелОМО.
E-mail: stanislav.rudikov@gmail.com

Цветков Виктор Юрьевич, доктор технических наук, профессор, заведующий кафедрой инфокоммуникационных технологий, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: vtsvet@bsuir.by

Шкадаревич Алексей Петрович, академик Национальной академии наук Беларуси, доктор физико-математических наук, профессор, директор, Научно-технический центр «ЛЭМТ» БелОМО.
E-mail: office@lemt.by

Information about the authors

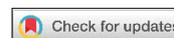
Stanislav I. Rudikov, M. Sc. (Eng.), Information Technology Deputy Director, Scientific and Technical Center LEMT of the BelOMO.
E-mail: stanislav.rudikov@gmail.com

Viktar Yu. Tsviatkou, D. Sc. (Eng.), Prof., Head of the Department of Infocommunication Technologies, Belarusian State University of Informatics and Radioelectronics.
E-mail: vtsvet@bsuir.by

Alexey P. Shkadarevich, Academician of the National Academy of Science of Belarus, D. Sc. (Phys.-Math.), Prof., Director, Scientific and Technical Center LEMT of the BelOMO.
E-mail: office@lemt.by

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.717
<https://doi.org/10.37661/1816-0301-2022-19-2-26-55>

Оригинальная статья
Original Paper

Экспериментальное сравнение эффективности программ минимизации систем булевых функций в классе дизъюнктивных нормальных форм

П. Н. Бибило[✉], И. П. Логинова

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
[✉]E-mail: bibilo@newman.bas-net.by*

Аннотация

Цели. Методы, алгоритмы и программы решения задач минимизации дизъюнктивных нормальных форм (ДНФ) представлений булевых функций широко используются при проектировании цифровых систем для уменьшения сложности (площади кристаллов) функциональных комбинационных блоков, размещаемых в составе цифровых СБИС.

Целью работы является сравнение программ, входящих в отечественную систему FLC-2 логической оптимизации, с двумя широко известными и свободно распространяемыми зарубежными программами минимизации Espresso ПС и АВС.

Методы. Для сравнения программ использованы четыре набора примеров входных данных: широко известные примеры, на которых проверялась эффективность программы Espresso ПС, псевдослучайные системы ДНФ и два набора промышленных примеров из практики проектирования логических схем. Предложены программные средства для применения программ совместной минимизации при отдельной минимизации функций. Разработаны алгоритмы и программы распараллеливания вычислений при отдельной минимизации функций в классе ДНФ.

Результаты. Выявлены области предпочтительного использования и время работы программ для исходных (минимизируемых) систем функций, характеризуемых большими значениями параметров (десятками аргументов и функций, десятками тысяч элементарных конъюнкций) и различными формами задания входных данных. Изучена эффективность применения программ минимизации для различных форм задания входных данных: ДНФ, ортогонализированных ДНФ, VDD-представлений систем функций, таблиц истинности и систем совершенных ДНФ.

Заключение. Результаты экспериментов показывают эффективность параллельных программ. Они позволяют сокращать время вычислений и увеличивать размерности решаемых задач отдельной минимизации систем булевых функций.

Ключевые слова: система булевых функций, дизъюнктивная нормальная форма (ДНФ), кратчайшая ДНФ, отдельная минимизация функций, кратчайшая система ДНФ, совместная минимизация функций, VDD-представления, стандарт OpenMP

Для цитирования. Бибило, П. Н. Экспериментальное сравнение эффективности программ минимизации систем булевых функций в классе дизъюнктивных нормальных форм / П. Н. Бибило, И. П. Логинава // Информатика. – 2022. – Т. 19, № 2. – С. 26–55. <https://doi.org/10.37661/1816-0301-2022-19-2-26-55>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 20.01.2022
Подписана в печать | Accepted 28.02.2022
Опубликована | Published 29.06.2022

Experimental comparison of the effectiveness of programs for minimizing systems of Boolean functions in the class of disjunctive normal forms

Petr N. Bibilo[✉], Irina P. Loginova

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
[✉]E-mail: bibilo@newman.bas-net.by*

Abstract

Objectives. Methods, algorithms and programs for solving problems of minimizing the DNF representations of Boolean functions are widely used in the design of digital systems to reduce the complexity (crystal area) of functional combinational blocks of digital systems placed into digital VLSI.

The objective of the work is experimental comparison of domestic programs for minimizing Boolean functions in the DNF class included in the FLC-2 with two well-known foreign freely distributed programs for minimizing DNF known as Espresso IIC and ABC.

Methods. Four sets sample of input data were used to compare the programs – there are widely known examples on which the effectiveness of the Espresso IIC program was tested and two sets of industrial examples from the practice of designing the logic circuits. Algorithms and programs for parallelization of calculations when separate functions of minimizing have been developed. Software tools for the application of joint minimization programs with separate minimization of functions are proposed.

Results. The areas of preferred use and the execution time of programs for the source systems of functions (for minimization) characterized by large parameter values of dozens of arguments and functions, tens of thousands of elementary conjunctions are revealed. The efficiency of application of minimization programs for various forms of input data assignment is investigated – DNF, orthogonalized DNF, BDD (Binary Decision Diagrams) representations for systems of functions, truth tables and perfect DNF systems.

Conclusion. The experimental results show the effectiveness of parallel programs – reducing the calculation time and increasing the dimensions of solved problems of separate minimization of Boolean function systems.

Keywords: system of Boolean functions, disjunctive normal form (DNF), shortest DNF, separate minimization of functions, the shortest DNF system, joint minimization of functions, Binary Decision Diagram (BDD), OpenMP standard

For citation. Bibilo P. N., Loginova I. P. *Experimental comparison of the effectiveness of programs for minimizing systems of Boolean functions in the class of disjunctive normal forms.* Informatika [Informatics], 2022, vol. 19, no. 2, pp. 26–55 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-2-26-55>

Conflict of interest. The authors declare of no conflict of interest.

Введение. Задача минимизации представлений булевых функций и систем в классе ДНФ является классической задачей теории булевых функций, начало исследований положено в работах Квайна [1] и МакКласки [2]. Как утверждается в статье [3], число публикаций по данной проблематике превышает тысячу. Например, в обзоре [4] процитировано 217 работ.

Задачу минимизации ДНФ-представлений в учебной литературе часто называют задачей минимизации булевых функций. Эта задача имеет многочисленные практические применения, среди которых, пожалуй, важнейшее место занимает ее использование в качестве средства технологически независимой логической оптимизации при синтезе логических схем [5]. Минимизация ДНФ одной булевой функции традиционно применялась (и применяется в настоящее время) в качестве метода глобальной оптимизации – начального этапа синтеза одновыходных комбинационных схем. В случае многовыходных комбинационных схем решается задача совместной либо раздельной минимизации системы булевых функций. Критериями оптимизации выступает число элементарных конъюнкций либо число литералов в минимизированных системах ДНФ. Появление программируемых логических матриц (ПЛМ) – функциональных блоков заказных СБИС – потребовало развития методов и программ совместной минимизации систем булевых функций, применяемых для сокращения площади ПЛМ [6]. Появление схем FPGA (от англ. Field-Programmable Gate Array, программируемая пользователем вентильная матрица) привело к развитию методов декомпозиции и оптимизации многоуровневых BDD-представлений (BDD – Binary Decision Diagram, бинарная диаграмма решений, диаграмма двоичного выбора) на основе разложения Шеннона [7]. Размерности задач проектирования комбинационной логики заказных сверхбольших интегральных схем (СБИС) и систем-на-кристалле возрастают так, что при логической оптимизации исходное функциональное описание часто требуется разбивать на блоки, подвергаемые оптимизации различными программами. Эксперименты [8] показали, что при синтезе комбинационных блоков заказных СБИС в базе библиотечных элементов для некоторых систем функций в качестве технологически независимой оптимизации более эффективна двухуровневая минимизация в классе ДНФ, для других – многоуровневая BDD-оптимизация. Минимизация используется для сокращения сложности функциональных описаний подсистем функций при их декомпозиции [8], а также как начальный этап оптимизации многоуровневых представлений систем функций, после которого следует применение факторизационных методов, позволяющих получать алгебраические скобочные представления [5].

В настоящей работе проводится экспериментальное сравнение отечественных программ минимизации булевых функций в классе ДНФ, входящих в систему логической оптимизации FLC-2 [9], с зарубежными, свободно распространяемыми программами Espresso ПС и ABC. Предлагаются средства распараллеливания как исходных данных, так и программ, что в целом позволяет обеспечивать сокращение времени решения задач минимизации функций и увеличивать размерности решаемых задач.

Основные определения и постановки задач. Булевыми называются двоичные (0, 1) функции $f(\mathbf{x}) = f(x_1, \dots, x_n)$ двоичных (булевых) переменных x_1, x_2, \dots, x_n . Пусть V^x – булево пространство, построенное над переменными булева вектора $\mathbf{x} = (x_1, \dots, x_n)$. Элементами этого пространства являются n -компонентные наборы (векторы) \mathbf{x}^* нулей и единиц. Булева функция, значения 0, 1 которой определены на всех элементах $\mathbf{x}^* \in V^x$, называется *полностью определенной*. Любая полностью определенная булева функция (кроме константы 0) может быть задана в виде ДНФ, т. е. в форме дизъюнкции элементарных конъюнкций. *Элементарная конъюнкция* – это конъюнкция литералов (булевых переменных x_i либо их инверсий \bar{x}_i). Литерал x_i называется *положительным*, литерал \bar{x}_i – *отрицательным*. Если в ДНФ каждая элементарная конъюнкция является полной, т. е. содержит литералы всех переменных, то такая ДНФ называется *совершенной* (СДНФ). Систему булевых функций будем обозначать $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, значениями системы функций на элементах \mathbf{x}^* булева пространства являются m -компонентные булевы векторы $\mathbf{f}(\mathbf{x}^*)$. Рассмотрим матричные формы задания ДНФ и СДНФ систем булевых функций.

В матричной форме система ДНФ, представляющая систему полностью определенных функций $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x}))$, $\mathbf{x} = (x_1, x_2, x_3, x_4)$, задается парой матриц (табл. 1). Строки троичной матрицы T^x представляют элементарные конъюнкции (троичные векторы – интервалы булева пространства V^x), а единичные значения элементов в булевой матрице B^f отмечают вхождения соответствующих конъюнкций в ДНФ функций:

$$D_{f_1}^1 = \bar{x}_2 x_3 \vee x_4; \quad D_{f_2}^1 = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_4; \quad D_{f_3}^1 = \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_2 x_3 x_4 \vee \bar{x}_3 \bar{x}_4; \\ D_{f_4}^1 = x_1 x_2 \bar{x}_3 \vee x_2 \bar{x}_4 \vee \bar{x}_3 \bar{x}_4.$$

Таблица 1
Матричная форма системы ДНФ

Table 1
Matrix form of the DNF system

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

Далее будут использоваться следующие обозначения:

n – число аргументов (число компонент булева вектора $\mathbf{x} = (x_1, \dots, x_n)$);

m – число функций системы $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$;

k – число общих элементарных конъюнкций, на которых заданы ДНФ всех функций системы $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$;

k_i – число элементарных конъюнкций в задании компонентной функции f_i системы;

$\sum_{i=1}^m k_i$ – суммарное число элементарных конъюнкций в ДНФ всех функций системы.

Для системы ДНФ из табл. 1 заданы следующие значения параметров: $n = 4$, $m = 4$, $k = 9$,

$k_1 = 2$, $k_2 = 3$, $k_3 = 4$, $k_4 = 3$, $\sum_{i=1}^m k_i = 2 + 3 + 4 + 3 = 12$ (число единиц в булевой матрице B^f).

Матричная форма системы СДНФ той же системы функций $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x}))$ представлена в табл. 2 и состоит из пары булевых матриц. Матрица B^x задает полные элементарные конъюнкции, матрица B^f – их вхождения в СДНФ компонентных функций системы $f(\mathbf{x})$. Для многих программ минимизации в целях более компактного представления исходных данных наборы, на которых $f(\mathbf{x}) = \mathbf{0}$, обычно не указываются. Таблица истинности отличается от матричной формы системы СДНФ тем, что в матрице B^x содержатся все 2^n полных элементарных конъюнкций, поэтому в матрице B^f могут оказаться нулевые строки. Заметим, что таких строк в табл. 2 нет. Кроме того, она является таблицей истинности – содержит все 2^4 двоичных наборов, соответствующих всем 16 полным элементарным конъюнкциям от четырех переменных.

Кратчайшей ДНФ D_f булевой функции f называется ДНФ минимальной длины, т. е. ДНФ с минимальным числом элементарных конъюнкций. Под длиной ДНФ булевой функции понимается число ее элементарных конъюнкций.

В табл. 3 приведены кратчайшие ДНФ функций из табл. 2.

Таблица 2
Таблица истинности системы полностью
определенных булевых функций

Table 2
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

Таблица 3
Раздельно минимизированные ДНФ
системы функций (табл. 2)

Table 3
Separately minimized DNF systems
of functions (table 2)

x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4
1	1	0	-	0	0	0	1
-	1	-	0	0	0	0	1
-	-	0	0	0	0	0	1
1	1	-	1	0	0	1	0
-	-	0	0	0	0	1	0
0	1	1	-	0	0	1	0
1	0	1	-	0	1	0	0
0	1	1	-	0	1	0	0
0	-	-	1	0	1	0	0
-	-	-	1	1	0	0	0
-	0	1	-	1	0	0	0

Конъюнкции из ДНФ различных компонентных функций могут совпадать. В табл. 2 это конъюнкции, представленные троичными векторами $(--00)$, $(011-)$. Для системы из табл. 3

$$k_1 = 2, k_2 = 3, k_3 = 3, k_4 = 3, \sum_{i=1}^m k_i = 11.$$

Кратчайшей системой ДНФ D_f для системы булевых функций $f(x) = (f_1(x), \dots, f_m(x))$ называется такая система ДНФ, которая содержит минимальное число общих элементарных конъюнкций, на которых заданы ДНФ D_{f_i} , $i = 1, \dots, m$, всех функций f_i системы $f(x) = (f_1(x), \dots, f_m(x))$.

Задача 1. Совместная минимизация системы булевых функций. Для заданной системы $f(x) = (f_1(x), \dots, f_m(x))$ булевых функций найти кратчайшую систему ДНФ D_f .

Задача 2. Раздельная минимизация системы булевых функций. Найти кратчайшие ДНФ D_{f_i} для каждой компонентной функции $f_i(x)$, $i = 1, \dots, m$, заданной системы $f(x) = (f_1(x), \dots, f_m(x))$ булевых функций.

Заметим, что в совместно минимизированной системе ДНФ (табл. 1) $\sum_{i=1}^m k_i = 12$. Конъюнкции, представленные троичными векторами $(--00)$, $(011-)$, $(110-)$ (см. табл. 1), входят в ДНФ пар компонентных функций.

При решении задачи 1 минимизируется число k общих элементарных конъюнкций, при решении задачи 2 минимизируются числа k_i . Для задач 1 и 2 существенную роль играют понятия импликанты и простой импликанты. *Импликантой* булевой функции $f(x_1, \dots, x_n)$ называется такая элементарная конъюнкция K литералов x_i, \bar{x}_i булевых переменных x_i , что на любом наборе значений переменных x_1, x_2, \dots, x_n , на котором K равна единице, значение функции $f(x_1, \dots, x_n)$ также равно единице. *Простой импликантой* называется та, которая перестает быть импликантой при удалении из нее любого символа. Понятие простой импликанты одной функции обобщается на случай простой импликанты системы функций [10]. Если для некоторого множества

функций системы существует простая импликанта K_{pr} , то это не означает, что она же будет являться простой импликантой для каждой функции в отдельности. Элементарная конъюнкция K_{pr} будет простой импликантой для системы функций F тогда и только тогда, когда выполняются следующие условия:

- найдется такая подсистема функций $F^* \subseteq F$, $F^* \neq \emptyset$, что K_{pr} является импликантой для каждой функции подсистемы F^* ;
- K_{pr} не будет импликантой хотя бы для одной функции подсистемы F^{**} , где $F^{**} = F^* \cup f^*$, $f^* \in F \setminus F^*$;
- импликанта K_{pr} не будет импликантой хотя бы для одной функции подсистемы F^* , если из K_{pr} удалить любой символ.

Программы минимизации. Программа Espresso [11] является самой известной программой минимизации и предназначена для совместной и раздельной минимизации систем полностью определенных и частичных булевых функций (и систем многозначных функций) в классе ДНФ по различным критериям – числу элементарных конъюнкций и числу литералов. В табл. 1 представлена кратчайшая система ДНФ для системы функций из табл. 2, полученная программой Espresso ПС (далее – Espresso).

Существуют различные варианты и модификации данной программы: Espresso-Exact (программа точной минимизации), Espresso-MV [12] (программа минимизации многозначных функций), Espresso-Signature [13]. Программе Espresso посвящена монография [11], на которую имеется большое число ссылок в научной литературе. В программе Espresso в процессе минимизации итеративно выполняются следующие операции: ESSENTIAL – выделение существенных импликант, EXPAND – расширение, REDUCE – сокращение, RESHAPE – модификация, IRREDUNDANT – безызбыточное покрытие. Выполнение процедуры ESSENTIAL заключается в выделении существенных импликант, которые входят в любое безызбыточное покрытие, состоящее из простых импликант. Эти импликанты переносятся в искомую безызбыточную ДНФ в качестве ее обязательной части и далее в поиске интервального покрытия исходных булевых функций не участвуют, что сокращает время поиска. Выполнение процедур REDUCE, EXPAND и IRREDUNDANT циклически повторяется до тех пор, пока решение не стабилизируется, т. е. пока на очередной итерации цикла не получится результат с таким же числом импликант (или литералов), что и на предыдущей итерации. Это означает, что решение достигло локального минимума относительно процедур REDUCE, EXPAND и IRREDUNDANT. Для сбрасывания полученного решения с минимума в ESPRESSO-MV используются более изощренные, чем REDUCE, процедуры LAST_GASP или SUPER_GASP [12] и NEW_GASP [14]. В экспериментах, описание которых приводится далее, использовалась программа Espresso (без ключей) для совместной минимизации системы полностью определенных функций; критерий оптимизации – число общих элементарных конъюнкций.

Программа ABC [15] для раздельной минимизации основывается на графовых моделях многоуровневых BDD-представлений систем функций. Сначала осуществляется построение совместной (shared) BDD для системы функций, затем используется модель бинарной диаграммы решений с подавлением нулей (Zero-suppressed BDD, ZDD). ZDD представляют структуры данных для решения комбинаторных задач, в которых необходима работа с множествами подмножеств. Такие структуры данных эффективны для «разреженных» объектов, в которых много дуг ведет к листовой вершине 0. Быстрое получение ZDD оказалось эффективным для раздельной минимизации ДНФ булевых функций большой размерности и используется в программе ABC. О модели ZDD можно найти информацию в работе [16, с. 358].

В программе Tie [10] путем совместной минимизации систем булевых функций, где критерий оптимизации – число общих элементарных конъюнкций, реализуется алгоритм, состоящий из следующих шагов: 1) поиск всех простых импликант исходной системы ДНФ булевых функций; 2) построение булевой матрицы A покрытия (матрицы Квайна) и ее сокращение; 3) нахождение кратчайшего столбцового покрытия матрицы A. Подробное описание алгоритма дано в работе [10]. Его особенностями являются оригинальная процедура нахождения импликант и процедура нахождения покрытия матрицы A из работы [17].

Программа *Minim* [18] предназначена для совместной и отдельной минимизации систем как полностью определенных, так и частичных булевых функций, заданных в интервальной форме. Критериями оптимизации являются число общих элементарных конъюнкций при совместной минимизации и число конъюнкций в кратчайшей ДНФ каждой компонентной функции системы при отдельной минимизации. Программа реализует обобщение алгоритма из работы [19], написана на языке C++ и использует специальные структуры данных для векторно-матричных объектов (булевых и троичных векторов и матриц) [20, 21]. Подробное описание форм представления данных, реализованных в программе алгоритмов и результатов экспериментов дано в работе [18]. В табл. 3 приведены кратчайшие ДНФ функций из табл. 2, полученные в результате отдельной минимизации программой *Minim*. Далее в экспериментах исследовался реализованный в программе *Minim* итерационный алгоритм как для совместной, так и для отдельной минимизации с заданием режима, ориентированного на получение решения лучшего качества [19].

Программы *Tie* и *Minim* включены в систему FLC-2 [9] и используют в качестве входных и выходных данных текстовый матричный формат SDF языка SF иерархического функционально-структурного описания комбинационной логики [22].

Примеры систем функций для экспериментов. Исследования были проведены для четырех наборов примеров систем булевых функций.

Первый набор примеров был представлен системами функций в формате PLA из библиотеки Berkeley PLA Test Set (URL: <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex>). Данные системы ДНФ булевых функций переводились из формата PLA в матричные описания на языке SF. В экспериментах исходными описаниями являлись матричные формы систем ДНФ, СДНФ, таблиц истинности. Если в параметрах исходного задания было указано значение $k = 2^n$, то в качестве исходных выступали таблицы истинности соответствующей системы булевых функций. Было установлено, что примеры GARY и IN0 задают одну и ту же систему функций в виде различных систем ДНФ, т. е. ДНФ с различными множествами элементарных конъюнкций. Примеры Too_large_matr, X3_matr систем ДНФ были получены из многоуровневых описаний систем функций в виде суперпозиций ДНФ.

Второй набор примеров составляли системы булевых функций, взятых из практики проектирования управляющей логики заказных СБИС. Примеры VERG1, VERG2 – системы СДНФ функций, задающих таблицы для хранения микропрограмм команд микропроцессоров; примеры Sist4, Sistem8 – системы ДНФ, задающие блоки управляющей логики в заказных СБИС. Примеры S12, ..., S16 – таблицы истинности, задающие SF-описания систем функций ($n = 12, \dots, 16$, $m = 12, \dots, 16$, $k = 2^{12}, \dots, 2^{16}$). Данные SF-описания были получены следующим образом: двоичные строки в матрице B^f являются значениями с заданной точностью (в двоичной системе счисления) функции $y = \sin x$ на интервале $[0, \pi/2]$, матрицу B^x образуют все полные элементарные конъюнкции, состоящие из n литералов. Для примера S12 ($n = 12$, $m = 12$, $k = 2^{12}$) первая строка матрицы B^x состоит из 12 нулей, последняя – из 12 единиц, матрица B^f – из 12 столбцов. Строго говоря, эти примеры не задают функциональных описаний математической функции $y = \sin x$ на интервале $[0, \pi/2]$, так как в левой части таблиц истинности перечислены не значения аргумента x в двоичном коде, а 2^n полных конъюнкций.

Третий набор примеров был сформирован из псевдослучайных систем ДНФ (табл. 4), сгенерированных программой [23], троичные строки в матрице T^x содержали случайное число $n - p$ значений « \rightarrow », булевы строки в матрице B^f содержали случайное число s единичных значений. При переходе от систем ДНФ к таблицам истинности в булевых матрицах B^f оказывалось много единичных элементов. Например, для примера Pseudo_1 в таблице истинности матрица B^f оказалась на 83 % заполнена единицами, так как сгенерированные элементарные конъюнкции в матрице T^x часто оказывались неортогональными. В практике экспериментальных исследований было установлено, что данные примеры имеют большую сложность при схемной реализации в FPGA, поэтому представлялась целесообразной проверка эффективности логической минимизации для этих примеров.

Таблица 4
Параметры псевдослучайных систем ДНФ

Table 4
Parameters of pseudorandom DNF systems

Пример <i>Example</i>	n	m	k	Среднее число литералов в конъюнкции, p <i>Average number literals in conjunction, p</i>	Среднее число вхождений конъюнкции в ДНФ функций, c <i>Average number occurrences conjunctions in DNF functions, c</i>	Число неопределенных элементов « \leftrightarrow » в матрице T^x , % <i>Number uncertain elements «\leftrightarrow» in matrix T^x, %</i>
Psevdo1	10	20	1 000	9	2	33
Psevdo2	15	20	5 000	14	2	32
Psevdo3	20	20	10 000	19	3	30
Psevdo4	25	20	25 000	24	4	31
Psevdo5	30	20	30 000	29	5	31

Четвертый набор примеров составляли системы функций, описывающие функционирование умножителей (устройств перемножения) и устройств возведения в квадрат целых неотрицательных чисел. Матричные описания систем ДНФ таких устройств были получены в результате перехода от структурных описаний логических схем, синтезированных в библиотечном базисе, путем элиминации (устранения) промежуточных переменных. Особенностью этих примеров является то, что они «частично» минимизированы, т. е. в процессе перехода от задания функций в виде взаимосвязанных логических уравнений к ДНФ были выполнены операции обобщенного склеивания и поглощения [6]. Подобные примеры систем функций появляются, например, при решении задач перепроектирования логических схем из одного базиса логических элементов в другой базис.

Эксперименты без распараллеливания вычислений. Эксперименты проводились на персональном компьютере Lenovo в 64-разрядной ОС Windows7 PC с процессором Intel Core i5-2320 и тактовой частотой 3,00 ГГц, ОЗУ 4 Гб. Программной средой для выполнения экспериментов была система FLC-2, языком описания данных в этой системе является язык SF функционально-структурного описания комбинационных логических схем. В языке SF допускаются иерархические описания, при этом для листовых описаний возможны как матричные описания систем ДНФ, СДНФ, таблиц истинности, так и логические уравнения в булевом базисе.

Эксперимент 1. Сравнение эффективности программ совместной минимизации Espresso, Tie и Minim для решения задачи 1. Исходными данными для большинства примеров в этом эксперименте являлись системы ДНФ, а для некоторых примеров – их частные случаи: системы СДНФ и таблицы истинности. Программа Minim выполнялась с установкой опций (совместной минимизации, итерационного метода).

Далее в таблицах с результатами экспериментов через k_{\min} обозначено число элементарных конъюнкций в кратчайшей системе ДНФ. Лучшие решения помечены символом «*». Символ « \leftrightarrow » означает, что для соответствующего примера программа минимизации не выполнялась. Например, для примера VTХ1 (табл. 5) программа Tie не выполнялась, так как она имеет ограничение $n \leq 22$ для исходных данных. Для некоторых примеров указано время работы программы и пометка «Нет решения», которая означает, что за указанное в таблице время программа минимизации не получила решения.

По результатам эксперимента 1 (табл. 5) можно сделать следующие выводы. При размерностях задачи совместной минимизации $n \leq 20$ лучшие решения получает программа Tie, хотя и затрачивает немного больше времени на решение. При $n > 20$ и большом числе конъюнкций в исходных ДНФ (либо таблицах истинности) лучшие решения получает программа Espresso. Однако при больших размерностях задачи на потоке псевдослучайных примеров конкурентоспособной оказывается программа Minim, которая, например, получает решение для примера Psevdo5 за практически приемлемое время, а программе Espresso за 1,5 ч (5 400 с) не удается

обработать данный пример. Эксперимент 1 позволил установить, что псевдослучайные примеры из третьего набора примеров не только имеют большую сложность при схемной реализации в FPGA, но и являются трудоемкими при совместной минимизации, так как требуют много времени для решения задачи 1, а полученные в результате решения ДНФ содержат много элементарных конъюнкций.

Таблица 5

Результаты эксперимента 1. Сравнение программ совместной минимизации, исходные данные – системы ДНФ

Table 5

Results of experiment 1. Comparison of joint minimization programs, initial data – DNF system

Исходные данные <i>Initial data</i>				Совместная минимизация <i>Joint minimization</i>					
Пример <i>Example</i>	<i>n</i>	<i>m</i>	<i>k</i>	Espresso		Tie		Minim	
				<i>k_{min}</i>	Время, с <i>Time, s</i>	<i>k_{min}</i>	Время, с <i>Time, s</i>	<i>k_{min}</i>	Время, с <i>Time, s</i>
<i>Первый набор примеров</i>									
Z4	7	4	128	*59	0,10	*59	*0,02	*59	0,05
Z5XP1	7	10	128	76	*0,12	*63	4,94	66	0,62
RADD	8	5	120	*75	0,11	*75	*0,02	*75	0,06
ROOT	8	5	256	*57	0,11	*57	*0,02	62	0,07
MLP4	8	8	256	133	*0,13	*121	4,99	135	0,16
MAX512	9	6	512	141	*0,13	*133	4,99	153	0,49
SYM10	10	1	837	*210	*0,14	*210	5,01	*210	0,91
MAX1024	10	6	1 024	276	*0,20	*260	4,99	297	3,43
ADD6	12	7	1 092	*355	*0,16	*355	1,48	*355	1,93
TIAL	14	8	640	581	*0,38	*575	12,0	636	5,89
MP2D	14	14	123	31	*0,13	*30	14,84	36	0,02
INTB	15	7	664	631	*0,48	*629	13,27	635	19,46
B12	15	9	431	42	0,14	*41	19,4	48	*0,09
M181	15	9	430	42	0,14	*41	16,04	48	*0,03
IN0	15	11	138	*107	0,14	*107	6,26	109	*0,13
GARY	15	11	442	*107	*0,17	*107	6,29	128	0,26
B9	16	5	123	*119	0,12	*119	20,01	123	*0,11
IN1	16	17	110	*104	0,15	*104	24,87	105	*0,12
IN2	19	10	137	135	*0,15	*134	489,98	137	0,20
VTX1	27	6	110	*110	0,17	–	–	*110	*0,13
X9DN	27	7	120	*120	0,15	–	–	*120	*0,13
jbp	36	57	166	*122	0,39	–	–	166	*0,13
Too_large_matr	38	3	1 027	*1 027	*0,97	–	–	*1 027	18,21
ibm	48	17	173	*173	1,05	–	–	*173	*0,58
SOAR	83	94	529	*353	*0,83	–	–	427	5,42
X3_matr	135	99	915	*915	*1,26	–	–	*915	18,58
<i>Второй набор примеров</i>									
S12	12	12	4 096	2 523	*7,36	*2 215	4,98	2 784	79,44
S13	13	13	8 192	4 929	*33,15	*4 275	20,01	5 836	279,59
S14	14	14	16 384	*9 672	197,06	10 148	*91,5	11 596	1 253,77
S15	15	15	32 768	*18 768	947,31	19 068	*780,49	Нет решения	5 400,00
S16	16	16	65 536	*36 509	*5 609,26	–	–	–	–
Sist4	17	12	370	266	*0,33	*260	419,94	282	1,45
Verg1	17	61	2 003	482	*1,01	*469	4,95	546	17,72
Verg2	18	63	2 129	528	*1,06	*508	9,95	607	32,39
Sistem8	25	20	45 548	28 668	*2 239,40	–	–	Нет решения	5 400,00
<i>Третий набор примеров</i>									
Psevdo1	10	20	1 000	*509	*2,42	893	5,01	704	13,88
Psevdo2	15	20	5 000	*4 754	*20,13	9 571	653,36	4 770	181,11
Psevdo3	20	20	10 000	*7 639	*376,05	Нет решения	5 400,00	*7 639	666,25
Psevdo4	25	20	25 000	*6 522	*453,84	–	–	*6 522	488,31
Psevdo5	30	20	30 000	Нет решения	5 400,00	–	–	*9 132	*869,39

Эксперимент 2. Сравнение программ Espresso, Tie и Minim совместной минимизации. Исходными данными являлись ортогонализированные системы ДНФ булевых функций. Ортогонализация выполнялась с помощью программы [24]. В ортогонализированной системе ДНФ каждая пара элементарных конъюнкций K_i, K_j является ортогональной: $K_i \& K_j = 0$. Результаты эксперимента приведены в табл. 6, где через k^{ort} обозначено число элементарных конъюнкций в ортогонализированной системе ДНФ.

Таблица 6
Результаты эксперимента 2. Сравнение программ совместной минимизации, исходные данные – ортогонализированные системы ДНФ

Table 6
Results of experiment 2. Comparison of joint minimization programs, initial data – orthogonalized DNF systems

Исходные данные <i>Initial data</i>				Ортогонализация <i>Orthogonalization</i>	Совместная минимизация <i>Joint minimization</i>					
Пример <i>Example</i>	n	m	k		k^{ort}	Espresso		Tie		Minim
				k_{min}		Время, с <i>Time, s</i>	k_{min}	Время, с <i>Time, s</i>	k_{min}	Время, с <i>Time, s</i>
RADD	8	5	120	255	*75	0,12	*75	*0,02	*75	0,07
TIAL	14	8	640	11 299	581	*1,28	*575	11,38	801	106,39
B12	15	9	431	1 606	*41	*0,16	*41	19,31	46	0,85
M181	15	9	430	2 239	*41	*0,17	*41	20,57	47	1,95
IN0	15	11	138	138	*107	*0,15	*107	6,29	109	0,16
GARY	15	11	442	282	*107	0,14	*107	6,29	112	*0,12
B9	16	5	123	12 793	*119	*0,41	*119	19,97	169	111,20
IN1	16	17	110	185	*104	*0,15	*104	25,04	109	*0,15
IN2	19	10	137	1 772	137	*0,17	*134	544,99	148	1,54

Эксперимент 3. Сравнение программ совместной минимизации. В качестве исходных данных использовались таблицы истинности систем булевых функций. Для эквивалентного перехода от систем ДНФ к таблицам истинности в системе FLC-2 имеется соответствующая программа. Результаты приведены в табл. 7.

Таблица 7
Результаты эксперимента 3. Сравнение программ совместной минимизации, исходные данные – таблицы истинности

Table 7
Results of experiment 3. Comparison of joint minimization programs, source data – truth tables

Исходные данные <i>Initial data</i>				Таблица истинности <i>Truth table</i>	Совместная минимизация <i>Joint minimization</i>					
Пример <i>Example</i>	n	m	k		Число наборов <i>Number of sets</i>	Espresso		Tie		Minim
				k_{min}		Время, с <i>Time, s</i>	k_{min}	Время, с <i>Time, s</i>	k_{min}	Время, с <i>Time, s</i>
RADD	8	5	120	256	*75	0,10	*75	0,02	*75	*0,09
TIAL	14	8	640	16 384	582	*1,32	*575	11,42	800	97,74
B12	15	9	431	32 768	*41	*1,25	*41	19,25	47	38,57
M181	15	9	430	32 768	*41	*1,21	*41	20,70	47	41,39
IN0	15	11	138	32 768	*107	*0,59	*107	6,30	120	38,18
GARY	15	11	442	32 768	*107	*0,60	*107	6,29	120	38,18
B9	16	5	123	65 536	*119	*1,39	*119	20,0	254	2 156,8
IN1	16	17	110	65 536	*104	*1,41	*104	25,09	107	14,46
IN2	19	10	137	524 288	137	*19,73	*134	559,53	–	–

Эксперименты 2 и 3 показали, что для программ Espresso и Tie форма исходного задания (ДНФ, ортогонализированные ДНФ, таблицы истинности) не влияет на качество и время реше-

ния. Для программы Minim более предпочтительной формой исходного задания являются ДНФ, а не таблицы истинности.

Эксперимент 4. Сравнение программ раздельной минимизации. Исходными данными являются системы ДНФ булевых функций. В качестве одного из параметров для исходных данных указывается $\sum_{i=1}^m k_i$ – суммарное число элементарных конъюнкций в исходной (неминимизированной) системе ДНФ, через $\sum_{i=1}^m k_{\min}^i$ (в табл. 8 и далее) обозначено суммарное число элементарных конъюнкций в раздельно минимизированных ДНФ.

Программа Minim выполнялась с установкой опций (раздельной минимизации, итерационного метода). Программа ABC в результате своей работы выдавала значение параметра Node (число узлов в совместной BDD функций). Применение программы Tie совместной минимизации для выполнения раздельной минимизации осуществлялось следующим образом. По исходному SF-описанию системы функций строилась логическая сеть, блоками которой являлись компонентные функции системы. Затем последовательно выполнялась минимизация каждого блока по отдельности с помощью программы Tie и подсчитывалось суммарное время на обработку всех блоков сети. Последовательная обработка листовых описаний логической сети проводилась с помощью стратегии «Минимизировать листовые SF-описания» [22]. Результаты эксперимента приведены в табл. 8.

Эксперимент 4 позволил установить тот факт, что программа ABC раздельной минимизации является наиболее быстродействующей, она проиграла программе Minim по быстродействию только в одном случае (см. пример Psevdo5, табл. 8). Однако программа ABC проигрывает по качеству решений программам Tie и Minim. Как и ожидалось, для диапазона $n \leq 20$ лучшие решения получила программа Tie совместной минимизации, используемая в данном эксперименте для нахождения кратчайших ДНФ отдельных функций.

Эксперимент 5. Проверка эффективности программы совместной минимизации Espresso после выполнения программы ABC раздельной минимизации. Результаты эксперимента приведены в табл. 9.

Основываясь на данных табл. 9, можно утверждать, что качество решений задачи совместной минимизации программой Espresso чаще всего можно улучшить, если предварительно провести раздельную минимизацию программой ABC. Тем не менее время вычислений может существенно возрасти для псевдослучайных примеров Psevdo3, Psevdo4 (табл. 9), хотя качество решений, получаемых в данных случаях программой Espresso, не меняется. Вместе с тем закономерность такого значительного увеличения времени выявить не удалось.

Эксперимент 6. Сравнение эффективности применения программы ABC для компонентных функций по отдельности и для системы функций в целом. Эксперимент проводился на примере Psevdo5, обработка которого заняла значительное время (табл. 9). Результаты эксперимента показали, что разбиение исходной системы функций на подсистемы, содержащие по одной функции, позволяет значительно сократить время работы программы ABC. Обработка компонентных функций по отдельности привела к получению суммарного числа узлов раздельных BDD для 20 функций системы – 3 934 961 (для системы в целом в совместной BDD было 3 357 956 узлов). Суммарное время минимизации 20 подсистем составило 233,36 с, время обработки системы в целом – 4 669,62 с (табл. 9). В итоге получили $\sum_{i=1}^m k_i = 45\,809$. Это тот же

результат, что и при минимизации системы в целом. Разбиение системы на «одиочные» функции позволило сократить время в 20 раз – по числу функций системы. Таким образом, если система функций содержит достаточно много компонентных функций и размерность задачи раздельной минимизации велика: $n > 30$, $k > 30\,000$, то целесообразно разбить исходную систему на подсистемы (например, содержащие по одной компонентной функции) и выполнить раздельную минимизацию программой ABC для подсистем.

Таблица 8
Результаты эксперимента 4. Сравнение программ раздельной минимизации

Table 8
Results of experiment 4. Comparison of separate minimization programs

Исходные данные <i>Initial data</i>					Раздельная минимизация <i>Separate minimization</i>							
					ABC			Minim			Разбиение системы функций на блоки и Tie <i>Splitting the system of functions into blocks and Tie</i>	
					Моноблок – система функций <i>Monoblock – function system</i>			Моноблок – система функций <i>Monoblock – function system</i>			Сеть компонентных функций <i>Network of Component Functions</i>	
Пример <i>Example</i>	<i>n</i>	<i>m</i>	<i>k</i>	$\sum_{i=1}^m k_i$	$\sum_{i=1}^m k_{\min}^i$	Node	Время, с <i>Time, s</i>	<i>k</i> _{общ}	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>
<i>Первый набор примеров</i>												
Z4	7	4	128	256	*59	26	*0,01	59	*59	0,03	*59	1,01
Z5XP1	7	10	128	576	*74	50	*0,01	71	*74	0,04	*74	2,44
RADD	8	5	120	120	*75	34	*0,01	75	*75	0,04	*75	1,01
ROOT	8	5	256	615	75	68	*0,01	71	*71	0,04	*71	2,10
MLP4	8	8	256	678	145	142	*0,01	143	*143	0,06	*143	1,81
MAX512	9	6	512	1 616	183	155	*0,01	171	172	0,13	*164	6,38
SYM10	10	1	837	837	240	43	*0,006	210	*210	0,67	*210	5,24
MAX1024	10	6	1 024	3 232	364	254	*0,01	328	*330	0,73	629	16,66
ADD6	12	7	1 092	1 092	*355	65	*0,01	355	*355	1,26	*355	1,53
TIAL	14	8	640	644	653	867	*0,01	639	643	0,76	*631	5,66
MP2D	14	14	123	204	*76	122	*0,005	38	*76	0,05	*76	6,51
INTB	15	7	664	664	635	633	*0,01	635	635	1,32	*629	3,65
B12	15	9	431	454	*53	81	*0,005	48	*53	0,03	*53	4,78
M181	15	9	430	453	*53	81	*0,01	48	*53	0,02	*53	4,87
IN0	15	11	138	487	203	391	*0,01	166	193	0,08	*191	7,61
GARY	15	11	442	442	203	391	*0,01	164	*191	0,05	*191	7,54
B9	16	5	123	123	156	85	*0,01	119	*119	0,06	*119	7,21
IN1	16	17	110	1 074	727	567	*0,01	273	*698	0,42	*698	17,59
IN2	19	10	137	310	239	290	*0,01	201	*231	0,10	*230	166,14
VTX1	27	6	110	110	*110	182	*0,01	110	*110	0,04	–	–
X9DN	27	7	120	120	*120	182	*0,01	120	*120	0,05	–	–

Окончание табл. 8

End of table 8

Исходные данные <i>Initial data</i>					Раздельная минимизация <i>Separate minimization</i>							
					ABC			Minim			Разбиение системы функций на блоки и Tie <i>Splitting the system of functions into blocks and Tie</i>	
					Моноблок – система функций <i>Monoblock – function system</i>			Моноблок – система функций <i>Monoblock – function system</i>			Сеть компонентных функций <i>Network of Component Functions</i>	
Пример <i>Example</i>	n	m	k	$\sum_{i=1}^m k_i$	$\sum_{i=1}^m k_{\min}^i$	Node	Время, с <i>Time, s</i>	$k_{\text{общ}}$	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>
jbp	36	57	166	189	190	437	*0,01	166	*189	0,02	–	–
Too_large_matr	38	3	1 027	1 056	*1 056	1 050	*0,05	1 027	*1 056	7,79	–	–
ibm	48	17	173	173	*173	471	*0,02	173	*173	0,07	–	–
SOAR	83	94	529	529	500	551	*0,02	433	*486	0,32	–	–
X3_matr	135	99	915	997	*997	851	*0,03	915	*997	2,76	–	–
<i>Второй набор примеров</i>												
S12	12	12	4 096	27 238	3 421	1 758	*0,02	2916	2 944	72,01	*2 770	43,34
S13	13	13	8 192	58 616	6 604	3 033	*0,06	5 540	5 584	278,94	*5 192	44,02
S14	14	14	16 384	125 497	12 859	5 136	*0,10	11 236	11 300	490,88	*9 859	50,63
S15	15	15	32 768	267 551	25 186	8 862	*0,24	25 671	25 792	1228,56	*18 830	63,06
Sin16	16	16	65 536	568 043	49 382	15 005	*0,48	52 853	53 034	4 897,58	*36 329	89,26
Sist4	17	12	370	373	326	407	*0,01	284	285	0,21	*284	19,07
Verg1	17	61	2 003	97 127	9 334	1 685	*0,01	1 044	8 686	25,36	*8 646	66,38
Verg2	18	63	2 129	107 281	7 652	1 680	*0,01	1 031	7 548	22,09	*7 334	180,98
Sistem8	25	20	45 548	45 947	37 035	14 435	*0,40	36 870	*36 883	2979,90	–	–
<i>Третий набор примеров</i>												
Psevdo1	10	20	1 000	2 103	1 248	1 629	*0,01	965	1 059	1,35	*977	81,32
Psevdo2	15	20	5 000	10 676	15 284	45 857	*0,57	6 724	9 604	100,58	*9 571	103,70
Psevdo3	20	20	10 000	26 657	20 775	534 629	*77,38	8 408	*20 320	476,15	*20 320	1 788,42
Psevdo4	25	20	25 000	92 300	23 939	1 278 016	*341,90	6 606	*23 938	566,25	–	–
Psevdo5	30	20	30 000	150 620	*45 809	3 357 956	4 669,62	9 140	45 899	*1 641,27	–	–

Таблица 9
Результаты эксперимента 5. Совместная минимизация программой Espresso
после выполнения раздельной минимизации программой ABC

Table 9
Results of experiment 5. Joint minimization by Espresso program
after performing separate minimization by ABC program

Пример Example	Исходное задание Initial task	Раздельная минимизация Separate minimization		Совместная минимизация Joint minimization			
		ABC		Espresso после ABC Espresso after ABC		Espresso от исходных описаний Espresso from original descriptions	
	$\sum_{i=1}^m k_i$	$\sum_{i=1}^m k_i^{ABC}$	Время, с Time, s	$\sum_{i=1}^m k_i^{Esp}$	Время, с Time, s	$\sum_{i=1}^m k_i^{Esp}$	Время, с Time, s
S12	27 238	3 421	0,01	*2 442	*5,39	2 523	7,36
S13	58 616	6 604	0,06	*4 667	37,07	4 929	*33,15
S14	125 497	12 859	0,10	*9 080	*125,40	9 672	197,06
S15	267 551	25 186	0,24	*17 441	*621,56	18 768	947,31
S16	568 043	49 382	0,48	*33 666	*3 060,86	36 509	5 609,26
Sistem8	45 947	37 035	0,40	*28 575	2 246,04	28 668	*2 239,40
Psevdo1	2 103	1 248	0,01	522	*1,44	*509	2,42
Psevdo2	10 676	15 284	0,57	4 806	59,16	*4 754	*20,13
Psevdo3	26 657	20 775	77,38	*7 639	1 621,76	*7 639	*376,05
Psevdo4	92 300	23 939	341,90	*6 522	18 572,60	*6 522	*453,84
Psevdo5	150 620	45 809	4 669,62	–	–	–	–

Эксперимент 7. Сравнение длин ДНФ, найденных с использованием BDD, с длинами кратчайших ДНФ функций. Результаты эксперимента приведены в табл. 10. По BDD, полученным программой OPT_BDD [24], строилась система ДНФ с помощью программы элиминации промежуточных переменных [25], которая входит в систему FLC-2. В данной программе в процессе получения ДНФ функций выполняются операции поглощения конъюнкций, а для результирующей ДНФ – операции обобщенного склеивания и поглощения конъюнкций.

В итоге строится матричная форма системы ДНФ булевых функций, которая содержит k^{BDD} общих элементарных конъюнкций. Суммарное число конъюнкций в ДНФ, построенных по BDD, обозначено через $\sum_{i=1}^m k_i^{BDD}$, где k_i^{BDD} – число элементарных конъюнкций в ДНФ i -й функции

системы, m – число функций системы. Исходное задание системы функций ранее минимизировалось программами раздельной минимизации. Результаты минимизации приведены в табл. 9, их можно сравнить с данными из табл. 10.

По эксперименту 7 можно сделать следующие выводы: переход от BDD к ДНФ меняет форму задания функций, подвергаемых минимизации. Поэтому в результате могут быть получены как лучшие, так и худшие решения по сравнению с исходными системам ДНФ. Если у пользователя нет программы минимизации функций в классе ДНФ, то он может воспользоваться программой минимизации BDD, после чего перейти к ДНФ. Это будет целесообразно, если $k^{BDD} \leq k$. Однако в общем случае такой подход неэффективен, так как зачастую k^{BDD} будет значительно больше числа k конъюнкций в исходном задании функций. Такая ситуация объясняется тем, что BDD задают ортогонализированные формы компонентных функций, а использование при элиминации простых преобразований ДНФ (склеивания и поглощения конъюнкций) не позволяет провести качественную минимизацию в классе ДНФ ортогонализированных форм, получаемых по BDD-представлениям.

Таблица 10

Результаты эксперимента 7. Раздельная минимизация по BDD-представлению системы функций

Table 10

Results of experiment 7. Separate minimization by BDD-representation of a system of functions

Исходные данные <i>Initial data</i>					BDD-минимизация и элиминация <i>BDD-minimization and elimination</i>	
Пример <i>Example</i>	n	m	k	$\sum_{i=1}^m k_i$	k^{BDD}	$\sum_{i=1}^m k_i^{BDD}$
Z4	7	4	128	256	59	*59
Z5XP1	7	10	128	576	79	*82
RADD	8	5	120	120	75	*75
ROOT	8	5	256	615	69	*79
MLP4	8	8	256	678	195	*226
MAX512	9	6	512	1 616	221	*236
SYM10	10	1	837	837	240	*240
MAX1024	10	6	1 024	3 232	432	*480
ADD6	12	7	1 092	1 092	651	*651
TIAL	14	8	640	*644	2 702	2 798
MP2D	14	14	123	204	48	*85
INTB	15	7	664	*664	1 785	1 791
B12	15	9	431	454	57	*63
M181	15	9	430	453	58	*64
IN0	15	11	138	487	206	*262
GARY	15	11	442	442	206	*262
B9	16	5	123	*123	249	249
IN1	16	17	110	*1 074	537	1 791
IN2	19	10	137	*310	566	659
VTX1	27	6	110	*110	612	612
X9DN	27	7	120	*120	1 818	1 818
jbp	36	57	166	*189	574	638
ibm	48	17	173	*173	1 380	1 380
SOAR	83	94	529	*529	1 112	1 182
X3_matr	135	99	915	*997	17 205	17 312
S12	12	12	4 096	27 238	5 035	*6 048
Sist4	17	12	370	*373	946	975
Verg1	17	61	2 003	97 127	1 837	*16 206
Verg2	18	63	2 129	107 281	2 473	*17 527

Многопоточное программирование и стандарт OpenMP. Разработка параллельных алгоритмов [26] и проектирование могопоточных программ представляют собой довольно сложные процессы и требуют использования стандартов и технологий параллельного программирования, которые позволяют упростить работу по организации параллельных вычислений. В настоящее время для организации параллельных вычислений был использован стандарт OpenMP. Этот стандарт реализован в языках программирования C/C++ и Fortran, работает как в Unix-, так и в Windows-системах. Технология OpenMP представляет собой набор особых директив компилятора, процедур и переменных окружения (URL: <http://www.openmp.org/>) [27, 28]. Процесс использования OpenMP сводится к вставке директив в места программы, где код может быть распараллелен [29].

Проектирование параллельной программы сводится к поэтапному выполнению таких действий, как разделение общих задач на подзадачи, выявление информационных зависимостей у подзадач, масштабирование подзадач и распределение по вычислительным элементам. (Для систем с общей памятью роль вычислительных элементов выполняют ядра процессора.)

В общем случае параллельные программы – программы с несколькими исполняемыми потоками или процессами – могут работать как на одном компьютере (многоядерной системе), так и на мультипроцессорной системе (на кластере или по сети). В последних двух случаях для синхронизации используется передача сообщений.

Поскольку выполняемая программа образует процесс, то возможна реализация параллельной программы с использованием двух подходов:

1. Программа, использующая несколько потоков в рамках одного процесса, может быть выполнена быстрее за счет параллельного выполнения ее отдельных частей (распределенных по потокам). Общаются потоки либо с помощью общей памяти, либо обмениваясь сообщениями.

2. Программа, использующая нескольких потоков в рамках нескольких процессов, тоже может быть реализована быстрее за счет параллельного выполнения отдельных ее процессов. Такая организация процесса вычислений эффективна, если потокам почти не надо общаться, когда много потоков выполняют операции параллельно и сводят свои результаты в один ответ только тогда, когда все потоки посчитали свои части.

Первый способ распараллеливания вычислений при раздельной минимизации булевых функций. Распараллеливание алгоритма раздельной минимизации системы функций, когда каждая из функций системы минимизируется независимо от других, может быть сравнительно просто реализовано на системе с общей памятью [27]. При параллельной реализации алгоритма раздельной минимизации систем булевых функций использовались современные многопоточные расширения C++ [26] и директивы OpenMP. Основным объектом распараллеливания (по числу функций m) является программный цикл, поскольку на циклических участках программы сосредоточены большие объемы вычислений. В частности, выполняется минимизация компонентной булевой функции f_i .

Таким образом, достаточно разбить исходную систему ДНФ булевых функций на m подсистем, содержащих по одной функции, после чего распределить подсистемы между m параллельными регионами и получить от них минимизированные ДНФ D_{f_i} . При этом возможно близкое (с учетом закона Амдала) к m -кратному увеличению быстродействия параллельной реализации алгоритма без потери качества на вычислительной системе по сравнению с последовательной реализацией. Основная часть алгоритма минимизации для компонентной функции, реализованного в работе [18], не меняется. Распараллеливание проводится на уровне обхода m подсистем для компонентных функций, на которые разбита исходная система ДНФ и для которых по отдельности выполняется алгоритм минимизации для компонентных функций.

Исходная система ДНФ полностью определенных булевых функций задается объектом `Sop1`. Минимизированная система ДНФ представляется объектом `ResSop`. Основой алгоритма представленной в листинге 1 функции `MiOi` (`Limit`, `Crit`, `Tm1`, `Tm`) является проведение минимизации в классе ДНФ одной полностью определенной булевой функции, заданной объектом `Tm1` (троичная матрица). Результат минимизации (троичная матрица) сохраняется в объекте `Tm`. Результат минимизации (троичная матрица) сохраняется в объекте `ResTm`. В функции `MiRi` качество решения управляется параметрами `Limit` и `Crit`, задание которых стабилизирует качество получаемой системы ДНФ.

Работа параллельной реализации последовательного алгоритма [18] начинается с единственного потока – основного. После размещения OpenMP-директивы (строка 6) образуются параллельные регионы, входя в которые, основной поток создает группы потоков (включающие основной поток). В конце параллельного региона группы потоков останавливаются, а выполнение основного потока продолжается (строка 26). Далее внутри отдельного параллельного региона посредством размещения OpenMP-директивы `task` (строка 8) в текущем потоке выделяется отдельная независимая задача (так называемый легковесный поток – нить), с которой ассоциирован блок операторов. Таким образом, задача (`task`) помещается внутрь параллельной области и задает блок операторов, который может выполняться в отдельном потоке. Однако задача (`task`) не образует новый поток, а помещается в пул, из которого ее может взять для выполнения один из свободных потоков. Задача может выполняться немедленно после создания или быть отложенной на неопределенное время и выполняться по частям. Размер таких частей

определяется размером фрагмента блока операторов, а порядок выполнения частей разных текущих или отложенных задач определяется системным планировщиком задач и условиями синхронизации. Динамическое распределение задач по потокам осуществляется алгоритмами планирования типа work stealing [28].

Листинг 1. Псевдокод параллельного алгоритма раздельной минимизации (первый способ распараллеливания)

```

1  MiRi (Limit, Crit, CSOP &Sop1, CSOP & ResSop )
2  {
3      n=Sop1.foo1(), m=Sop1.foo2(), k=Sop1.foo3();
        // n – число входов, m – число выходов, k – число конъюнкций
4      CSOP SopR(...); // создать объект для ДНФ  $D_{f_i}$ 
5      num_core= omp_get_num_procs (); // определить число ядер
6      #pragma omp parallel for shared(Limit, Crit, Sop1, SopR, n, m)
        num_threads(num_core)
7      for (i = 0 ; i < m ; i++) {
8      #pragma omp task untied firstprivate(i) shared(Sop1, k, SopR, n, m)
9      {
10         CTM Tm, Tm1; // создать два объекта: Tm, Tm1 – троичные матрицы
11         CSOP SopRR(...); // в каждой task объект для ДНФ  $D_{f_i}$ 
12         .....
13         Tm1=foo4(...);
14         .....
15         if(Tm1.foo5())
16         { // минимизация одной полностью определенной функции
17             MiOi (Limit, Crit, Tm1, Tm);
18             for (j = 0; j < Tm.foo4(); j++) {
19                 {SopRR.foo5(Tm.foo6(j),...);}
20             #pragma omp critical
21             {
22                 SopR.foo7(...SopRR);
23             } // конец критической секции
24         } // конец проверки условия в 15-й строке
25     } // конец кода в блоке task
26 } // конец кода в блоке parallel
27 ResSop.foo8(...SopR); // результирующая система ДНФ
28 }
```

Введение дополнительной опции untied для OpenMP-директивы task означает, что отложенная задача может быть продолжена любым потоком, выполняющимся в параллельной области. Если дополнительная опция не указана, то задача может быть продолжена только породившим ее потоком. С помощью OpenMP-директивы critical (строка 20) оформляется критическая секция программы, в которой собираются минимизированные ДНФ компонентной функции, полученные отдельными задачами. В каждый момент времени в критической секции может находиться не более одной задачи. Все другие задачи будут заблокированы, пока вошедшая задача не закончит выполнение. Как только вошедший в критическую секцию поток выйдет из нее, один из заблокированных потоков туда войдет. Если на входе стоит несколько потоков, то случайным образом выбирается один из них, а остальные продолжают ожидание. Естественно, введение критических секций снижает степень параллелизма. Задачи (tasks) являются альтернативой традиционным потокам (threads), позволяющей реализовать лучший баланс нагрузки, эффективное использование имеющихся ресурсов и высокий уровень абстракции, благодаря которому можно не заботиться о непосредственном управлении параллельным исполнением [28]. Вхождение той или иной задачи в критическую секцию, где формируется объединение ДНФ текущих решений, определяется скоростью нахождения минимальной ДНФ i -й компонентной функции.

Программа Minipar отдельной минимизации является модификацией программы Minim и использует параллельные вычисления в рамках первого подхода. Исходными данными для программы служит система ДНФ, представленная одним блоком в формате SDF языка SF.

Второй способ распараллеливания вычислений при отдельной минимизации булевых функций. Суть организации параллельного выполнения, основанного на использовании средств современных ОС, состоит в следующем. Процесс – это экземпляр программы, загруженной в память компьютера для выполнения. Когда программа производит запуск другой программы, ОС всегда создает новый процесс, т. е. процесс создает дочерний процесс. Средства создания нового процесса полезны для использования функциональных возможностей программ, не имеющих графического интерфейса и работающих с командной строкой.

Для среды разработки Qt [29] процессы можно создавать с помощью класса QProcess, который определен в заголовочном файле QProcess. Объекты этого класса в состоянии считывать информацию, выводимую запущенными процессами, и даже подтверждать их запросы на ввод информации. Этот класс содержит методы для манипулирования системными переменными процесса. Работа с объектами класса QProcess осуществляется в асинхронном режиме, что позволяет сохранять работоспособность графического интерфейса программы-источника в моменты, когда запущенные процессы функционируют. При появлении данных или других событий объекты класса QProcess посылают сигналы. Например, при возникновении ошибок объект процесса вышлет сигнал error() с кодом этой ошибки.

Для создания процесса его нужно запустить. Запуск процесса выполняется методом start(), в который необходимо передать имя команды и список ее аргументов либо все вместе – команду и аргументы одной строкой. В поле аргументов метода start() может быть введена любая команда, соответствующая ОС. Как только процесс будет запущен, высылается сигнал started(), а после завершения его работы – сигнал finished(). Вместе с сигналом finished() высылается код и статус завершения работы процесса. Для чтения данных запущенного процесса класс QProcess предоставляет два разделенных канала: канал стандартного вывода (stdout) и канал ошибок (stderr). Считывать и записывать данные в процесс можно с помощью методов класса QIODevice::write(), read(), readLine() и getChar(). Также для чтения можно воспользоваться методами, привязанными к конкретным каналам: readAllStandardOutput() и readAllStandardError().

В рамках второго подхода параллельные вычисления реализует программа StartParBlock (листинг 2).

Листинг 2. Псевдокод алгоритма отдельной минимизации, реализующего блочно-параллельный подход (второй способ распараллеливания)

```
// n – число входов, num – число выходов, k – число конъюнкций,
// num_core – число ядер
1  #pragma omp parallel for firstprivate (...) shared(..., num)
   num_threads(num_core)
2  for (i=1; i<=num; i++) // цикл по числу блоков (num)
3  {
4     #pragma omp task untied firstprivate(i, ...) shared (....., num)
5     {
6     .....
7     QProcess* m_process = new QProcess();
      // образование объекта – нового Qt-процесса
8     m_process->setProcessChannelMode(QProcess::MergedChannels);
      // образование канала для вывода протоколов i-го процесса
9     m_process->setStandardOutputFile("Qt_protocol.log");
      // Qt_protocol.log – протокол i-го блока
10    QString strCommand = NameProg;
      // NameProg – имя программы минимизации
      // основная часть командной строки для запуска дочернего i-го процесса
11    QString strCommand1;
      // объект для основной части команды запуска программы
```

```

12     strCommand1+="-i"+W_in.data(); // исходный sf-файл i-го блока
13     strCommand1+="-o"+W_out.data();
    // sf-файл - результат минимизации i-го блока;
14     strCommand+=strCommand1;
    // основная часть строки запуска процесса минимизации
15     .....
    // формирование командной строки для запуска программ минимизации
16     if (NameProg=="Minim") // для программы Minim
17     {
18         strCommand +="-r"+ config.ini";
19     }
20     if (NameProg=="MinimPar") // для программы MinimPar
21     {
22         strCommand +="-r"+ config.ini";
23     }
24     if (NameProg=="Espresso") // для программы Espresso
25     {
26         ...// подготовка дополнительных параметров программы Espresso
27         strCommand +="-f"+"Probe.prb";
28     }
29     m_process->start(strCommand); // запуск i-го процесса
30     m_process->waitForFinished(-1);
31     rc = m_process->exitCode();
32     } // конец отр-директивы task
33 } // конец цикла по числу блоков (Qt-процессов)
34 foo1 (.....); // собрать время всех Qt-процессов в один протокол
35 foo2 (.....);
    // собрать сеть с результатами минимизации для всех Qt-процессов

```

Программа StartParBlock организует запуск в параллельном режиме процессов, являющихся контейнерами для исполняемых модулей различных программ минимизации. В листинге 2 приведены команды запуска для программ минимизации Minim, MinimPar и Espresso. Описанием входных данных для программы StartParBlock является многоуровневое представление системы булевых функций в виде иерархического двухуровневого SF-описания в формате 2-CONNECT [22]. Каждый листовый блок представляет отдельную выделенную подсистему булевых функций. Программа StartParBlock подготавливает элементы исходного описания в отдельные блоки с автономными SF-описаниями в формате SDF. Именно с SDF-форматом работают исполняемые модули программ Minim, MinimPar, Espresso и др. Программа StartParBlock подготавливает также командную строку для запуска соответствующей программы минимизации и организует параллельную работу процессов-контейнеров.

Реализация программ MinimPar и StartParBlock осуществлена на языке программирования C++(11.0), реализация директив и библиотек OpenMP (версии 3.1) – в среде Qt [29] с компилятором Mingw (версии 5.7.0). Эксперименты для программ минимизации с распараллеливанием вычислений выполнялись в 64-разрядной операционной системе Windows7 на четырехъядерных процессорах Intel Core 7-4820K с тактовой частотой 3,7 ГГц, ОЗУ 32 ГБ с включением режима гипертрейдинга.

Эксперименты с распараллеливанием вычислений

Эксперимент 8. Распараллеливание вычислений при отдельной минимизации с использованием программы Espresso. По исходному SF-описанию системы функций строилась логическая сеть в формате 2-CONNECT, блоками которой являлись компонентные функции системы. Данная сеть подавалась на вход программы StartParBlock, которая запускала программу Espresso для каждого листового блока в двух режимах: последовательной и параллельной обработки листовых описаний. Результаты эксперимента 8 представлены в табл. 11.

Таблица 11
Результаты эксперимента 8. Распараллеливание вычислений
при раздельной минимизации с использованием программы Espresso

Table 11
Results of experiment 8. Parallelization of calculations
with separate minimization using the Espresso program

Исходные данные <i>Initial data</i>					Раздельная минимизация <i>Separate minimization</i>			
					Сеть компонентных функций <i>Network of Component Functions</i>		Сеть компонентных функций <i>Network of Component Functions</i>	
					Последовательная обработка m блоков сети программой Espresso <i>Sequential processing of m network blocks by the Espresso program</i>		Параллельная обработка m блоков сети программой Espresso <i>Parallel processing of m network blocks by Espresso program</i>	
Пример <i>Example</i>	n	m	k	$\sum_{i=1}^m k_i$	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>
<i>Второй набор примеров</i>								
S12	12	12	4 096	27 238	2 921	2,34	2 921	*0,69
S13	13	13	8 192	58 616	5 548	4,14	5 548	*1,14
S14	14	14	16 384	125 497	10 572	11,65	10 572	*2,78
S15	15	15	32 768	267 551	20 150	45,46	20 150	*11,01
S16	16	16	65 536	568 043	38 636	188,46	38 636	*44,11
System8	25	20	45 548	45 947	32 378	143,64	32 378	*39,37
<i>Третий набор примеров</i>								
Psevd01	10	20	1 000	2 103	1 011	2,20	1 011	*0,69
Psevd02	15	20	5 000	10 676	9 574	4,57	9 574	*1,24
Psevd03	20	20	10 000	26 657	20 320	33,52	20 320	*8,37
Psevd04	25	20	25 000	92 300	23 938	198,15	23 938	*56,36
Psevd05	30	20	30 000	150 620	45 809	941,62	45 809	*286,00
<i>Четвертый набор примеров</i>								
mult_7_DNF	14	14	13 060	13 252	8 133	7,76	8 133	*1,79
mult_7_TABL	14	14	16 384	93 334	8 102	10,57	8 102	*2,18
mult_8_DNF	16	16	52 810	53 621	30 600	102,59	30 600	*29,26
mult_8_TABL	16	16	65 536	434 660	30 538	169,64	30 538	*39,69
mult_9_DNF	18	18	208 598	212 047	113 128	2 999,20	113 128	*1 183,30
mult_9_TABL	18	18	262 144	1 302 835	112 850	4 692,82	112 850	*1 514,90
square_12_DNF	12	24	5 612	5 902	3 350	3,40	3 350	*0,71
square_12_TABL	12	24	4 096	39 962	3 326	3,69	3 326	*0,84
square_13_DNF	13	26	11 259	11 904	6 501	5,32	6 501	*1,11
square_13_TABL	13	26	8 191	87 928	6 455	6,28	6 455	*1,67
square_14_DNF	14	28	22 867	24 151	12 452	10,91	12 452	*2,71
square_14_TABL	14	28	16 384	192 100	12 401	15,25	12 401	*3,68
square_15_DNF	15	30	45 568	48 045	23 841	31,91	23 841	*8,44
square_15_TABL	15	30	32 768	416 811	23 711	52,62	23 711	*12,27

Эксперимент 9. Распараллеливание вычислений при отдельной минимизации с использованием программ *Minim*, *MinimPar*. Результаты эксперимента представлены в табл. 12. Суммарные длины ДНФ для минимизированных систем функций часто различаются, однако все результаты минимизации являются корректными решениями. Все решения были проверены, и установлена их функциональная эквивалентность с исходными заданиями на минимизацию. Различия в решениях (суммарных длинах ДНФ) объясняются тем, что программа *Minim* реализует макроалгоритм [18], который в зависимости от сочетания различных параметров может выполнять процедуры минимизации в разной последовательности из-за того, что исходные задания для программ (табл. 12) различаются. Для моноблока – это пара матриц, для сети – иерархические описания, в которых описания отдельных функций являются листовыми.

Эффективность реализации распараллеливания первым и вторым способами примерно одинакова. Так, второй способ распараллеливания немного выигрывает по быстродействию у первого. Вместе с тем результаты отдельной минимизации, получаемые первым способом, лучше. Естественно, что по сравнению с *Minim* параллельный вариант *MinimPar* является более быстродействующим: время работы этой программы примерно в пять раз меньше.

Эксперимент 10. Сравнение эффективности и трудоемкости программ совместной минимизации *Espresso* и *Minim*. Результаты эксперимента представлены в табл. 13.

Эксперимент 10 показывает значительное преимущество программы *Espresso* по сравнению с программой *Minim* при решении задачи совместной минимизации как по качеству, так и по времени решения. Только в одном случае (пример *Pseudo5*) программа *Espresso* не смогла получить решение, проработав 7,5 ч (27 000 с). Обе сравниваемые в табл. 13 программы быстрее обрабатывают исходные системы ДНФ (*_DNF*), чем таблицы истинности (*_TABL*).

Отметим также то, что предварительная совместная минимизация функций в классе ДНФ позволяет сокращать время BDD-оптимизации. Например, выполняя BDD-оптимизацию с помощью программы *BDD-Builder*, входящей в систему *FLC-2*, время получения BDD-представления для примера *mult_9_DNF* сокращается в шесть раз по сравнению с получением того же BDD-представления, когда в качестве исходных данных выступает таблица истинности *mult_9_TABL*.

Сравнение результатов экспериментальных исследований. Результаты экспериментов для различных модификаций программы *Espresso* (-II, -IIC, -MV, -Exact) опубликованы в работе [12]. Однако в этой работе, а также в работе [18] приведено лишь общее время решения всех 134 примеров и не представлено время обработки для отдельных примеров. Указывается, что программа *Espresso-Exact* нашла точное решение для 104 примеров.

В публикации [30] описывается метод минимизации булевых функций с большим числом импликант (более миллиона), использующий представление множества импликант в виде трюичного дерева, на котором выполняются операции слияния листьев. Процедура слияния листьев соответствует слиянию двоичных наборов из таблицы истинности, которые отличаются только одной последней переменной, чем достигается ускорение выполнения операций. Предложенный алгоритм реализован в программе *TT-Min*. Приводятся результаты экспериментальных исследований на стандартных примерах *Benchmark* в сравнении с программой *Espresso*. Показано, что программа *TT-Min* проигрывает программе *Espresso* по качеству получаемого решения, но если ее использовать для предварительной обработки функции, то совместное время (*TT-Min+Espresso*) может оказаться лучше, чем время, затраченное только программой *Espresso*. Целесообразно также использовать программу *TT-Min* для функций, порождающих большое число импликант.

Программа *BOOM* [31, 32] эффективна для функций с несколькими сотнями входных переменных, значения которых определены лишь на небольшой части булева пространства. Эксперименты показали, что она дает лучшие результаты для больших схем, чем программа *Espresso*. Программа *BOOM-II* [33] объединяет две программы минимизации: *BOOM* и *FC-Min* [34]. Для программы *BOOM-II* проведено экспериментальное сравнение с программой *Espresso*.

Таблица 12

Результаты эксперимента 9. Распараллеливание вычислений при раздельной минимизации с использованием программ Minim и MinimPar

Table 12

Results of experiment 9. Parallelization of calculations with separate minimization using Minim and MinimPar programs

Исходные данные <i>Initial data</i>					Моноблок <i>Monoblock</i>		Сеть <i>Net</i>			
					Первый способ распараллеливания MinimPar <i>The first way to parallelize MinimPar</i>		Последовательная обработка m блоков сети программой Minim <i>Sequential processing of m network blocks by the Minim program</i>		Второй способ распараллеливания. Параллельная обработка m блоков сети программой MinimPar <i>The second way of parallelization. Parallel processing of m network blocks by the MinimPar program</i>	
Пример <i>Example</i>	n	m	k	$\sum_{i=1}^m k_i$	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>	$\sum_{i=1}^m k_{\min}^i$	Время, с <i>Time, s</i>
<i>Второй набор примеров</i>										
S12	12	12	4 096	27 238	*2 916	19,9	2 944	67,57	2 944	*19,17
S13	13	13	8 192	58 616	*5 613	39,47	5 653	167,65	5 655	*36,51
S14	14	14	16 384	125 497	12 737	81,16	*11 896	340,63	12 767	*58,31
S15	15	15	32 768	267 551	*26 694	334,20	26 737	1 064,68	26 818	*274,52
S16	16	16	65 536	568 043	53 521	1 819,80	*53 420	4 449,4	53 694	*1 023,59
System8	25	20	45 548	45 947	37 070	759,30	*36 949	2 664,47	37 083	*700,19
<i>Третий набор примеров</i>										
Psevdo1	10	20	1 000	2 103	1 059	*0,40	1 059	1,42	1 059	0,43
Psevdo2	15	20	5 000	10 676	9 604	22,44	9 604	97,83	9 604	*17,41
Psevdo3	20	20	10 000	26 657	20 320	100,26	20 320	420,75	20 320	*79,87
Psevdo4	25	20	25 000	92 300	23 938	100,70	23 938	447,22	23 938	*81,50
Psevdo5	30	20	30 000	150 620	45 809	244,92	45 809	1 517,54	45 809	*201,13
<i>Четвертый набор примеров</i>										
mult_7_DNF	14	14	13 060	13 252	8 268	84,21	8 298	348,113	8 268	*78,84
mult_7_TABL	14	14	16 384	93 334	8 289	94,63	8 255	324,145	8 312	*72,07
mult_8_DNF	16	16	52 810	53 621	36 005	691,33	36 018	2 362,42	36 018	*623,49
mult_8_TABL	16	16	65 536	434 660	38 057	937,68	37 800	2 315,10	38 170	*711,59
square_12_DNF	12	24	5 612	5 902	3 373	11,05	3 380	32,90	3 380	*10,27
square_12_TABL	12	24	4 096	39 962	3 359	12,57	3 368	36,48	3 368	*11,85
square_13_DNF	13	26	11 259	11 904	6 570	69,85	6 576	216,57	6 578	*65,04
square_13_TABL	13	26	8 191	87 928	6 555	57,64	6 561	174,05	6 561	*54,79
square_14_DNF	14	28	22 867	24 151	12 960	95,71	12 794	442,90	13 058	*87,37
square_14_TABL	14	28	16 384	192 100	12 941	118,00	12 762	476,20	12 980	*94,17
square_15_DNF	15	30	45 568	48 045	27 601	285,467	27 493	907,76	27 636	*218,01
square_15_TABL	15	30	32 768	416 811	28 464	334,685	28 094	1 038,20	28 497	*270,45

Таблица 13

Результаты эксперимента 10. Сравнение программ Espresso и Minim совместной минимизации

Table 13

Results of experiment 10. Comparison of Espresso and Minim joint minimization programs

Пример Example				Совместная минимизация Joint minimization			
				Espresso		Minim	
Пример Example	n	m	k	k_{\min}	Время, с Time, s	k_{\min}	Время, с Time, s
<i>Второй набор примеров</i>							
S12	12	12	4 096	*2 523	*6,67	2 685	89,96
S13	13	13	8 192	*4 929	*28,77	5 836	239,38
S14	14	14	16 384	*9 672	*176,32	11 596	1 142,07
S15	15	15	32 768	*18 768	*940,39	22 902	6 564,36
S16	16	16	65 536	*36 509	*5 462,30	–	–
System8	25	20	45 548	*28 668	*2 113,57	37 048	19 336,82
<i>Третий набор примеров</i>							
Psevdo1	10	20	1 000	*509	*1,83	704	12,98
Psevdo2	15	20	5 000	*4 754	*16,08	4 770	168,67
Psevdo3	20	20	10 000	*7 639	*333,96	*7 639	556,92
Psevdo4	25	20	25 000	*6 522	1 464,04	*6 522	*409,83
Psevdo5	30	20	30 000	Нет решения	27 000,00	*9 132	*822,08
<i>Четвертый набор примеров</i>							
mult_7_DNF	14	14	13 060	*7 224	*47,058	9 312	691,366
mult_7_TABL	14	14	16 384	*7 795	*147,727	9 827	852,310
mult_8_DNF	16	16	52 810	*27 161	*1 009,221	36 001	11 109,94
mult_8_TABL	16	16	65 536	*29 363	*2 433,198	Нет решения	14 400,00
mult_9_DNF	18	18	208 598	Нет решения	6 800,00	–	–
square_12_DNF	12	24	5 612	*2 801	*7,310	3 482	83,04
square_12_TABL	12	24	4 096	3 393	*13,565	*3 304	95,59
square_13_DNF	13	26	11 259	*5 483	*26,457	7 370	295,25
square_13_TABL	13	26	8 191	*6 847	*86,345	7 333	255,55
square_14_DNF	14	28	22 867	*10 681	*102,202	14 471	555,18
square_14_TABL	14	28	16 384	*13 932	*438,661	14 671	5 759,00
square_15_DNF	15	30	45 568	*20 664	*425,612	28 151	3 959,00
square_15_TABL	15	30	32 768	*28 407	*1 848,140	29 458	4 661,07

В алгоритме Scherzo [35] булевы функции представлены в виде двух графов BDD, причем размер этих графов напрямую не связан с числом элементов, которые они описывают. Это позволяет сократить размер используемой памяти и уменьшить время работы программы. Разработано новое циклическое ядро для работы с этими двумя графами. Алгоритм применяет новую нижнюю границу в методе ветвей и границ, основанную на отсечении возможных решений, что позволяет сократить число рекурсивных вызовов процедуры поиска. Представлены результаты экспериментальных исследований описанного алгоритма, реализованного в программе Scherzo, на примерах серии MCNC Benchmark в сравнении с программами Espresso-Signature и Espresso-Exact. Показано [35], что программа Scherzo выигрывает по времени у программ Espresso-Signature и Espresso-Exact на большинстве примеров, однако не улучшает качества получаемых решений.

В работе [36] для минимизации используются два подхода: BDD-оптимизации и BDD-дизъюнктивной декомпозиции, программа получила название MUSASHI. На стандартных примерах проведено экспериментальное сравнение этих программ минимизации.

Применение программ выполнимости конъюнктивной нормальной формы (SAT-solvers) и процедур программы Espresso позволило создать мощный логический минимизатор SAT-Espresso. В работе [37] утверждается что SAT-Espresso работает в 5–20 раз быстрее, чем Espresso, и в 3–5 раз быстрее, чем минимизатор BOOM на множестве больших примеров. Про-

грамма Espresso используется не только для минимизации, но и для интеллектуального анализа данных [38]. В статье [39] проведено сравнение разработанной для решения задач маршрутизации в сетях компьютеров программы минимизации с программами Espresso и Rondo на 16 стандартных примерах. Для примеров подсчитаны значения числа простых импликант, что показывает большую комбинаторную сложность нахождения точных решений задачи совместной минимизации. Утверждается, что полученные приближенным алгоритмом решения отличаются на 3–5 % от точных решений, однако точные решения не приводятся.

В лаборатории логического проектирования Объединенного института проблем информатики НАН Беларуси традиционно развивались методы решения задач минимизации систем булевых функций в классе ДНФ. В монографии [6] описаны точные и приближенные методы минимизации частичных и слабо определенных булевых функций. Предложены методы, усовершенствующие алгоритм Квайна – МакКласки. Для совместной минимизации систем булевых функций предложен метод конкурирующих интервалов – последовательное построение простых импликант, включаемых в решение, путем расширения интервала, представляющего импликанту, за счет включения в него непокрытых элементарных конъюнкций. В работе [3] описан метод минимизации частичных булевых функций. Частичная булева функция представляется в виде двух 2^n -компонентных булевых векторов. В основе метода лежит технология [40] параллельных операций над соседними элементами в булевом пространстве многих переменных, позволяющая быстро находить в множестве единичных значений функции элементы с малым числом соседей и определяемые ими импликанты минимизируемой булевой функции. Алгоритм ориентирован на функции с числом переменных до 24. В работах [41, 42] развивается подход к минимизации систем функций на основе аппарата покрытий троичных матриц, используемого также при решении задач декомпозиции систем функций. Опубликованные в статье [41] результаты экспериментов по совместной минимизации, проведенных на 10 примерах из первого набора примеров (табл. 14), вполне согласуются с результатами, представленными в настоящей статье. Программа из работы [41] выигрывает у Minim, получает одинаковые решения с программой Tie и лучшие решения, чем у Espresso. Преимуществом программы из статьи [41] перед Tie является то, что она может обрабатывать примеры с числом переменных $n > 22$, в то время как программа Tie обрабатывает примеры, для которых $n \leq 22$.

Таблица 14
Сравнение программ совместной минимизации

Table 14
Comparison of joint minimization programs

Пример Example	n	m	k	Программа [41] k_{\min} Program [41] k_{\min}	Minim k_{\min}	Espresso k_{\min}	Tie k_{\min}
Max512	9	6	512	*133	153	141	*133
Alu1	12	8	19	*19	*19	*19	*19
Mp2d	14	14	123	*30	36	31	*30
B12	15	9	431	*41	48	42	*41
Gary	15	11	442	*107	128	*107	*107
B2	16	17	110	*104	105	*104	*104
In2	19	10	137	*134	135	137	*134
In5	24	14	62	*62	*62	*62	–
In7	26	10	84	*54	55	*54	–
Chkn	29	7	153	*140	*140	141	–

Программные эксперименты, проведенные в работе [18], касались сравнения программы Minim с программой Espresso MV на 12 примерах, среди которых z5xp1, Radd, Root и др. Были получены одинаковые результаты совместной минимизации при решении задачи 1. Однако сравнение проводилось на примерах небольшой размерности ($n \leq 10$), был только один пример с числом входных переменных $n = 25$. Другие эксперименты касались исследования эффективности программы Minim для различных режимов ее работы на потоках псевдослучайных примеров. В работе [43] проведено детальное сравнение программ Espresso и Minim на двух при-

мерах: test2 (35 функций от 11 переменных) и pdc (40 функций от 16 переменных) – по числу импликант, размеру матрицы покрытия Квайна и по результатам минимизации, а также по сравнению техники программирования, использованной при создании упомянутых программ. В работе [10] дано сравнение программы Tie с программой Espresso и показана ее конкурентоспособность для примеров $n \leq 10$ как по качеству, так и по времени нахождения решений.

В работе [44] описан подход к распараллеливанию вычислений при программной реализации метода Квайна – МакКласки, однако приведено мало экспериментальных данных. Для учебных целей разработано много программ минимизации, описания некоторых из них представлены в методических работах [45–47].

По итогам сравнения результатов экспериментов можно сделать следующие выводы. Программа ABC является быстродействующей программой отдельной минимизации, но если на вход подается система ДНФ большой размерности, целесообразно для сокращения времени вычислений разбить данную систему на подсистемы, содержащие по несколько либо по одной функции.

Программа Espresso по-прежнему остается эффективным инструментом минимизации при числе переменных, большем 20, и числе элементарных конъюнкций в несколько тысяч. Предварительная отдельная минимизация с помощью программы ABC часто (но не всегда) позволяет сокращать время работы программы Espresso для совместной минимизации функций. Несомненным достоинством Espresso является не только ее быстродействие, но и устойчивость получения одинаковых решений для различных форм исходного задания – систем ДНФ, ортогонализированных ДНФ, систем СДНФ (таблиц истинности).

Эффективными являются программа Tie совместной минимизации для систем функций, зависящих не более чем от 20 переменных, и программа Minim, позволяющая быстрее получать решения задачи отдельной минимизации в диапазоне большой размерностей, когда вычисления с помощью Espresso требуют много времени. Вместе с тем программа Minim чувствительна к форме исходного задания минимизируемых систем функций. Она получает лучшие решения при условии, что на ее вход подаются системы ДНФ. Если же входными данными для Minim являются системы СДНФ, то качество решений программы может незначительно ухудшиться.

Оба подхода к распараллеливанию вычислений при отдельной минимизации обеспечивают сокращение времени получения решений. Второй подход к распараллеливанию является универсальным и может применяться не только для блочной минимизации функций в классе ДНФ, но и для блочной (и отдельной – по одной функции) оптимизации подсистем функций, выполняемой на основе логической минимизации многоуровневых представлений систем булевых функций.

Можно выделить области предпочтительного использования программ. Если число переменных минимизируемой системы функций не превышает 20, то целесообразно применять программу Tie как для совместной, так и отдельной минимизации. Если же число переменных больше 20 и число конъюнкций в исходном задании не превышает 10 000, то надо применять Espresso. Если данная программа будет работать долго, то при отдельной минимизации следует применить Minim. Для обработки примеров, в которых число конъюнкций превышает 50 000, целесообразно применять ABC. Если же требуется совместная минимизация, то надо выполнить сначала программу ABC, затем Espresso (однако такой подход не всегда дает сокращение времени). Для отдельной минимизации конкуренцию ABC может составить распараллеливание работы Espresso. Могут быть случаи, когда применение MinimPar даст решение быстрее, чем Espresso (в проведенных экспериментах это были псевдослучайные системы ДНФ). В очень ответственных случаях следует учитывать диапазоны применения программ и проводить вычисления, используя все имеющиеся подходы и программы, после чего выбрать лучшее решение.

Заключение. Система FLC-2 логической оптимизации, содержащая эффективные отечественные и зарубежные свободно распространяемые программы минимизации ДНФ представлений систем булевых функций, а также программные средства, позволяющие разбивать исходные системы функций на подсистемы и обрабатывать иерархические описания логических

сетей, обеспечивает решение задач минимизации функций в классе ДНФ практической размерности за приемлемое время. Благодаря распараллеливанию вычислений появляется возможность дополнительно сокращать время вычислений и увеличивать размерности решаемых задач раздельной минимизации функций. Выявлены области предпочтительного использования программ для систем булевых функций, характеризуемых как относительно небольшими значениями параметров (не более 20 переменных, менее 1000 элементарных конъюнкций), так и большими значениями (десятки аргументов и функций, десятки тысяч элементарных конъюнкций).

Вклад авторов. *И. П. Логинова* – разработка программных средств для распараллеливания программ, выполнение соответствующих экспериментов, подготовка разделов статьи, относящихся к распараллеливанию программ. *П. Н. Бибило* – проведение экспериментов без распараллеливания программ минимизации, подготовка остальных разделов статьи.

Авторы выражают благодарность В. И. Романову за помощь в программной генерации псевдослучайных систем ДНФ и реализации программной оценки сложности результатов раздельной минимизации систем функций.

Список использованных источников

1. Quine, W. V. The problem of simplifying of truth functions / W. V. Quine // *The American Mathematical Monthly*. – 1952. – Vol. 59, no. 8. – P. 521–531.
2. McCluskey, E. J. Minimization of Boolean functions / E. J. McCluskey // *The Bell System Technical J.* – 1956. – Vol. 35, no. 6. – P. 1417–1444.
3. Закревский, А. Д. ДНФ-реализация частичных булевых функций многих переменных / А. Д. Закревский, Н. Р. Торопов, В. И. Романов // *Информатика*. – 2010. – № 1(25). – С. 102–111.
4. Сапоженко, А. А. Минимизация булевых функций в классе дизъюнктивных нормальных форм / А. А. Сапоженко, И. П. Чухров // *Итоги науки и техники. Теория вероятностей. Математическая статистика. Теоретическая кибернетика*. – 1987. – Т. 25. – С. 68–116.
5. Брейтон, Р. К. Синтез многоуровневых комбинационных логических схем / Р. К. Брейтон, Г. Д. Хэчтел, А. Л. Санджованни-Винченелли // *ТИИЭР*. – 1990. – Т. 78, № 2. – С. 38–83.
6. Закревский, А. Д. Логический синтез каскадных схем / А. Д. Закревский. – М. : Наука, 1981. – 416 с.
7. Бибило, П. Н. Применение диаграмм двоичного выбора при синтезе логических схем / П. Н. Бибило. – Минск : Беларус. навука, 2014. – 231 с.
8. Авдеев, Н. А. Эффективность логической оптимизации при синтезе комбинационных схем из библиотечных элементов / Н. А. Авдеев, П. Н. Бибило // *Микроэлектроника*. – 2015. – Т. 44, № 5. – С. 383–399.
9. Бибило, П. Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // *Проблемы разработки перспективных микро- и нанoeлектронных систем*. – 2020. – Вып. 4. – С. 9–16.
10. Леончик, П. В. Минимизация систем булевых функций в классе дизъюнктивных нормальных форм / П. В. Леончик // *Информатика*. – 2006. – № 1(9). – С. 88–96.
11. Logic Minimization Algorithm for VLSI Synthesis / K. R. Brayton [et al.]. – Boston : Kluwer Academic Publishers, 1984. – 193 p.
12. Rudell, R. Multiple-valued minimization for PLA optimization / R. Rudell, A. L. Sangiovanni-Vincentelli // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. – 1987. – Vol. CAD-6, no. 5. – P. 727–751.
13. McGeer, P. Espresso-signature: A new exact minimizer for logic functions / P. McGeer, A. L. Sangiovanni-Vincentelli // *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. – 1993. – Vol. 1, no. 4. – P. 618–624.
14. Гольдберг, Е. И. Метод сбрасывания решения с локального минимума при минимизации интервального покрытия / Е. И. Гольдберг. – Минск : Ин-т техн. кибернетики АН Беларуси, 1991. – 18 с. (Препринт № 13).
15. Mishchenko, A. An Introduction to Zero-Suppressed Binary Decision Diagrams / A. Mishchenko. – Berkeley : University of California, 2014. – 15 p.
16. Карпов, Ю. Г. Model checking. Верификация параллельных и распределенных программных систем / Ю. Г. Карпов. – СПб. : БХВ-Петербург, 2010. – 560 с.

17. Леончик, П. В. Алгоритм покрытия разреженных булевых матриц / П. В. Леончик // Информатика. – 2007. – № 2(14). – С. 53–61.
18. Торопов, Н. Р. Минимизация систем булевых функций в классе ДНФ / Н. Р. Торопов // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1999. – Вып. 4. – С. 4–19.
19. Торопов, Н. Р. Приближенный алгоритм минимизации систем слабоопределенных булевых функций / Н. Р. Торопов // Известия АН СССР. Техническая кибернетика. – 1969. – № 1. – С. 72–78.
20. Романов, В. И. Булевы векторы и матрицы в C++ / В. И. Романов, И. В. Василькова // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1997. – Вып. 2. – С. 150–158.
21. Черемисинов, Д. И. Тройные векторы и матрицы в C++ / Д. И. Черемисинов, Л. Д. Черемисинова // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1998. – Вып. 3. – С. 146–156.
22. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов. – Минск : Беларус. навука, 2011. – 279 с.
23. Закревский, А. Д. Генераторы псевдослучайных логико-комбинаторных объектов в C++ / А. Д. Закревский, Н. Р. Торопов // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1999. – Вып. 4. – С. 49–63.
24. Кардаш, С. Н. Ортогонализация системы ДНФ булевых функций / С. Н. Кардаш // VII Междунар. науч.-практ. конф. «BIG DATA and Advanced Analytics» (BIG DATA 2021) : материалы междунар. науч. конф., Минск, Беларусь, 19–20 мая 2021 г. – Минск : БГУИР, 2021. – С. 26–30.
25. Торопов Н. Р. Преобразование многоярусной комбинационной сети в двухъярусную / Н. Р. Торопов // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – Вып. 5. – С. 4–14.
26. Williams, A. C++ Concurrency in Action: Practical Multithreading / A. Williams. – Manning Publications, 2012. – 528 p.
27. Darryl, G. Multicore Application Programming: for Windows, Linux, and Oracle Solaris / G. Darryl. – Addison-Wesley, 2010. – 480 p.
28. The design of OpenMP tasks / E. Ayguade [et al.] // IEEE Transactions on Parallel and Distributed Systems. – 2009. – Vol. 20, no. 3. – P. 404–418.
29. Шлее, М. Qt 5.10. Профессиональное программирование на C++ / М. Шлее. – СПб. : БХВ-Петербург, 2018. – 1072 с.
30. Fišer, P. A fast SOP minimizer for logic functions described by many product terms / P. Fišer, D. Toman // Proc. of 12th Euromicro Conf. on Digital Systems Design (DSD'09), Patras, 27–29 Aug. 2009. – Patras, 2009. – P. 757–764.
31. Fišer, P. Two-level Boolean minimizer BOOM-II / P. Fišer, H. Kubatova // Proc. of the 6th Intern. Workshop on Boolean Problems (IWSBP'04), Freiberg, Germany, 23–24 Sept. 2004. – Freiberg, 2004. – P. 221–228.
32. Hlavicka, J. BOOM – a heuristic Boolean minimizer / J. Hlavicka, P. Fiser // Computers and Information. – 2003. – Vol. 22, no. 1. – P. 19–51.
33. Fišer, P. Flexible two-level Boolean minimizer BOOM-II and its applications / P. Fišer, H. Kubatova // Proc. of 9th Euromicro Conf. on Digital System Design (DSD'06), Washington, USA, 30 Aug. – 1 Sept. 2006. – Washington, 2006. – P. 369–376.
34. Fišer, P. FC-Min: A fast multi-output Boolean minimizer / P. Fišer, J. Hlavicka, H. Kubatova // Proc. Euromicro Symp. on Digital Systems Design (DSD'03), Belek-Antalya, Turkey, 3–5 Sept. 2003. – Belek-Antalya, 2003. – P. 451–454.
35. Coudert, O. Doing two-level logic minimization 100 times faster / O. Coudert // Discrete Algorithms : Proc. of the Sixth Annual ACM-SIAM Symp., San Francisco, California, USA, 21 Jan. 1995. – San Francisco, 1995. – P. 112–121.
36. Mishchenko, A. Large-scale SOP minimization using decomposition and functional properties / A. Mishchenko, T. Sasao // Proc. of the 40th Design Automation Conf. (DAC 2003), Anaheim, California, 2–6 June 2003. – Anaheim, 2003. – P. 149–154.
37. Sapra, S. SAT-based algorithms for logic minimization / S. Sapra, M. Theobald, E. Clarke // Proc. 21st Intern. Conf. on Computer Design, San Jose, California, 13–15 Oct. 2003. – San Jose, 2003. – P. 510–517.
38. Ashmouni, E. F. Espresso for rule mining / E. F. Ashmouni, R. Ramadan, A. A. Rashed // The 5th Intern. Conf. on Ambient Systems, Networks and Technologies (ANT-2014), Hasselt, Belgium, 2–5 June 2014. – Hasselt, 2014. – P. 596–603.

39. Смагин, А. А. Применение методов минимизации булевых функций для оптимизации цифровых устройств / А. А. Смагин, А. В. Шиготаров // Изв. Самарск. науч. центра РАН. – 2009. – Т. 11, № 3(2). – С. 343–349.
40. Закревский, А. Д. Вычисления в многомерном булевом пространстве / А. Д. Закревский. – Минск : ОИПИ НАН Беларуси, 2011. – 106 с.
41. Поттосин, Ю. В. Метод минимизации системы полностью определенных булевых функций / Ю. В. Поттосин, Н. Р. Торопов, Е. А. Шестаков // Информатика. – 2008. – № 2(18). – С. 102–110.
42. Поттосин, Ю. В. Метод минимизации системы не полностью определенных булевых функций / Ю. В. Поттосин, Н. Р. Торопов, Е. А. Шестаков // Информатика. – 2009. – № 3(23). – С. 16–26.
43. Черемисинов, Д. И. Сравнение двух программ минимизации булевых функций / Д. И. Черемисинов // Автоматизация проектирования дискретных систем (CAD DD'10) : материалы Седьмой Междунар. конф., Минск, 16–17 нояб. 2010 г. – Минск : ОИПИ НАН Беларуси, 2010. – С. 194–200.
44. Модификация метода минимизации булевых функций для мультиядерной INTEL-архитектуры / С. В. Михтонок [и др.] // Радиоэлектроника и информатика. – 2007. – № 3. – С. 50–55.
45. Alharbi, E. Truth graph: A novel method for minimizing Boolean algebra expressions by using graphs / E. Alharbi // Proc. 11th Intern. Conf. on the Theory and Application of Diagrams (Diagrams 2020), Tallinn, Estonia, 24–28 Aug. 2020. – Tallinn, 2020. – P. 461–469.
46. Михеева, Е. А. Минимизация булевых функций геометрическим методом / Е. А. Михеева, А. Ф. Еникеева // Ученые записки УлГУ. Сер. Математика и информационные технологии. – 2018. – № 1. – С. 72–82.
47. Riznyk, V. Minimization of Boolean functions by combinatorial method / V. Riznyk, M. Solomko // Technology audit and production reserves. Information and Control Systems. – 2017. – Vol. 4, no. (36). – P. 49–64.

References

1. Quine W. V. The problem of simplifying of truth functions. *The American Mathematical Monthly*, 1952, vol. 59, no. 8, pp. 521–531.
2. McCluskey E. J. Minimization of Boolean functions. *The Bell System Technical Journal*, 1956, vol. 35, no. 6, pp. 1417–1444.
3. Zakrevskij A. D., Toropov N. R., Romanov V. I. *DNF-implementation of partial Boolean functions of many variables*. Informatika [Informatics], 2010, no. 1(25), pp. 102–111 (In Russ.).
4. Sapozhenko A. A., CHuhrov I. P. *Minimization of Boolean functions in the class of disjunctive normal forms*. Itogi nauki i tekhniki. Teoriya veroyatnostej. Matematicheskaya statistika. Teoreticheskaya kibernetika, [Results of Science and Technology. Probability Theory. Mathematical Statistics. Theoretical Cybernetics], 1987, vol. 25, pp. 68–116 (In Russ.).
5. Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L. *Synthesis of multi-level combinational logic circuits*. Trudy Institute inzhenerov po jelektronike i radiotekhnike [Proceedings of the Institute of Electronics and Radio Engineering], 1990, vol. 78, no. 2, pp. 38–83 (In Russ.).
6. Zakrevskij A. D. *Logicheskij sintez kaskadnyh skhem*. Logical Synthesis of Cascading Circuit. Moscow, Nauka, 1981, 416 p. (In Russ.).
7. Bibilo P. N. *Primenenie diagram dvoichnogo vybora pri sinteze logicheskikh shem*. Application of Binary Decision Diagrams in the Synthesis of Logic Circuits. Minsk, Belaruskaja navuka, 2014, 231 p. (In Russ.).
8. Avdeev N. A., Bibilo P. N. *Logical optimization efficiency in the synthesis of combinational circuits*. Mikroelektronika [Microelectronics], 2015, vol. 44, no. 5, pp. 383–399 (In Russ.).
9. Bibilo P. N., Romanov V. I. *The system of logical optimization of functional structural descriptions of digital circuits based on production-frame knowledge representation model*. Problemy razrabotki perspektivnyh mikro- i nanoelektronnyh system [Problems of Developing Promising Micro- and Nanoelectronic Systems], 2020, iss. 4, pp. 9–16 (In Russ.).
10. Leonchik P. V. *Minimization of Boolean function systems in the class of disjunctive normal forms*. Informatika [Informatics], 2006, no. 1(9), pp. 88–96 (In Russ.).
11. Brayton K. R., Hachtel G. D., McMullen C., Sangiovanni-Vincentelli A. L. *Logic Minimization Algorithm for VLSI Synthesis*. Boston, Kluwer Academic Publishers, 1984, 193 p.
12. Rudell R., Sangiovanni-Vincentelli A. L. Multiple-valued minimization for PLA optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1987, vol. CAD-6, no. 5, pp. 727–751.
13. McGeer P., Sangiovanni-Vincentelli A. L. Espresso-signature: A new exact minimizer for logic functions. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1993, vol. 1, no. 4, pp. 618–624.

14. Gol'dberg E. I. Metod sbrasyvaniya resheniya s lokal'nogo minimuma pri minimizacii interval'nogo pokrytiya. *The Method of Resetting the Solution from the Local Minimum While Minimizing the Interval Coverage*. Minsk, Institut tehniceskoy kibernetiki Akademii nauk Belarusi, 1991, 18 p. (Preprint no. 13) (In Russ.).
15. Mishchenko A. *An Introduction to Zero-Suppressed Binary Decision Diagrams*. Berkeley, University of California, 2014, 15 p.
16. Karpov Y. G. Model checking. Verifikaciya parallel'nyh i raspredelennyh programmnyh system. *Model Checking. Verification of Parallel and Distributed Software Systems*. Saint Petersburg, BHV-Peterburg, 2010, 560 p. (In Russ.).
17. Leonchik P. V. *Algorithm of sparse Boolean matrix covering*. *Informatika [Informatics]*, 2007, no. 2(14), pp. 53–61 (In Russ.).
18. Toropov N. R. *Minimization of systems of Boolean functions in the class DNF*. Logicheskoe proektirovanie [*Logical Design*]. Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 1999, iss. 4, pp. 4–19 (In Russ.).
19. Toropov N. R. *An approximate algorithm for minimizing systems of weakly defined Boolean functions*. *Izvestija Akademii nauk SSSR. Tekhnicheskaya kibernetika [Proceedings of the Academy of Sciences of the USSR. Technical cybernetics]*, 1969, no. 1, pp. 72–78 (In Russ.).
20. Romanov V. I., Vasil'kova I. V. *Boolean vectors and matrices in C++*. Logicheskoe proektirovanie [*Logical Design*]. Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 1997, iss. 2, pp. 150–158 (In Russ.).
21. Cheremisinov D. I., Cheremisinova L. D. Ternary vectors and matrices in C++. Logicheskoe proektirovanie [*Logical Design*]. Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 1998, iss. 3, pp. 146–156 (In Russ.).
22. Bibilo P. N., Romanov V. I. Logicheskoe proektirovanie diskretnyh ustrojstv s ispol'zovaniem produkcionno-frejmovoj modeli predstavlenija znaniy. *Logical Design of Discrete Devices with Use of Productional and Frame Model of Representation of Knowledge*. Minsk, Belaruskaja navuka, 2011, 279 p. (In Russ.).
23. Zakrevskij A. D., Toropov N. R. *Generators of pseudorandom logical-combinatorial objects in C++*. Logicheskoe proektirovanie [*Logical Design*], Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 1999, iss. 4, pp. 49–63 (In Russ.).
24. Kardash S. N. *Orthogonalization of the DNF system of Boolean functions*. VII Mezhdunarodnaya nauchno-prakticheskaya konferenciya "BIG DATA and Advanced Analytics" (BIG DATA 2021) : materialy mezhdunarodnoj nauchnoj konferencii, Minsk, Belarus, 19–20 maya 2021 g. [*VII International Scientific and Practical Conference "BIG DATA and Advanced Analytics" (BIG DATA 2021) : Materials of the International Scientific Conference, Minsk, Belarus, 19–20 May 2021*]. Minsk, Belorusskij gosudarstvennyj universitet informatiki i radiojelektroniki, 2021, pp. 26–30 (In Russ.).
25. Toropov N. R. *Transformation of a multi-level combinational network into a two-level one*. Logicheskoe proektirovanie [*Logical Design*], Minsk, Institut tehniceskoy kibernetiki Nacional'noj akademii nauk Belarusi, 2000, iss. 5, pp. 4–14 (In Russ.).
26. Williams A. *C++ Concurrency in Action: Practical Multithreading*. Manning Publications, 2012, 528 p.
27. Darryl G. *Multicore Application Programming: for Windows, Linux, and Oracle Solaris*. Addison-Wesley, 2010, 480 p.
28. Ayguade E., Coptly N., Duran A., Hoeflinger J., Lin Y., ..., Zhang G. The design of OpenMP tasks. *IEEE Transactions on Parallel and Distributed Systems*, 2009, vol. 20, no. 3, pp. 404–418.
29. SHlee M. Qt 5.10. Professional'noe programmirovaniye na S++. *Qt 5.10. Professional programming in C++*. Saint Petersburg, BHV-Peterburg, 2018, 1072 p.
30. Fišer P., Toman D. A fast SOP minimizer for logic functions described by many product terms. *Proceedings of 12th Euromicro Conference on Digital Systems Design, Patras, 27–29 August 2009*. Patras, 2009, pp. 757–764.
31. Fišer P., Kubatova H. Two-Level Boolean minimizer BOOM-II. *Proceedings of the 6th International Workshop on Boolean Problems (IWSBP'04), Freiberg, Germany, 23–24 September 2004*. Freiberg, 2004, pp. 221–228.
32. Hlavicka J., Fišer P. BOOM – a heuristic Boolean minimizer. *Computers and Information*, 2003, vol. 22, no. 1, pp. 19–51.
33. Fišer P., Kubatova H. Flexible two-level Boolean minimizer BOOM-II and its applications. *Proceedings of 9th Euromicro Conference on Digital System Design (DSD'06), Washington, US, 30 August – 1 September 2006*. Washington, 2006, pp. 369–376.
34. Fišer P., Hlavicka J., Kubatova H. FC-Min: A fast multi-output Boolean minimizer. *Proceedings Euromicro Symposium on Digital Systems Design (DSD'03), Belek-Antalya, Turkey, 3–5 September 2003*. Belek-Antalya, 2003, pp. 451–454.

35. Coudert O. Doing two-level logic minimization 100 times faster. *Discrete Algorithms: Proceedings of the Sixth Annual ACM-SIAM Symposium, San Francisco, California, USA, 21 January 1995*. San Francisco, 1995, pp. 112–121.
36. Mishchenko A., Sasao T. Large-scale SOP minimization using decomposition and functional properties. *Proceedings of the 40th Design Automation Conference (DAC 2003), Anaheim, California, 2–6 June 2003*. Anaheim, 2003, pp. 149–154.
37. Saprà S., Theobald M., Clarke E. SAT-based algorithms for logic minimization. *Proceedings 21st International Conference on Computer Design, San Jose, California, 13–15 October 2003*. San Jose, 2003, pp. 510–517.
38. Ashmouni E. F., Ramadan R. A., Rashed A. A. Espresso for rule mining. *The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), Hasselt, Belgium, 2–5 June 2014*. Hasselt, 2014, pp. 596–603.
39. Smagin A. A., SHigotarov A. V. *Application of Boolean function minimization methods to optimize digital devices*. Izvestiya Samarskogo nauchnogo centra Rossijskoj akademii nauk [Proceedings of the Samara Scientific Center of the Russian Academy of Sciences], 2009, vol. 11, no. 3(2), pp. 343–349.
40. Zakrevskij A. D. Vychisleniya v mnogomernom bulevom prostranstve. *Computations in a Multidimensional Boolean Space*. Minsk, Ob"edinjonnyj institut problem informatiki Nacional'noj akademii nauk Belarusi, 2011, 106 p. (In Russ.).
41. Pottosin Y. V., Toropov N. R., Shestakov E. A. A method for minimizing a system of completely specified Boolean functions. *Informatika [Informatics]*, 2008, no. 2(18), pp. 102–110 (In Russ.).
42. Pottosin Y. V., Toropov N. R., Shestakov E. A. A method for minimizing the system of incompletely specified Boolean functions. *Informatika [Informatics]*, 2009, no. 3(23), pp. 16–26 (In Russ.).
43. Cheremisinov D. I. *Comparison of two Boolean function minimization programs*. Avtomatizaciya proektirovaniya diskretnyh sistem (CAD DD'10): materialy Sed'moj Mezhdunarodnoj konferencii, Minsk, 16–17 noyabrya 2010 g. [Discrete Systems Design Automation (CAD DD'10): Proceedings of the Seventh International Conference, Minsk, 16–17 November 2010], Minsk, Ob"edinjonnyj institut problem informatiki Nacional'noj akademii nauk Belarusi, 2010, pp. 194–200.
44. Mihtonyuk S. V., Davydov M. D., Kuznecov E. S., Parfentij A. N. *Modification of the Boolean function minimization method for multi-core INTEL architecture*. Radioelektronika i informatika [Radio Electronics and Computer Science], 2007, no. 3, pp. 50–55 (In Russ.).
45. Alharbi E. Truth graph: A novel method for minimizing Boolean algebra expressions by using graphs. *Proceedings 11th International Conference on the Theory and Application of Diagrams (DIAGRAMS 2020), Tallinn, Estonia, 24–28 August 2020*. Tallinn, 2020, pp. 461–469.
46. Miheeva E. A., Enikeeva A. F. *Minimization of Boolean functions by the geometric method*. Uchenye zapiski Ul'janovskogo gosudarstvennogo universiteta. Seriya Matematika i informacionnye tehnologii [Scientific notes of Ulyanovsk State University. Series Mathematics and Information Technology], 2018, no. 1, pp. 72–82.
47. Riznyk V., Solomko M. Minimization of Boolean functions by combinatorial method. *Technology Audit and Production Reserves. Information and Control Systems*, 2017, vol. 4, no. 2(36), pp. 49–64.

Информация об авторах

Бибилу Петр Николаевич, доктор технических наук, профессор, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: bibilo@newman.bas-net.by

Логинова Ирина Петровна, кандидат технических наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси. E-mail: irilog@mail.ru

Information about the authors

Petr N. Bibilo, D. Sc. (Eng.), Professor, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: bibilo@newman.bas-net.by

Irina P. Loginova, Ph. D. (Eng.), The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: irilog@mail.ru

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

MATHEMATICAL MODELING



УДК 519.872
<https://doi.org/10.37661/1816-0301-2022-19-2-56-67>

Оригинальная статья
Original Paper

Система массового обслуживания с разделением процессора, повторными вызовами и нетерпеливостью запросов

В. И. Клименок

*Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь*
✉E-mail: klimenok@bsu.by

Аннотация

Цели. Рассматривается задача построения и исследования математической модели стохастической системы с разделением процессора, повторными вызовами и нетерпеливостью запросов. Данная система формализована в виде системы массового обслуживания, построен процесс функционирования системы, найдено условие существования стационарного распределения и предложены алгоритмы вычисления стационарного распределения и стационарных характеристик производительности системы.

Методы. Используются методы теории вероятностей, теории массового обслуживания и теории матриц.

Результаты. Функционирование системы описано в терминах многомерной цепи Маркова. Показано, что эта цепь имеет стационарное распределение, совпадающее с эргодическим, при любых приемлемых значениях параметров, описывающих входной поток, время обслуживания, процесс повторных вызовов и процесс ухода запросов из системы вследствие нетерпеливости.

Заключение. Исследован стационарный режим функционирования системы массового обслуживания с повторными вызовами, разделением процессора и двумя типами запросов, поступающих в систему в соответствии с маркированным марковским потоком. Пропускная способность канала делится между запросами двух типов в некоторой пропорции, а число запросов каждого из типов, одновременно находящихся на приборе, ограничено. Запросы одного из типов, заставшие все отведенные для них каналы занятыми, с некоторой вероятностью уходят из системы необслуженными и с дополнительной вероятностью идут на орбиту бесконечного объема, откуда делают попытки попасть на обслуживание через случайные промежутки времени. Запросы второго типа, заставшие все отведенные для них каналы занятыми, теряются. Запросы, находящиеся на орбите, проявляют нетерпеливость: каждый из них может покинуть орбиту навсегда по истечении экспоненциально распределенного времени при условии, что он не попадет на обслуживание за это время. Времена обслуживания запросов распределены по фазовому закону с разными параметрами. Функционирование системы описано в терминах многомерной цепи Маркова. Доказано, что при любых значениях параметров системы эта цепь имеет стационарное распределение. Предложены алгоритмы вычисления стационарного распределения и ряда характеристик производительности системы. Результаты исследования могут быть использованы для моделирования работы соты фиксированной емкости в беспроводной сотовой сети связи и других реальных систем, функционирующих в режиме разделения процессора.

Ключевые слова: система массового обслуживания, неоднородный входной поток, повторные вызовы, ограниченное разделение процессора, стационарное распределение, характеристики производительности

Для цитирования. Клименок, В. И. Система массового обслуживания с разделением процессора, повторными вызовами и нетерпеливостью запросов / В. И. Клименок // Информатика. – 2022. – Т. 19, № 2. – С. 56–67. <https://doi.org/10.37661/1816-0301-2022-19-2-56-67>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 01.03.2022

Подписана в печать | Accepted 18.04.2022

Опубликована | Published 29.06.2022

A retrial queueing system with processor sharing and impatient customers

Valentina I. Klimenok

Belorussian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus

✉E-mail: klimenok@bsu.by

Abstract

Objectives. The problem of constructing and investigating a mathematical model of a stochastic system with processor sharing, repeated calls, and customer impatience is considered. This system is formalized in the form of a queueing system. The operation of the queue is described in terms of multi-dimensional Markov chain. A condition for the existence of a stationary distribution is found, and algorithms for calculating the stationary distribution and stationary performance characteristics of the system are proposed.

Methods. Methods of probability theory, queueing theory and matrix theory are used.

Results. The steady state operation of a queueing system with repeated calls, processor sharing and two types of customers arriving in a marked Markovian arrival process is studied. The channel bandwidth is divided between two types of customers in a certain proportion, and the number of customers of each type simultaneously located on the server is limited. Customers of one of the types that have made all the channels assigned to them busy leave the system unserved with some probability and, with an additional probability, go to the orbit of infinite size, from where they make attempts to get service at random time intervals. Customers of the second type, which caused all the channels assigned to them to be busy, are lost. Customers in orbit show impatience: each of them can leave orbit forever if the time of its stay in orbit exceeds some random time distributed according to an exponential law. Service times of customers of different types are distributed according to the phase law with different parameters. The operation of the system is described in terms of a multi-dimensional Markov chain. It is proved that for any values of the system parameters this chain has a stationary distribution. Algorithms for calculating the stationary distribution and a number of performance measures of the system are proposed. The results of the study can be used to simulate the operation of a fixed capacity cell in a wireless cellular communication network and other real systems operating in the processor sharing mode.

Keywords: queueing system, heterogeneous input, repeated calls, limited processor sharing, stationary distribution, performance measures

For citation. Klimenok V. I. *A retrial queueing system with processor sharing and impatient customers*. *Informatika [Informatics]*, 2022, vol. 19, no. 2, pp. 56–67 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-2-56-67>

Conflict of interest. The author declare of no conflict of interest.

Введение. Одним из важных разделов математического моделирования является теория массового обслуживания, представляющая собой теоретические основы эффективного конструирования и эксплуатации систем массового обслуживания. Система массового обслуживания характеризуется своими структурой и дисциплиной обслуживания. С развитием телекоммуникационных и компьютерных сетей все большее внимание уделяется дисциплине разделения процессора, при которой все или некоторые пользователи могут получать обслу-

живание одновременно. В частности, разделение процессора используется при планировании задач в многопрограммных компьютерных системах и в сетях мобильной сотовой связи, при кешировании популярного мультимедийного контента.

Существует множество научных работ, в которых исследователи в области телекоммуникаций и теории массового обслуживания изучают системы и сети с дисциплиной разделения процессора (см., например, статьи [1–5] и ссылки в них). В большинстве работ предполагается, что входной поток в систему является стационарным пуассоновским, а времена обслуживания запросов распределены по экспоненциальному закону. Такие предположения снижают ценность результатов для приложений, так как в них не учитывается реальный характер трафика в современных сетях. В частности, не учитывается коррелированность потоков информации и то, что времена обслуживания не имеют свойства отсутствия последствия, характерного для экспоненциального распределения. Упомянутые ограничения сняты в работах [6, 7], где рассматриваются марковский поток и фазовое распределение времени обслуживания. Вместе с тем в этих работах не принимается во внимание тот факт, что реальные потоки в телекоммуникационных сетях, как правило, являются не только коррелированными, но и неоднородными. В таких потоках запросы разных типов могут делить пропускную способность канала в разных пропорциях и иметь разные распределения времени обслуживания. Неоднородность потока информации учтена в статье [8], где рассмотрена система массового обслуживания с маркированным марковским потоком *ММАР* (общепризнанная в мировой литературе аббревиатура от англ. Marked Markovian Arrival Process), ограниченным разделением процессора и повторными вызовами. Явление повторных вызовов – неотъемлемая черта многих современных телекоммуникационных и компьютерных сетей. Повторные вызовы характеризуются тем, что пользователь, получив отказ в обслуживании, не уходит из системы, а становится в режим ожидания (поступает на так называемую орбиту) и через случайные моменты времени повторно пытается получить обслуживание. Системы массового обслуживания с повторными вызовами менее изучены, чем системы с ожиданием и потерями, что объясняется сложностями при их исследовании, вызванными пространственной неоднородностью случайных процессов, описывающих функционирование таких систем.

В настоящей статье рассматривается система массового обслуживания с повторными вызовами, *ММАР*-потоком запросов двух типов и ограниченным разделением процессора. Главной отличительной чертой данной системы по сравнению с системой [8] является нетерпеливость запросов, находящихся на орбите. Каждый из таких запросов может покинуть систему навсегда в течение экспоненциально распределенного времени при условии, что он не попадет на обслуживание за это время. Рассматриваемая система может служить для математического моделирования соты мобильной сети связи с двумя типами пользователей: новыми (*new*), которые находятся в зоне видимости соты постоянно, и хэндовер (*handover*), которые пересекают территорию соты транзитом. Процесс функционирования системы описан многомерной цепью Маркова, найдено ее условие эргодичности и вычислено стационарное распределение вероятностей состояний. На основе стационарного распределения получены формулы для ряда характеристик производительности системы.

Описание системы. Представляется система массового обслуживания с разделением процессора между двумя типами запросов, которые поступают в систему в *ММАР*-потоке. В общем случае *ММАР* является коррелированным потоком $K(0 < K < \infty)$ типов запросов. Запросы поступают под управлением неприводимой цепи Маркова с непрерывным временем $v_t, t \geq 0$, которая принимает значения в множестве $\{0, 1, 2, \dots, W\}$ и называется управляющим процессом *ММАР*. Процесс v_t пребывает в состоянии v в течение экспоненциально распределенного времени с параметром $\lambda_v, v = \overline{0, W}$, после чего с вероятностью $p_k(v, v')$ переходит в состояние v' и генерируется запрос k -го типа, $k \in \{1, 2, \dots, K\}$, или с вероятностью $p_0(v, v')$ цепь переходит в состояние v' без генерации запроса, причем $p_0(v, v) = 0$. Для указанных вероятностей выполняются естественные ограничения $\sum_{k=1}^K \sum_{v'=0}^W p_k(v, v') = 1$,

$v, v' = \overline{0, W}$. Всю информацию о *ММАР* удобно хранить в виде набора матриц D_k , $k = \overline{1, K}$, порядка $(W + 1) \times (W + 1)$, элементы которых определяются следующим образом:

$$(D_k)_{v, v'} = \lambda_v p_k(v, v'), \quad v, v' = \overline{0, W}, \quad k = \overline{1, K};$$

$$(D_0)_{v, v'} = \begin{cases} \lambda_v p_0(v, v'), & v \neq v', v, v' = \overline{0, W}; \\ -\lambda_v, & v = v' = \overline{0, W}. \end{cases}$$

Из формул видно, что элементами матриц D_k , $k = \overline{1, K}$, являются интенсивности переходов процесса v_t , сопровождающиеся генерацией запроса k -го типа. Аналогичный смысл имеют недиагональные элементы матрицы D_0 , а диагональные элементы этой матрицы есть взятые с противоположным знаком интенсивности выхода процесса v_t из соответствующих состояний. Матрицы D_k , $k = \overline{1, K}$, можно задавать их матричной производящей функцией $D(z) = \sum_{k=0}^K D_k z^k$, $|z| < 1$. Значение этой функции в точке $z = 1$ является инфинитезимальным генератором управляющего процесса v_t , $t \geq 0$. Стационарное распределение данного процесса, представленное в виде вектора-строки θ , определяется как решение системы линейных алгебраических уравнений: $\theta D(1) = \theta$, $\theta \mathbf{e} = 1$. Здесь и далее $\mathbf{0}$ – вектор-строка, состоящая из нулей, а \mathbf{e} – вектор-столбец, состоящий из единиц. Интенсивность λ_k поступления запросов k -го типа в *ММАР*-потоке задается формулой $\lambda_k = \theta D_k \mathbf{e}$. Более подробное описание *ММАР*, включающее формулы для дисперсии длин интервалов между моментами поступления запросов k -го типа и коэффициентов корреляции длин двух соседних интервалов между моментами поступления запросов каждого из типов, можно найти, например, в публикациях [9, 10].

В настоящей работе, как уже было сказано ранее, сделано предположение, что в систему поступают в *ММАР*-потоке запросы двух типов, т. е. $K = 2$. На обслуживающем приборе одновременно могут обслуживаться до N запросов первого типа и до R запросов второго типа. Если на приборе обслуживается только один запрос k -го типа, $k = 1, 2$, то время его обслуживания имеет фазовое распределение *РН* (общепризнанная аббревиатура от англ. phase type distribution), заданное неприводимым представлением (β_k, S_k) и управляющим процессом $m_t^{(k)}$, $t \geq 0$, с пространством состояний $\{1, \dots, M_k, M_k + 1\}$, где состояние $M_k + 1$ является поглощающим. Первоначальное состояние (фаза) обслуживания выбирается в соответствии со стохастическим вектором-строкой β_k . Интенсивности переходов в поглощающее состояние определяются вектором-столбцом $S_0^{(k)} = -S_k \mathbf{e}$. Более детальное описание *РН*-распределения можно найти в работах [10, 11].

Запросы каждого из типов делят отведенную им пропускную способность обслуживающего прибора поровну. Если на приборе одновременно обслуживается $n^{(k)}$ запросов k -го типа, то время обслуживания любого из этих запросов имеет *РН*-распределение, заданное представлением $(\beta^{(k)}, \frac{1}{n^{(k)}} S^{(k)})$.

Если поступающий запрос первого типа видит, что в системе уже имеется N запросов его типа, то с вероятностью q поступающий запрос уходит из системы навсегда (теряется) и с дополнительной вероятностью $\bar{q} = 1 - q$ уходит на орбиту бесконечного объема, откуда делает попытки попасть на обслуживание через случайные моменты времени, распределенные по экспоненциальному закону с параметром γ . В аналогичной ситуации, если поступающий запрос второго типа видит, что в системе уже имеется R запросов его типа, то он теряется. Запросы, находящиеся на орбите, проявляют нетерпеливость: через экспоненциально распределенное с параметром α время каждый из таких запросов покидает систему навсегда.

Цепь Маркова, описывающая процесс функционирования системы. Функционирование системы описывается регулярной неприводимой цепью Маркова с непрерывным временем

$$\xi_t = \{i_t, n_t, \eta_t^{(1)}, \eta_t^{(2)}, \dots, \eta_t^{(M_1)}, r_t, \tau_t^{(1)}, \tau_t^{(2)}, \dots, \tau_t^{(M_2)}, v_t\},$$

где i_t – число запросов первого типа на орбите, $i \geq 0$;

n_t – число запросов первого типа на приборе, $n_t = \overline{0, N}$;

$\eta_t^{(m^{(1)})}$ – число запросов первого типа, которые обслуживаются на фазе $m^{(1)}$, $\eta_t^{(m^{(1)})} = \overline{0, n_t}$,
 $m^{(1)} = \overline{1, M_1}$;

r_t – число запросов второго типа на приборе, $r_t = \overline{0, R}$;

$\tau_t^{(m^{(2)})}$ – число запросов второго типа, которые обслуживаются на фазе $m^{(2)}$, $\tau_t^{(m^{(2)})} = \overline{0, r_t}$,
 $m^{(2)} = \overline{1, M_2}$;

v_t – состояние управляющего процесса ММАР, $v_t = \overline{0, W}$, в момент времени t .

Введем некоторые обозначения:

$$R = \sum_{r=0}^R C_{r+M_2-1}^{M_2-1}, \bar{W} = W + 1;$$

$\otimes (\oplus)$ – символ кронекерова произведения (суммы) матриц (см., например, [12]);

$diag\{a_1, a_2, \dots, a_n\}$ – блочная диагональная матрица, у которой диагональные блоки равны элементам, перечисленным в скобках, а остальные блоки нулевые;

$diag^+\{a_1, a_2, \dots, a_n\}$ – квадратная блочная матрица, у которой наддиагональные блоки равны элементам, перечисленным в скобках, а остальные блоки нулевые;

$diag^-\{a_1, a_2, \dots, a_n\}$ – квадратная блочная матрица, у которой поддиагональные блоки равны элементам, перечисленным в скобках, а остальные блоки нулевые;

$$\mathbf{u}_t^{(1)} = \{\eta_t^{(1)}, \eta_t^{(2)}, \dots, \eta_t^{(M_1)}\}, \mathbf{u}_t^{(2)} = \{\tau_t^{(1)}, \tau_t^{(2)}, \dots, \tau_t^{(M_2)}\};$$

C_n^m – биномиальный коэффициент.

Упорядочим состояния рассматриваемой цепи Маркова $\xi_t, t \geq 0$, следующим образом. Перенумеруем компоненты i_t, n_t, r_t в прямом лексикографическом порядке и при фиксированных значениях этих компонент перенумеруем состояния процессов $\mathbf{u}_t^{(1)}$ и $\mathbf{u}_t^{(2)}$ в обратном лексикографическом порядке. Упорядочение в обратном лексикографическом порядке требуется для дальнейшего описания интенсивностей переходов процессов $\mathbf{u}_t^{(1)}$ и $\mathbf{u}_t^{(2)}$ с использованием введенных в статьях [13, 14] матриц $P_i(\cdot), A_i(\cdot, \cdot), L_i(\cdot, \cdot)$.

$$\text{Введем в рассмотрение матрицы } \tilde{S}_l = \begin{pmatrix} 0 & O \\ \mathbf{S}_0^{(l)} & S_l \end{pmatrix}, l = 1, 2.$$

Дадим краткое объяснение вероятностных значений матриц $P_i(\cdot), A_i(\cdot, \cdot), L_i(\cdot, \cdot)$:

$L_k(n, \tilde{S}_l)$ – матрица порядка $C_{n-k+M_l-1}^{M_l-1} \times C_{n-k+M_l-2}^{M_l-1}$, которая содержит интенсивности переходов процесса $\mathbf{u}_t^{(l)}$, приводящих к завершению обслуживания одного из $n - k$ запросов l -го типа (k есть число свободных каналов для запросов l -го типа, n – общее число каналов, отведенных для запросов l -го типа);

$P_n(\beta_l)$ – матрица порядка $C_{n+M_l-1}^{M_l-1} \times C_{n+M_l}^{M_l-1}$, которая содержит вероятности переходов процесса $\mathbf{u}_t^{(l)}$, приводящих к увеличению числа запросов l -го типа, находящихся на приборе, с n до $n + 1$;

$A_n(k, S_l)$ – матрица порядка $C_{n+M_l-1}^{M_l-1} \times C_{n+M_l}^{M_l-1}$, которая содержит интенсивности переходов процесса $\mathbf{u}_t^{(l)}$ в его пространстве состояний без увеличения или уменьшения числа запросов l -го типа, находящихся на обслуживании (n есть число запросов l -го типа, находящихся на обслуживании; k – общее число каналов, отведенных для запросов l -го типа).

В дальнейшем полагаем $L_0(0) = A_0(\cdot) = P_{-1}(\cdot) = 0$.

Алгоритм вычисления матриц $P_i(\cdot), A_i(\cdot, \cdot), L_i(\cdot, \cdot)$ следует из результатов В. Рамасвами и Д. Лукантони, опубликованных в статьях [13, 14]. Шаги этого алгоритма описаны в работе [15] и приложении настоящей статьи.

Введем обозначение $Q_{n,n'}$ для интенсивностей переходов цепи из состояний, соответствующих значению n первой компоненты, в состояния, соответствующие значению n' этой компоненты, $n, n' = \overline{0, N}$. Тогда инфинитезимальный генератор цепи определяется следующей леммой.

Лемма. Инфинитезимальный генератор цепи Маркова $\xi_t, t \geq 0$, имеет блочную структуру

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & 0 & 0 & 0 & \dots \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & 0 & 0 & \dots \\ 0 & Q_{2,1} & Q_{2,2} & Q_{2,3} & 0 & \dots \\ 0 & 0 & Q_{3,2} & Q_{3,3} & Q_{3,4} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

где

$$\begin{aligned} Q_{i,i-1} &= i\gamma \operatorname{diag}^+ \{P_n(\beta_1), n = \overline{0, N-1}\} \otimes I_R \otimes I_{\bar{W}} + i\alpha I, i \geq 1; \\ Q_{i,i+1} &= \operatorname{diag} \{O_{C_{n+M_1-1}^{M_1-1}}, n = \overline{0, N-1}, I_{C_{n+M_1-1}^{M_1-1}}\} \otimes I_R \otimes \bar{q}D_1, i \geq 0; \\ (Q_{i,i})_{n,n-1} &= \frac{1}{n} L_{N-n}(N, \tilde{S}_1) \otimes I_R \otimes I_{\bar{W}}, n = \overline{1, N}, i \geq 0; \\ (Q_{i,i})_{n,n+1} &= P_n(\beta_1) \otimes I_R \otimes D_1, n = \overline{0, N-1}, i \geq 0; \\ (Q_{i,i})_{n,n} &= \Psi_n - i(\gamma + \alpha) I_{C_{n+M_1-1}^{M_1-1} R \bar{W}} + \Delta_{i,n}, n = \overline{0, N-1}, i \geq 0; \\ (Q_{i,i})_{N,N} &= \Psi_N + qD_1 - i\alpha I_{C_{n+M_1-1}^{M_1-1} R \bar{W}} + \Delta_N, i \geq 0; \\ \Psi_n &= I_{C_{n+M_1-1}^{M_1-1}} \otimes \operatorname{diag}^- \left\{ \frac{1}{r} L_{R-r}(R, \tilde{S}_2), r = \overline{1, R} \right\} \otimes I_{\bar{W}} + \\ &\quad + \frac{1}{n} A_n(N, S_1) \oplus \operatorname{diag} \left\{ 0, \frac{1}{r} A_r(R, S_2); \right. \\ &\quad \left. r = \overline{1, R} \right\} \oplus D_0 + I_{C_{n+M_1-1}^{M_1-1}} \otimes \operatorname{diag} \left\{ O_{\sum_{r=0}^{R-1} C_{r+M_2-1}^{M_2-1}}, I_{C_{R+M_2-1}^{M_2-1}} \right\} \otimes D_2 + \\ &\quad + I_{C_{n+M_1-1}^{M_1-1}} \otimes \operatorname{diag}^+ \{P_r(\beta_2), r = \overline{0, R-1}\} \otimes D_2. \end{aligned}$$

Здесь $\Delta_{i,n}, i \geq 0, n = \overline{0, N-1}$, и Δ_N – диагональные матрицы, которые обеспечивают выполнение равенства $Qe = 0$.

Доказательство леммы проводится путем анализа поведения цепи Маркова $\xi_t, t \geq 0$, на бесконечно малом интервале времени. Опишем кратко смысл ненулевых блоков генератора.

Блок $Q_{i,i-1}, i \geq 1$, состоит из интенсивностей переходов рассматриваемой цепи Маркова, сопровождающихся «удачной» повторной попыткой с орбиты запроса первого типа либо потерей запроса с орбиты из-за нетерпеливости. В любом из этих случаев число запросов на орбите уменьшается с i до $i-1$.

Блок $Q_{i,i+1}, i \geq 0$, состоит из интенсивностей переходов, сопровождающихся поступлением первичного запроса первого типа, который находит все каналы, отведенные для данного типа запросов, занятыми и уходит на орбиту. При этом число запросов на орбите увеличивается с i до $i+1$.

Блок $(Q_{i,i})_{n,n-1}, n = \overline{1, N}, i \geq 0$, состоит из интенсивностей переходов, сопровождающихся окончанием обслуживания одного из n запросов первого типа, находящихся на приборе. При этом число запросов первого типа на приборе уменьшается с n до $n-1$.

Блок $(Q_{i,i})_{n,n+1}, n = \overline{0, N-1}, i \geq 0$, состоит из интенсивностей переходов, сопровождающихся поступлением первичного запроса первого типа, заставшего свободные каналы, отведенные для запросов данного типа. При этом число запросов первого типа на приборе увеличивается с n до $n + 1$.

Недиагональные элементы блока $(Q_{i,i})_{n,n}, n = \overline{0, N}, i \geq 0$, есть интенсивности переходов, не вызывающих ни изменения числа запросов на орбите, ни изменения числа запросов первого типа на приборе. Соответствующие переходы могут быть вызваны либо окончанием обслуживания одного из запросов второго типа (матрица $I_{C_{n+M_1-1}^{M_1-1}} \otimes \text{diag}^{-1} \{ \frac{1}{r} L_{R-r}(R, S_2), r = \overline{1, R} \} \otimes I_{\overline{W}}$), либо перераспределением числа приборов, обслуживающих запросы на разных фазах, и холостыми переходами управляющего процесса ММАР (матрица $\frac{1}{n} A_n(N, S_1) \oplus \text{diag} \{ 0, \frac{1}{r} A_r(R, S_2), r = \overline{1, R} \} \oplus D_0$), либо поступлением запроса второго типа, заставшего все места занятыми (матрица $I_{C_{n+M_1-1}^{M_1-1}} \otimes \text{diag} \{ 0_{\sum_{r=0}^{R-1} C_{r+M_2-1}^{M_2-1}}, I_{C_{R+M_2-1}^{M_2-1}} \} \otimes D_2$, либо поступлением запроса второго типа, который застает свободные места и поступает на обслуживание (матрица $I_{C_{n+M_1-1}^{M_1-1}} \otimes \text{diag}^+ \{ P_r(\beta), r = \overline{0, R-1} \} \otimes D_2$). Диагональные элементы рассматриваемого блока есть взятые с противоположным знаком интенсивности выхода цепи из состояний, соответствующих i запросам на орбите и n запросам первого типа на приборе.

Следствие. Цепь Маркова $\xi_t, t \geq 0$, принадлежит классу многомерных асимптотически квазитеплицевых цепей Маркова (АКТЦМ) [16] с непрерывным временем.

Доказательство. Пусть T_i – диагональная матрица, диагональными элементами которой являются модули диагональных элементов матрицы $Q_{i,i}, i \geq 0$. Согласно [16], где приведено определение АКТЦМ, следствием будет доказано, что существуют пределы

$$Y_0 = \lim_{i \rightarrow \infty} T_i^{-1} Q_{i,i-1}, Y_1 = \lim_{i \rightarrow \infty} T_i^{-1} Q_{i,i} + I, Y_2 = \lim_{i \rightarrow \infty} T_i^{-1} Q_{i,i+1} \quad (1)$$

и матрица $Y_0 + Y_1 + Y_2$ является стохастической.

Простые вычисления приводят к следующим выражениям для матриц Y_k :

$$Y_0 = \text{diag}^+ \{ P_n(\beta_1), n = \overline{0, N-1} \} \otimes I_{R\overline{W}} + \text{diag} \{ I_{C_{n+M_1-1}^{M_1-1} R\overline{W}}, n = \overline{0, N-1}, I_{C_{N+M_1-1}^{M_1-1} R\overline{W}} \},$$

$$Y_1 = 0, Y_2 = 0.$$

Таким образом, пределы (1) существуют. Сумма $Y_0 + Y_1 + Y_2$ есть стохастическая матрица, поскольку она равна Y_0 и выполняется равенство $Y_0 \mathbf{e} = \mathbf{e}$. Из этого следует, что цепь Маркова ξ_t принадлежит классу АКТЦМ.

Стационарное распределение. Интуитивно понятно, что стационарное (совпадающее с эргодическим) распределение рассматриваемой цепи Маркова существует при любых допустимых значениях параметров вследствие того, что запросы могут уходить с орбиты из-за нетерпеливости. Строгое доказательство этого факта следует из результатов [16] для АКТЦМ и приведено в следующей теореме.

Теорема. Цепь Маркова $\xi_t, t \geq 0$, является эргодической при любых значениях параметров.

Доказательство. Как следует из работы [16], АКТЦМ $\xi_t, t \geq 0$, является эргодической, если выполняется неравенство

$$\mathbf{x} Y_2 < \mathbf{x} Y_0, \quad (2)$$

где вектор \mathbf{x} – единственное решение системы линейных алгебраических уравнений

$$\mathbf{x}(Y_0 + Y_1 + Y_2) = \mathbf{x}, \quad (3)$$

$$\mathbf{x} \mathbf{e} = 1. \quad (4)$$

Поскольку выполняется равенство $Y_0 + Y_1 + Y_2 = Y_0$ и матрица Y_0 является неприводимой стохастической, решение системы (3) определяется с точностью до константы, которая единственным образом находится из уравнения (4). В итоге получаем стохастический вектор \mathbf{x} . Подставляя этот вектор в неравенство (2) и учитывая, что $Y_2 = 0$, убеждаемся, что данное неравенство выполняется всегда. Из последнего факта следует, что рассматриваемая цепь Маркова эргодична при любых значениях параметров.

Далее будем обозначать через \mathbf{p}_i вектор-строку стационарных вероятностей, соответствующий значению i первой компоненты цепи ξ_t , $i \geq 0$. Для вычисления векторов \mathbf{p}_i , $i \geq 0$, используется численно устойчивый алгоритм, который был разработан в работе [16] для многомерных квазитеплицевых цепей Маркова общего вида и адаптирован на случай рассматриваемой системы.

Алгоритм [16] состоит из следующих шагов:

1. Находим матрицы G_i , $i \geq 0$, из уравнения обратной рекурсии:

$$G_i = (-Q_{i+1,i+1} - Q_{i+1,i+2}G_{i+1})^{-1}Q_{i+1,i}.$$

При реализации шага 1 используем факт существования предела $\lim_{i \rightarrow \infty} G_i = Y_0$, чтобы найти начальное условие для уравнения обратной рекурсии. Для этого выбираем некоторое число i_0 , полагаем $G_{i_0+1} = Y_0$, вычисляем по уравнению G_{i_0} и проверяем условие $\|G_{i_0} - Y_0\| < \epsilon$, где ϵ – заранее заданное малое число. Если условие выполняется, то полагаем все матрицы G_i для $i \geq i_0$ равными Y_0 . Остальные матрицы G_i находим из уравнения обратной рекурсии. Если для этого i_0 условие не выполняется, то выбираем новое (большее) значение i_0 .

2. Вычисляем матрицы $\bar{Q}_{i,i}$, $\bar{Q}_{i,i+1}$ по формулам

$$\bar{Q}_{i,i} = Q_{i,i} + Q_{i,i+1}G_i, \quad \bar{Q}_{i,i+1} = Q_{i,i+1}, \quad i \geq 0.$$

3. Находим матрицы F_i из рекуррентных соотношений

$$F_0 = I, \quad F_i = F_{i-1}\bar{Q}_{i-1,i}(-\bar{Q}_{i,i})^{-1}, \quad i \geq 1.$$

4. Вычисляем вектор \mathbf{p}_0 как единственное решение системы линейных алгебраических уравнений

$$\mathbf{p}_0(-\bar{Q}_{0,0}) = \mathbf{0}, \quad \mathbf{p}_0 \sum_{i=0}^{\infty} F_i \mathbf{e} = 1.$$

5. Вычисляем векторы \mathbf{p}_i по формулам $\mathbf{p}_i = \mathbf{p}_0 F_i$, $i \geq 0$.

Характеристики производительности. Вычислив стационарное распределение \mathbf{p}_i , $i \geq 0$, можно найти ряд вероятностных характеристик производительности системы. Приведем важнейшие из них:

- Распределение числа запросов первого типа, находящихся на орбите, $p_i = \mathbf{p}_i \mathbf{e}$, $i \geq 0$.
- Среднее число запросов первого типа, находящихся на орбите, $Z_{orbit} = \sum_{i=1}^{\infty} i p_i$.
- Совместное распределение числа запросов первого типа, находящихся на приборе, и состояний ММАР-потока:

$$\boldsymbol{\pi}_n^* = \sum_{i=0}^{\infty} \mathbf{p}_i \left[\begin{array}{c} \mathbf{0}^T \\ \sum_{m=0}^{n-1} C_{m+M_1-1}^{M_1-1} R \\ \mathbf{e}_{C_{n+M_1-1}^{M_1-1} R} \\ \mathbf{0}^T \\ \sum_{m=n+1}^N C_{m+M_1-1}^{M_1-1} R \end{array} \right] \otimes I_{\bar{W}}, \quad n = \overline{0, N}.$$

- Распределение числа запросов первого типа, находящихся на приборе, $\pi_n = \boldsymbol{\pi}_n^* \mathbf{e}_{\bar{W}}$, $n = \overline{0, N}$.

- Среднее число запросов первого типа, находящихся на приборе, $\bar{N} = \sum_{n=1}^N n\pi_n$.
- Вероятность того, что запрос первого типа пойдет на обслуживание, не посещая орбиту,

$$P_{imm} = 1 - \frac{1}{\lambda_1} \boldsymbol{\pi}_N^* \bar{q} D_1 \mathbf{e}.$$

Пояснение. Здесь вычитаемое представляет собой отношение интенсивности запросов первого типа, заставших в системе N запросов и ушедших на орбиту, к интенсивности поступления всех запросов первого типа. Другими словами, вычитаемое есть вероятность того, что любой поступивший запрос первого типа застанет в системе N запросов первого типа и уйдет на орбиту. Дополнительная вероятность дает искомую вероятность P_{imm} .

- Вероятность потери запроса первого типа из-за недостатка свободных каналов

$$P_{loss,1} = \frac{1}{\lambda_1} \boldsymbol{\pi}_N^* q D_1 \mathbf{e}.$$

- Вероятность потери запроса первого типа из-за нетерпеливости

$$P_{loss,1}^{imp} = \frac{\alpha}{\lambda_1} Z_{orbit}.$$

- Совместное распределение числа запросов второго типа, находящихся на приборе, и состояний ММАР-потока

$$q_r^* = \sum_{i=0}^{\infty} \mathbf{p}_i \sum_{n=0}^N \left[\begin{pmatrix} \mathbf{0}_{\sum_{m=0}^{n-1} C_{m+M_1-1}^{M_1-1}} \\ \mathbf{e}_{C_{n+M_1-1}^{M_1-1}} \\ \mathbf{0}_{\sum_{m=n+1}^N C_{m+M_1-1}^{M_1-1}} \end{pmatrix} \otimes \begin{pmatrix} \mathbf{0}_{\sum_{m=0}^{r-1} C_{m+M_2-1}^{M_2-1}} \\ \mathbf{e}_{C_{r+M_2-1}^{M_2-1}} \\ \mathbf{0}_{\sum_{m=r+1}^R C_{m+M_2-1}^{M_2-1}} \end{pmatrix} \otimes I_{\bar{W}} \right], r = \overline{0, \bar{R}}.$$

- Распределение числа запросов второго типа на приборе $q_r = q_r^* \mathbf{e}, r = \overline{0, \bar{R}}$.
- Среднее число запросов второго типа на приборе $\bar{R} = \sum_{r=1}^{\bar{R}} r q_r$.
- Вероятность потери запроса второго типа $P_{loss,2} = \frac{1}{\lambda_2} q_R^* D_2 \mathbf{e}$.

Заключение. В статье исследована система массового обслуживания с повторными вызовами, разделением процессора и двумя типами запросов, поступающих в систему в соответствии с ММАР. Времена обслуживания запросов разных типов имеют РН-распределения с разными параметрами. Запросы, находящиеся на орбите, проявляют нетерпеливость, вследствие чего могут уходить из системы необслуженными. Функционирование системы описано в терминах многомерной цепи Маркова. Доказано, что при любых допустимых значениях параметров системы эта цепь имеет стационарное распределение. Предложен алгоритм вычисления стационарного распределения и формулы для ряда характеристик производительности системы. Результаты исследования могут быть использованы для моделирования работы соты фиксированной емкости в беспроводной сотовой сети связи и других реальных систем, функционирующих в режиме разделения процессора.

References

1. Ghosh A., Banik A. D. An algorithmic analysis of the BMAP/MSP/1 generalized processor-sharing queue. *Computers and Operations Research*, 2017, vol. 79, pp. 1–11.
2. Telek M., van Houdt B. Response time distribution of a class of limited processor sharing queues. *Performance Evaluation Review*, 2018, vol. 45, no. 3, pp. 143–155. <https://doi.org/10.1145/3199524.3199548>

3. Yashkov S., Yashkova A. Processor sharing: a survey of the mathematical theory. *Automation and Remote Control*, 2007, vol. 68, pp. 662–731.
4. Zhen Q., Knessl C. On sojourn times in the finite capacity $M/M/1$ queue with processor sharing. *Operations Research Letters*, 2009, vol. 37, pp. 447–450.
5. Masuyama H., Takine T. Sojourn time distribution in a $MAP/M/1$ processor-sharing queue. *Operations Research Letters*, 2003, vol. 31, pp. 406–412.
6. Dudin S., Dudin A., Dudina O., Samouylov K. Analysis of a retrial queue with limited processor sharing operating in the random environment. *Lecture Notes in Computer Science*, 2017, vol. 10372, pp. 38–49.
7. Dudin A., Dudin S., Dudina O., Samouylov K. Analysis of queuing model with limited processor sharing discipline and customers impatience. *Operations Research Perspectives*, 2018, vol. 5, pp. 245–255.
8. Klimenok V., Dudin A. A retrial queueing system with processor sharing. *Communications in Computer and Information Science*, 2021, vol. 1391, pp. 46–60.
9. He Q. M. Queues with marked customers. *Advances in Applied Probability*, 1996, vol. 28, pp. 567–587.
10. Dudin A. N., Klimenok V. I., Vishnevsky V. M. *The Theory of Queuing Systems with Correlated Flows*. Springer, 2020, 410 p.
11. Neuts M. F. *Matrix-Geometric Solutions in Stochastic Models*. Baltimore, the Johns Hopkins University Press, 1981, 352 p.
12. Graham A. *Kronecker Products and Matrix Calculus with Applications*. Cichester, Ellis Horwood, 1981, 130 p.
13. Ramaswami V. Independent Markov processes in parallel. *Communications in Statistics. Stochastic Models*, 1985, vol. 1, pp. 419–432.
14. Ramaswami V., Lucantoni D. M. Algorithms for the multi-server queue with phase-type service. *Communications in Statistics. Stochastic Models*, 1985, vol. 1, pp. 393–417.
15. Dudina O., Kim C. S., Dudin S. Retrial queueing system with Markovian arrival flow and phase type service time distribution. *Computers and Industrial Engineering*, 2013, vol. 66, pp. 360–373.
16. Klimenok, V. I., Dudin A. N. Multi-dimensional asymptotically quasi-Toeplitz Markov chains and their application in queueing theory. *Queueing Systems*, 2006, vol. 54, pp. 245–259.

Приложение [15]

Алгоритм вычисления матриц $A_i(N, S)$, $i \in \{0, \dots, N\}$.

1. Вычисляем матрицы $\tau^{(k)}(S)$, $k \in \{0, \dots, M - 1\}$, которые получаются удалением k первых строк и k первых столбцов из матрицы S .
2. Вычисляем матрицы $T_j = \tau^{(M-2-j)}(S)$, $j \in \{1, \dots, M - 2\}$.
3. Вычисляем матрицы $L_i^{(w)}(T_j)$, используя рекуррентные формулы

$$L_i^{(0)}(T_j) = (N - i)t_{r_j,1}^j, \quad i \in \{0, \dots, N - 1\}, j \in \{1, \dots, M - 2\},$$

$$L_i^{(w)}(T_j) = \begin{pmatrix} (N - i)t_{r_j-w,1}^j I & 0 & \dots & 0 \\ L_{N-1}^{(w-1)}(T_j) & (N - i - 1)t_{r_j-w,1}^j I & \dots & 0 \\ 0 & L_{N-2}^{(w-1)}(T_j) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & t_{r_j-w,1}^j I \\ 0 & 0 & \dots & L_i^{(w-1)}(T_j) \end{pmatrix},$$

$$w \in \{1, \dots, r_j - 2\}, \quad i \in \{0, \dots, N - 1\}, \quad j \in \{1, \dots, M - 2\},$$

где $t_{k,l}^j$ – (k,l) -й элемент матрицы T_j , r_j – число строк матрицы T_j .

4. Вычисляем матрицы $U_i^{(w)}(T_j)$, используя рекуррентные формулы

$$U_i^{(0)}(T_j) = t_{1,r_j}^j, i \in \{1, \dots, N\}, j \in \{1, \dots, M-2\},$$

$$U_i^{(w)}(T_j) = \begin{pmatrix} t_{1,r_j-w}^j & U_N^{(w-1)}(T_j) & 0 & \dots & 0 & 0 \\ 0 & t_{1,r_j-w}^j & U_{N-1}^{(w-1)}(T_j) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & t_{1,r_j-w}^j & U_i^{(w-1)}(T_j) \end{pmatrix},$$

$$w \in \{1, \dots, r_j - 2\}, i \in \{1, \dots, N\}, j \in \{1, \dots, M-2\}.$$

5. Вычисляем матрицы $L_i(N, T_j) = L_i^{(r_j-2)}(T_j)$, $i \in \{0, \dots, N-1\}$, и $U_i(N, T_j) = iU_i^{(r_j-2)}(T_j)$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, M-2\}$.

6. Вычисляем матрицы $A_i^{(w)}$, используя рекуррентные формулы

$$A_i^{(0)} = \begin{pmatrix} 0 & iS_{M-1,M} & 0 & \dots & 0 & 0 \\ S_{M,M-1} & 0 & (i-1)S_{M-1,M} & \dots & 0 & 0 \\ 0 & 2S_{M,M-1} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & S_{M-1,M} \\ 0 & 0 & 0 & \dots & iS_{M,M-1} & 0 \end{pmatrix}, i \in \{1, \dots, N\},$$

$$A_i^{(j)} = \begin{pmatrix} 0 & \frac{iU_N(N, T_j)}{N} & 0 & \dots & 0 & 0 \\ L_{N-1}(N, T_j) & A_1^{(j-1)} & \frac{(i-1)U_{N-1}(N, T_j)}{N-1} & \dots & 0 & 0 \\ 0 & L_{N-2}(N, T_j) & A_2^{(j-1)} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A_{i-1}^{(j-1)} & \frac{U_{N-i+1}(N, T_j)}{N-i+1} \\ 0 & 0 & 0 & \dots & L_{N-i}(N, T_j) & A_i^{(j-1)} \end{pmatrix},$$

$$i \in \{1, \dots, N\}, j \in \{1, \dots, M-2\}.$$

7. Вычисляем матрицы $A_i(N, S)$ следующим образом: $A_0(N, S) = O_{1 \times 1}$, $A_i(N, S) = A_i^{(M-2)}$, $i \in \{1, \dots, N\}$.

Замечание 1. Если $M \leq 2$, то шаги 1-5 пропускаем и вычисляем только матрицы $A_i^{(0)}$, $i \in \{1, \dots, N\}$, на шаге 6.

Алгоритм вычисления матриц $L_i(N, \tilde{S})$, $i \in \{0, \dots, N-1\}$.

1. Вычисляем матрицы $L_i(\tilde{S})$, используя рекуррентные формулы

$$L_i^{(0)}(\tilde{S}) = (N-i)t_{r,1}, i \in \{0, \dots, N-1\},$$

$$L_i^{(w)}(\tilde{S}) = \begin{pmatrix} (N-i)t_{r-w,1}I & O & \dots & O \\ L_{N-1}^{(w-1)}(\tilde{S}) & (N-i-1)t_{r-w,1}I & \dots & O \\ O & L_{N-2}^{(w-1)}(\tilde{S}) & \dots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & t_{r-w,1}I \\ O & O & \dots & L_i^{(w-1)}(\tilde{S}) \end{pmatrix},$$

$$w \in \{1, \dots, r-2\}, i \in \{0, \dots, N-1\},$$

где $t_{k,l}$ – (k,l) -й элемент матрицы \tilde{S} , r – число строк матрицы \tilde{S} .

2. Вычисляем матрицы $L_i(N, \tilde{S}) = L_i^{(r-2)}(\tilde{S})$, $i \in \{0, \dots, N-1\}$.

3. Вычисляем матрицы $L_i(N, \tilde{S})$ следующим образом: $L_i(N, \tilde{S}) = L_i^{(M-1)}(\tilde{S})$, $i \in \{0, \dots, N-1\}$, $L_N(N, \tilde{S}) = O_{1 \times 1}$.

Замечание 2. Здесь наиболее трудоемкий шаг 1 повторяет шаг 3 при вычислении матриц $A_i(N, S)$, только матрицы T_j заменяются на матрицу \tilde{S} и не зависят от j .

Алгоритм вычисления матриц $P_i(\beta)$, $i \in \{1, \dots, N-1\}$.

1. Вычисляем матрицы $P_i^{(j)}$ размерности $(i+1) \times (i+2)$, используя рекуррентные формулы

$$P_i^{(0)} = \begin{pmatrix} \beta_{M-1} & \beta_M & 0 & \dots & 0 & 0 \\ 0 & \beta_{M-1} & \beta_M & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \beta_{M-1} & \beta_M \end{pmatrix}, i \in \{1, \dots, N-1\},$$

$$P_i^{(j)} = \begin{pmatrix} \beta_{M-j-1} & z^{(j)} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \beta_{M-j-1}I & P_1^{(j-1)} & O & \dots & O & O \\ \mathbf{0}^T & O & \beta_{M-j-1}I & P_2^{(j-1)} & \dots & O & O \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}^T & O & O & O & \dots & \beta_{M-j-1}I & P_i^{(j-1)} \end{pmatrix},$$

$$j \in \{1, \dots, M-2\}, i \in \{1, \dots, N-1\},$$

где $z^{(j)} = (\beta_{M-j}, \beta_{M-j+1}, \dots, \beta_M)$, $j \in \{1, \dots, M-2\}$.

2. Вычисляем матрицы $P_i(\beta)$ следующим образом: $P_0(\beta) = \beta$, $P_i(\beta) = P_i^{(M-2)}$, $i \in \{1, \dots, N-1\}$.

Информация об авторе

Клименок Валентина Ивановна, доктор физико-математических наук, профессор, главный научный сотрудник научно-исследовательской лаборатории прикладного вероятностного анализа, Белорусский государственный университет.
 E-mail: klimenok@bsu.by

Information about the author

Valentina I. Klimenok, D. Sc., Prof., Chief Scientific Researcher of Laboratory of Applied Probability, Belorussian State University.
 E-mail: klimenok@bsu.by



УДК 519.8; 621.3:007; 621.3:001.891.57
<https://doi.org/10.37661/1816-0301-2022-19-2-68-84>

Оригинальная статья
Original Paper

Об оптимизации смешанной зарядной инфраструктуры электробусов для городских маршрутов

Б. М. Розин[✉], И. А. Шатерник

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉E-mail: rozin@newman.bas-net.by*

Аннотация

Цели. При переходе от парка дизельных автобусов к парку электробусов актуальной является оптимизация зарядной инфраструктуры, совмещающей технологии медленной зарядки батарей в депо в ночное время и быстрой подзарядки на конечных остановках маршрутов. Цель исследования заключается в создании моделей и методов выработки экономически эффективных решений по выбору зарядной инфраструктуры такого типа для парка электробусов, обслуживающих систему городских маршрутов с учетом ряда специфических условий. Функционирование парка и зарядной инфраструктуры моделируется как для депо в ночной период, так и для конечных остановок в наиболее представительный период дня, характеризующийся наибольшей интенсивностью пассажиропотока и максимальным расходом электроэнергии.

Методы. Используются методы теории множеств, теории графов и линейной аппроксимации.

Результаты. Разработана математическая модель задачи оптимизации зарядной инфраструктуры смешанного типа для парка электробусов. В качестве целевой функции выбрана суммарная дневная стоимость зарядных станций, износа батарей электробусов парка и потребленной электроэнергии. Модель сформулирована в виде задачи смешанного целочисленного линейного программирования.

Заключение. Для решения сформулированной задачи целочисленного линейного программирования могут использоваться стандартные решатели типа IBM ILOG CPLEX. Решение задачи заключается в выборе длительностей и расписаний зарядки электробусов на зарядных станциях малой мощности в депо в ночное время и на зарядных станциях большой мощности конечных остановок в заданном диапазоне часов пик.

Ключевые слова: электробус, аккумуляторная батарея, станция зарядки, оптимизация, смешанное целочисленное линейное программирование

Благодарности. Исследование выполнялось в рамках проекта «Модели и методы для систем поддержки принятия решений при проектировании, планировании и управлении высокотехнологичными, роботизированными производствами и электротранспортом» программы ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства» (2021–2025 гг.).

Для цитирования. Розин, Б. М. Об оптимизации смешанной зарядной инфраструктуры электробусов для городских маршрутов / Б. М. Розин, И. А. Шатерник // Информатика. – 2022. – Т. 19, № 2. – С. 68–84.
<https://doi.org/10.37661/1816-0301-2022-19-2-68-84>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 15.11.2021
Подписана в печать | Accepted 11.02.2022
Опубликована | Published 29.06.2022

On optimization of the mixed charging infrastructure of electric buses for urban routes

Boris M. Rozin[✉], Ilya A. Shaternik

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
✉E-mail: rozin@newman.bas-net.by*

Abstract

Objectives. When transition from a fleet of diesel buses to a fleet of electric buses, it is important to optimize the charging infrastructure, which combines the slow-charging technologies at the depot overnight and fast recharging at the terminals of the routes. The purpose of the study is to create models and methods for developing the cost-effective solutions for selecting this type of charging infrastructure for a fleet of electric buses serving the city route system, taking into account a number of specific conditions. The operation of the fleet and charging infrastructure is modeled both for the depot at night and for the terminal stops in the most representative period of the day, characterized by the highest intensity of passenger traffic and maximum power consumption.

Methods. Methods of set theory, graph theory and linear approximation are used.

Results. A mathematical model has been developed for the problem of optimizing a mixed-type charging infrastructure for an electric bus fleet. The total daily cost of charging stations, degradation of electric bus batteries and consumed electricity was chosen as the objective function. The model is formulated as a mixed integer linear programming problem.

Conclusion. To solve the formulated problem, standard solvers like IBM ILOG CPLEX can be used. The solution of the problem lies in the choice of durations and schedules for charging electric buses at low-capacity charging stations in the depot at night and at high-capacity charging stations of terminal stops in a given range of peak hours.

Keywords: electric bus, battery, charging station, optimization, mixed integer linear programming

Acknowledgements. The study was carried out within the Project "Models and methods for decision support systems in the design, planning and management of high-tech, robotic production and electric transport" in frames of the State research program "Digital and space technologies, human, society and state safety" (2021–2025).

For citation. Rozin B. M., Shaternik I. A. *On optimization of the mixed charging infrastructure of electric buses for urban routes*. *Informatika [Informatics]*, 2022, vol. 19, no. 2, pp. 68–84 (In Russ.).
<https://doi.org/10.37661/1816-0301-2022-19-2-68-84>

Conflict of interest. The authors declare of no conflict of interest.

Введение. В литературе значительное внимание в последние годы уделяется различным аспектам планирования зарядной инфраструктуры для подзарядки городского электротранспорта, в частности электробусов (ЭБ) [1–10]. Применяемые технологии зарядки ЭБ отличаются большим разнообразием: ЭБ могут заряжаться только в депо; в депо и на конечных остановках маршрутов; в депо, на конечных и промежуточных автобусных остановках (URL: <https://zeeus.eu/uploads/publications/documents/zeeus-report2017-2018-final.pdf>). На практике основными альтернативами технологий зарядки являются частые зарядки аккумуляторных батарей (далее – батарея) ЭБ небольшой емкости на остановках маршрутов либо редкие зарядки батарей большой емкости в депо. Для зарядки на маршруте обычно требуются зарядные устройства большой мощности (до 400–500 кВт), в то время как для зарядки в депо могут использоваться зарядные устройства малой мощности (80–100 кВт) [1, 2]. При выборе подходящего варианта батареи принимаются во внимание ее технические характеристики и экономическая целесообразность обслуживания парка ЭБ (см., например, [2, 3]). Большие батареи с медленной зарядкой обеспечивают больший запас хода на одном заряде, чем небольшие батареи с быстрой зарядкой, но они более дорогие и уменьшают пассажировместимость ЭБ. С другой стороны, использование небольших батарей требует больших инвестиций, связанных с инфраструктурой зарядки в пути [4].

В настоящей работе рассматривается задача оптимизации конфигурации зарядной инфраструктуры смешанного типа для парка ЭБ. Конфигурация зарядной инфраструктуры смешанного типа характеризуется числом идентичных зарядных станций медленной зарядки, размещаемых в депо и используемых для зарядки ЭБ преимущественно в ночное время, а также числами идентичных станций быстрой зарядки ЭБ, устанавливаемых на конечных остановках обслуживаемых маршрутов, на которых ЭБ проводят значительное время (7–30 мин).

Рассматривается набор замкнутых круговых маршрутов, обслуживаемых парком ЭБ нескольких типов. Каждому ЭБ парка назначается единственный маршрут обслуживания, включающий вспомогательные рейсы из депо на стартовую конечную остановку маршрута и возвращения в депо с одной из конечных остановок, а также циклическое выполнение пары сервисных рейсов со стартовой конечной остановки до альтернативной конечной остановки (если их две) и обратно. Для ЭБ задано также количество циклов обслуживания его маршрута в течение дня. Транспортировка пассажиров по маршрутам рассматривается для наиболее *представительного периода*, характеризующегося наименьшими интервалами времени между ЭБ и наибольшим расходом электроэнергии на маршрутах (как правило, для Европы это часы пик рабочего дня февраля). Способность зарядной инфраструктуры обеспечить функционирование парка ЭБ в представительный период дает возможность их функционирования в любой другой период года. Каждый маршрут может обслуживаться ЭБ различных типов.

Различные ЭБ парка одного типа оборудуются одинаковыми батареями. В процессе функционирования ЭБ батарея изнашивается. Основными факторами, влияющими на износ батареи, являются скорости ее зарядки и разрядки, уровень заряда (УЗ), при котором осуществляются ее зарядка и разрядка, и температура батареи [5, 6]. Как правило, производителем батареи указывается диапазон УЗ, при котором батарея может функционировать достаточно продолжительное время, а также характеристики достижимого числа циклов ее зарядки и разрядки за период эксплуатации в зависимости от глубины разряда, скоростей зарядки и разрядки при нормальной температуре. Емкость батареи, применяемые скорости ее разрядки и зарядки определяются типом ЭБ, характеристиками обслуживаемых маршрутов, интенсивностью пассажиропотока, номинальной мощностью и силой тока используемых зарядных станций.

Величины среднего расхода электроэнергии на маршрутах для каждого типа ЭБ в представительный период времени предполагаются известными. УЗ батареи любого ЭБ в процессе функционирования не должен выходить за пределы заданного диапазона и должен быть полностью восстановлен в депо в конце дневного цикла перед выполнением рейсов следующего дня.

Средние потери УЗ для любых ЭБ одного типа в любом цикле обслуживания одного и того же маршрута в рамках представительного периода считаются одинаковыми. Длительности подзарядки ЭБ одного типа, обслуживающих один и тот же маршрут, на фиксированной конечной остановке данного маршрута считаются одинаковыми для любого посещения этой остановки, но могут отличаться от длительности зарядки ЭБ на альтернативной конечной остановке. Длительности зарядки в депо для ЭБ одного типа, обслуживающих один и тот же маршрут, также считаются одинаковыми. Номинальная мощность каждой станции зарядки предполагается неизменной в течение всего периода эксплуатации.

Требуется выбрать такую конфигурацию рассматриваемой зарядной инфраструктуры, которая минимизирует суммарную приведенную дневную стоимость этой инфраструктуры, износа батарей парка ЭБ и потребленной электроэнергии и обеспечивает обслуживание данным парком ЭБ городских маршрутов согласно расписаниям представительного периода.

Описание проблемной ситуации. Приведем перечень принятых при моделировании проблемной ситуации основных предположений:

1. Парк ЭБ включает электробусы нескольких типов. Каждый маршрут из заданного набора могут обслуживать ЭБ различных типов.

2. ЭБ оборудованы аккумуляторными батареями, идентичными для ЭБ одного типа.

3. Каждый ЭБ обслуживает назначенный ему единственный маршрут, содержащий одну либо две конечные остановки, предназначенные для быстрой подзарядки батарей ЭБ в течение дня. Любая конечная остановка может принадлежать нескольким маршрутам. После выполне-

ния дневного расписания каждый ЭБ парка возвращается в единственное депо, где в ночное время выполняется подзарядка его батареи на одной из идентичных зарядных станций медленной зарядки до максимально допустимого уровня перед выходом на маршрут на следующий день. Расписание движения ЭБ парка на маршруте определяется последовательностью выхода транспортных единиц на маршрут из депо и временным интервалом между соседними ЭБ в последовательности.

4. Зарядка ЭБ как при приходе в депо, так и при каждом посещении конечной остановки выполняется однократно на одной зарядной станции без прерываний. Одновременно на одной зарядной станции может заряжаться только один ЭБ. Подзарядка ЭБ производится при каждом посещении любой конечной остановки обслуживаемого маршрута, в том числе при первом выходе на маршрут и перед отправлением в депо после завершения последнего цикла обслуживания маршрута.

5. Зарядка в депо в ночное время выполняется для всех ЭБ всех маршрутов, обслуживаемых в этом депо. Все события зарядки на каждой конечной остановке маршрутов рассматриваются в выбранный ограниченный представительный период дня.

6. Представительным назовем такой период времени, в течение которого принятые для этого периода решения по выбору конфигурации зарядной инфраструктуры парка ЭБ обеспечивают ее допустимое функционирование в любой другой период времени [7]. С некоторой степенью неопределенности наиболее представительный период может характеризоваться наибольшим расходом ЭБ энергии при движении по тем же отрезкам маршрутов и наименьшими интервалами между ЭБ в расписании. (Обычно это часы пик, в том числе зимой, при полной нагрузке всех электроприборов.) Будем считать, что диапазон представительного периода одинаков для всех маршрутов.

7. Для обеспечения робастности функционирования выбираемой зарядной инфраструктуры любой конечной остановки используется, кроме представительного периода, заданное *гарантирующее расписание* прихода ЭБ различных маршрутов на эту остановку. С некоторой степенью неопределенности такому расписанию соответствуют синхронизированные друг относительно друга расписания прихода на рассматриваемую конечную остановку ЭБ различных маршрутов для их подзарядки.

8. Для каждого типа ЭБ заданы величины среднего расхода энергии на маршрутах для представительного периода.

9. Расход энергии ЭБ для представительного периода и минимальный интервал между последовательно движущимися ЭБ для каждого маршрута распространяются на весь рассматриваемый день обслуживания.

10. Длительность зарядки ЭБ на конечной остановке обслуживаемого маршрута предполагается одинаковой для любого цикла обслуживания маршрута этим ЭБ.

11. Заданы табличные зависимости достижимого числа циклов зарядки и разрядки батареи ЭБ каждого типа за период ее эксплуатации от глубины разряда и скорости зарядки (разрядки).

12. Величины номинальной применяемой выходной мощности зарядных станций депо и конечных остановок заданы для каждой пары (тип станции, тип ЭБ) и не зависят от времени суток.

13. При обслуживании любого маршрута УЗ батареи ЭБ не должен выходить за границы заданного диапазона.

14. В допустимом диапазоне изменения УЗ батареи ЭБ возрастает линейно в зависимости от времени зарядки при постоянной мощности зарядки.

Отметим, что зависимость УЗ от времени движения ЭБ по маршруту является линейно убывающей функцией при движении по маршруту, а зависимость УЗ от времени зарядки на зарядной станции – неубывающей кусочно-линейной функцией. При этом, если батарея заряжена не полностью, функция возрастает и функция постоянна при полностью заряженной батарее. Таким образом, на каждом цикле обслуживания маршрута график функции УЗ батареи представляет собой непрерывную кусочно-линейную ломаную (рис. 1), где интервалы [2, 3], [4, 5] и [6, 7] соответствуют зарядке.

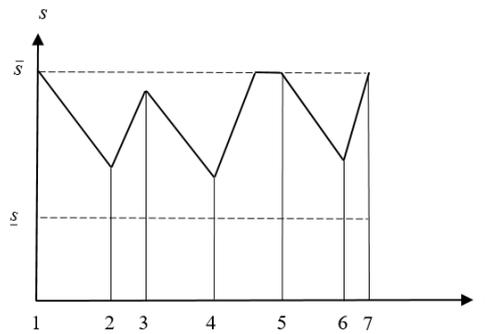


Рис. 1. Пример ломаной линии уровня заряда ЭБ

Fig. 1. An example of a polygonal line of the EB charge level

Постановка задачи выбора конфигурации зарядной инфраструктуры. Обозначим через R множество городских маршрутов, обслуживаемых парком ЭБ. Парк характеризуется заданным множеством $J = \{1, \dots, n\}$ индексов ЭБ, тип каждого $j \in J$ соответствует одному из типов ЭБ множества EB . Подзарядка ЭБ производится в единственном депо dep преимущественно в ночное время на идентичных станциях cd медленной зарядки и на конечных остановках маршрутов из множества $\Theta = \{1, \dots, \theta\}$ идентичными станциями c быстрой зарядки. Скорость зарядки (разрядки) батареи ЭБ задается в величинах xC (1/ч), где x – величина, обратная максимальному времени зарядки (разрядки) полностью разряженной (заряженной) батареи при данной скорости, $1C$ обозначает полную зарядку (разрядку) батареи за 1 ч, $2C$ – то же, но за 0,5 ч и т. д.

Представим систему обслуживаемых маршрутов из R взвешенным ориентированным мультиграфом $G = (U, V)$ с множеством вершин $U = \{\Theta \cup \{dep\}\}$ и множеством дуг $V = \{(\alpha, \beta) \mid \alpha, \beta \in U\}$. Обозначим $\Theta(r) \subset \Theta$ подмножество конечных остановок маршрута r , $\bigcup_{r \in R} \Theta(r) = \Theta$, $|\Theta(r)| \leq 2$, $r \in R$.

Каждой из вершин множества U сопоставляется вариант устанавливаемых в ней зарядных станций: станций cd медленной зарядки для депо и станций c быстрой зарядки для конечных остановок. Каждая дуга $(\alpha, \beta) \in V$ соответствует рейсу одного из маршрутов, указывает направление движения ЭБ в рейсе и характеризуется длиной $L_{\alpha, \beta}$ рейса, в общем случае $L_{\alpha, \beta} \neq L_{\beta, \alpha}$. Каждую пару вершин могут соединять несколько одинаково направленных дуг, обозначающих рейсы различных маршрутов.

Каждому маршруту $r \in R$ сопоставляются:

– конечные остановки $v_\sigma(r)$ маршрута, $\sigma = 1, \dots, \bar{v}_r$, где $\bar{v}_r \in \{1, 2\}$, $v_\sigma(r) \in \Theta(r)$;

– последовательность обхода конечных остановок маршрута $\pi_r = (dep, (v_s(r), v_p(r), v_s(r))_1, \dots, (v_s(r), v_p(r), v_s(r))_q, \dots, v_f(r), dep)$ (s обозначает стартовую конечную остановку, p – альтернативную остановку для случая $\bar{v}_r = 2$, $f \in \{s, p\}$); π_r начинается и заканчивается в депо, содержит повторяющиеся идентичные циклы $q = 1, 2, \dots$ из двух сервисных рейсов $(v_s(r), v_p(r))$, $(v_p(r), v_s(r))$, где $s = 1$, $p = 2$ либо $s = 2$, $p = 1$ в зависимости от стартовой конечной остановки $v_s(r) \in \Theta(r)$, и финальную перед возвращением в депо конечную остановку $v_f(r)$. При $f = s$ все циклы (полные) одинаковы, при $f = p$ последний цикл является неполным и состоит из одного рейса $(v_s(r), v_p(r))$. В случае одной конечной остановки маршрута ($\bar{v}_r = 1$) все циклы обслуживания маршрута состоят из одного сервисного рейса с единственной начальной и конечной вершиной $v_s(r)$;

– подмножество $J_r \subseteq J$ ЭБ, обслуживающих маршрут, $\bigcup_{r \in R} J_r = J$, $J_\eta \cap J_\mu = \emptyset$ при $\eta \neq \mu$, $\eta, \mu \in R$;

– средняя скорость V_r движения ЭБ по маршруту;

– минимальный интервал λ_r времени между соседними ЭБ в течение представительного периода.

В табл. 1 приведены данные по трем маршрутам (27, 43 и 59) в Минске, обслуживаемым ЭБ. Для обозначения конечных остановок маршрутов использованы их сокращенные наименования: ДС Курасовщина – *Кур*, ДС Дружная – *Др*, ДС Серова – *Сер*, Долгобродская – *Дол*, Депо троллейбусного парка № 2 – *dep* (URL: <https://www.minsktrans.by>; URL: <https://wikiroutes.info/minsk/catalog>). На рис. 2 изображена схема этих маршрутов. Рейсы, составляющие маршрут, обозначаются составными числами <номер маршрута>-<номер рейса>.

Таблица 1
 Характеристики маршрутов 27, 43 и 59, обслуживаемых электробусами

Table 1
 Characteristics of routes 27, 43 and 59 served by electric buses

Маршрут Route	Рейс Flight	Кон. ост. 1– кон. ост. 2 Term.1–term.2	Расстояние, км Distance, km	Минимальный интервал, мин Minimum interval, min
27	27-1	<i>Кур–Др</i>	7,4	4–5
	27-2	<i>Др–Кур</i>	7,44	4–5
	27-3	<i>dep–Кур</i>	12,28	–
	27-4	<i>Кур–dep</i>	11,62	–
43	43-1	<i>Сер–Др</i>	7,38	13
	43-2	<i>Др–Сер</i>	8,01	11–13
	43-3	<i>dep–Сер</i>	12,78	–
	43-4	<i>Сер–dep</i>	11,59	–
59	59-1	<i>Сер–Дол</i>	11,99	6–7
	59-2	<i>Дол–Сер</i>	12,81	6–7
	59-3	<i>dep–Дол</i>	2,51	–
	59-4	<i>dep–Сер</i>	11,89	–
	59-5	<i>Сер–dep</i>	11,08	–
	59-6	<i>Дол–dep</i>	2,47	–

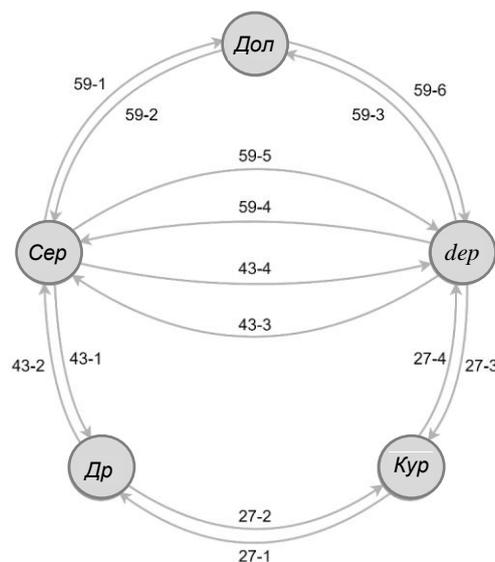


Рис. 2. Схема маршрутов 27, 43 и 59 в Минске

Fig. 2. Scheme of routes 27, 43 and 59 of Minsk

Легко убедиться, что маршрут с двумя конечными остановками может включать четыре либо шесть различных рейсов (дуг мультиграфа G). Маршрут с одной конечной остановкой

включает три рейса (три дуги мультиграфа G), два из которых соответствуют вспомогательным рейсам из (в) депо, один (дуга мультиграфа G является петлей) – сервисному рейсу перевозки пассажиров.

Для каждого типа ЭБ $e \in EB$ заданы:

- емкость E_e (энергетическая) батареи, кВт·ч;
- инвестиционная стоимость cst_b_e батареи;
- диапазон $[\underline{s}_e, \bar{s}_e]$ допустимых относительных значений s_e УЗ батареи по отношению к E_e , $0 \leq \underline{s}_e \leq \bar{s}_e \leq 1$;
- средний удельный расход sl_e энергии (заряда) на маршрутах, кВт·ч/км;
- постоянная скорость crd_e зарядки батареи на станции cd в депо;
- максимальная длительность τ_e^{\max} зарядки в депо от УЗ \underline{s}_e до УЗ \bar{s}_e при скорости зарядки crd_e ;
- постоянная скорость $crst_e$ зарядки батареи на станции c любой конечной остановки $v \in \Theta$;
- максимальная длительность t_e^{\max} зарядки на любой конечной остановке $v \in \Theta$ от УЗ \underline{s}_e до УЗ \bar{s}_e при скорости зарядки $crst_e$;
- скорость dr_e разрядки батареи в процессе движения по маршрутам;
- характеристики достижимого числа циклов $ACC_e(DoD, \circ)$ (от англ. *Attainable Cycle Count*) зарядки (разрядки) батареи за период ее эксплуатации в зависимости от глубины разряда DoD (от англ. *Depth of Discharge*) и скорости зарядки ($\circ = crd_e$ или $\circ = crst_e$) либо разрядки ($\circ = dr_e$).

Отметим, что мощность зарядки (разрядки) батареи определяется произведением ее скорости зарядки (разрядки) и емкости соответственно: $crd_e \cdot E_e$, $crst_e \cdot E_e$ ($dr_e \cdot E_e$).

В табл. 2 приведены характеристики ЭБ двух типов производства ОАО «Белкоммунмаш», работающих на маршрутах в Минске. Для зарядки в депо и на остановках применяются зарядные станции СЗА-1 с выходной мощностью 300 кВт, выходным напряжением 600 В, максимальным током 500 А, сроком эксплуатации 10 лет, ориентировочной инвестиционной стоимостью 830 тыс. руб. и годовой операционной стоимостью 12 тыс. руб. Характеристики ЭБ и зарядных станций предоставлены ОАО «Белкоммунмаш», данные по стоимости батарей заимствованы из работы [5]. Применяемые скорости зарядки батарей: 0,5С в депо и 6С на конечных остановках маршрутов. Соответственно, номинальные величины мощности и тока зарядки составляют 24 кВт и 40 А для депо, 288 кВт и 480 А на остановках при номинальном напряжении 600 В батареи. Применяемый диапазон $[\underline{s}_e, \bar{s}_e]$ допустимых УЗ батарей составляет [0,3; 1].

Таблица 2
Характеристики электробусов производства ОАО «Белкоммунмаш»

Table 2
Characteristics of electric buses produced by JSC Belkommunmash

Тип ЭБ <i>EB type</i>	Пассажиро- вместимость, чел. <i>Passenger capacity, persons</i>	Диапазон хода на одном заряде, км <i>Travel range per charge, km</i>	Средний расход энергии, кВт·ч/км <i>Average energy consumption, kWh/km</i>	Емкость батареи, кВт·ч <i>Battery capacity, kWh</i>	Макс. время зарядки в депо, ч <i>Max. depot charging time, hour</i>	Макс. время зарядки на остановках, мин <i>Max. charging time at stops, min</i>	Стоимость батареи, евро <i>Battery cost, EUR</i>
E433	153	15	2,25	48	1,4	7	20 640
E420	87	15	1,75	48	1,4	7	20 640

Каждому ЭБ $j \in J$ сопоставляются:

- его тип $e(j) \in EB$;
- маршрут $r(j) \in R$ обслуживания;
- моменты td_j , ta_j первого выхода из депо на маршрут и прихода в депо в конце рабочего дня;

- число k_j циклов сервисного обслуживания маршрута $r(j)$ в день, k_j -й цикл может быть неполным;
- последовательность $\pi_{r(j)}$ обхода конечных остановок $v_s(r(j)), v_p(r(j)) \in \Theta(r(j))$ маршрута;
- моменты d_{jq}^v, a_{jq}^v , соответствующие отправлению и прибытию согласно расписанию на конечную остановку $v \in \Theta(r(j))$ в цикле $q = 1, \dots, k_j$.

Для депо dep заданы:

- тип cd станций медленной зарядки;
- годовая инвестиционная и эксплуатационная стоимость cst_{cd} одной зарядной станции типа cd ;
- множество J событий зарядки ЭБ парка в ночное время (соответствует множеству индексов ЭБ);
- тариф cst_{ed} на электроэнергию в ночное время.

Каждой конечной остановке $v \in \Theta$ соответствуют:

- подмножество $R(v) \subseteq R$ маршрутов, содержащих эту остановку;
- подмножество $J^v = \bigcup_{r \in R(v)} J_r \subseteq J$ ЭБ, обслуживающих маршруты $r \in R(v)$ и заряжающихся на конечной остановке;
- диапазон $[\gamma st, \delta st]$ времени зарядки в представительный период для всех $j \in J^v$;
- номера циклов μ_j, \dots, g_j для гарантирующего расписания ЭБ $j \in J^v$, в которых в представительный период выполняется зарядка на остановке, где $1 \leq \mu_j \leq g_j \leq k_j, q = \mu_j, \dots, g_j$;
- моменты $\tilde{a}_{jq}^v, \tilde{d}_{jq}^v$ прибытия на остановку и отправления с остановки ЭБ $j \in J^v$ для гарантирующего расписания в диапазоне $[\gamma st, \delta st]$ представительного периода $\gamma st \leq \tilde{a}_{jq}^v < \tilde{d}_{jq}^v \leq \delta st$ в циклах $q = \mu_j, \dots, g_j$;
- тип c зарядных станций быстрой зарядки;
- годовая инвестиционная и операционная стоимость cst_c одной зарядной станции типа c ;
- тариф cst_{est} на электроэнергию в дневное время.

Под конфигурацией зарядной инфраструктуры парка ЭБ рассматриваемой транспортной сети понимаются наборы зарядных станций заданных вариантов cd для депо и c для каждой из конечных остановок.

Множество событий зарядки ЭБ парка в депо в ночное время включает по одному событию зарядки каждого ЭБ парка, причем каждый ЭБ заряжается без переключений один раз на одной из зарядных станций депо. Для каждой конечной остановки рассматривается часть событий зарядки ЭБ, посещающих эту остановку в течение представительного периода дня. В каждом цикле обслуживания маршрута в промежутках между событиями зарядки на конечных остановках маршрута ЭБ циклически выполняют сервисные рейсы перевозки пассажиров от одной конечной остановки к другой. При этом расходуется заряд батарей ЭБ. Требуется, чтобы УЗ каждой батареи не выходил за пределы заданного диапазона значений. Зависимости снижения УЗ (в рамках заданного диапазона) при движении по маршруту и увеличения УЗ при зарядке на зарядной станции задаются линейными убывающими либо линейными возрастающими функциями с известными коэффициентами для ЭБ каждого типа и скорости зарядки. Основываясь на принципе гарантированного результата, распространяем представительный период, характеризующийся наибольшими затратами электроэнергии и наибольшей интенсивностью движения ЭБ по маршрутам, на весь день так, чтобы получаемая для такой модели конфигурация зарядной инфраструктуры обеспечивала функционирование парка ЭБ в любой другой период времени.

Сопоставим вершине dep подграф $Gd = (Ud, Vd)$ событий зарядки каждого ЭБ парка в депо в ночное время суток. Каждой вершине $v \in \Theta$, представляющей конечную остановку, сопоставляется подграф $Gv = (Ust^v, Vst^v)$ событий зарядки ЭБ на соответствующей остановке в течение выделенного представительного периода дня для соответствующего гарантирующего расписания. Каждое событие зарядки ЭБ в вершинах сети представляет однократную зарядку без прерываний на одной из идентичных станций для данной вершины при любом посещении ее ЭБ, обслужива-

ющим проходящий через эту вершину маршрут. Подмножества Ud и Ust^v событий зарядки соответственно подграфов Gd и Gv , $v \in \Theta$, между собой попарно не пересекаются. Множества дуг Vd , Vst^v этих подграфов определяют отношения предшествования, которые требуется определить, для событий зарядки Ud и Ust^v , $v \in \Theta$, соответственно.

Представим множества Ud и Ust^v в следующем виде (см., например, Rogge [8]):

$Ud := 0d \cup fd \cup J$, где $0d$, fd – начальная и конечная вершины событий зарядки в депо соответственно; J – множество событий зарядки ЭБ парка в депо в ночное время;

$Ust^v := 0st^v \cup fst^v \cup Fst^v$, где $0st^v$, fst^v – начальная и конечная вершины событий зарядки на остановке v в выделенном диапазоне соответственно; Fst^v – множество событий зарядки ЭБ на остановке v в течение представительного периода, в который попадают события зарядки ЭБ $j \in J^v$ с номерами их циклов μ_j, \dots, g_j , $\gamma st \leq \tilde{a}_{jq}^v < \tilde{d}_{jq}^v \leq \delta st$, $\mu_j \leq q \leq g_j$, $j \in J^v$.

Предполагается, что для любых $j \in J$ выполняются условия $0 \leq td_j < \gamma st < \delta st < ta_j < td_j + 24$, поэтому множества Ud и Ust^v событий зарядки попарно разделены во времени для любой $v \in \Theta$.

Обозначим $\Omega d^0 = 0d \cup J$, $\bar{\Omega} d = J \cup fd$, $\Omega st^{v0} = 0st^v \cup Fst^v$, $\bar{\Omega} st^v = Fst^v \cup fst^v$.

На рис. 3 показан пример подграфа $Gd = (Ud, Vd)$ событий зарядки в депо для ЭБ типов e_1 и e_2 , возвращающихся в депо с трех маршрутов. Прямоугольники представляют интервалы пребывания соответствующих ЭБ парка в депо, закрашенная их часть соответствует собственно процессу зарядки без прерываний, а дуги из множества Vd отображают порядок выполнения событий зарядки. Число путей в подграфе (два) из исходной вершины $0d$ в конечную вершину fd соответствует числу зарядных станций в депо.

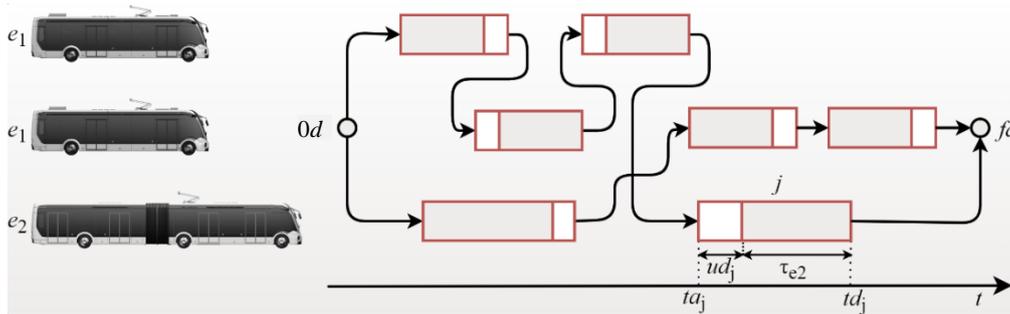


Рис. 3. Пример подграфа событий зарядки в депо для электробусов двух типов
Fig. 3. An example of a subgraph of charging events in a depot for two types of electric buses

Отметим, что в целевой функции наряду с инвестиционными и операционными затратами на зарядную инфраструктуру и затратами на потребленную электроэнергию учитывается также стоимость износа батарей парка ЭБ. При этом допускается полный износ батарей ранее гарантийного срока службы, что влечет необходимость замены изношенных батарей до истечения гарантийного срока эксплуатации ЭБ и, соответственно, увеличение доли затрат на батареи парка ЭБ в целевой функции.

В следующем разделе показано, что стоимость износа батареи ЭБ на любом отрезке траектории ее УЗ может быть определена на основе функции плотности стоимости ее износа в зависимости от значения УЗ и скорости зарядки (разрядки) при нормальной температуре окружающей среды. Кусочно-постоянная функция плотности стоимости износа батареи однозначно определяется по табличной зависимости $ACC_e(DoD, \circ)$ достижимого числа циклов ее зарядки (разрядки) за расчетный период эксплуатации от глубины разряда DoD при фиксированной скорости зарядки (при $\circ = crd_e$ либо $\circ = crst_e$) или разрядки (при $\circ = dr_e$). Температура предполагается равной 25°C (в зимнее время она поддерживается за счет подогрева).

Расчет стоимости износа батареи электробуса. Построим кусочно-постоянные функции $W_e(s, \circ)$ плотности стоимости износа батареи при пропускании через нее при зарядке (разрядке) единицы электроэнергии (1 кВт·ч) в каждой точке $s \in [0, 1]$ траектории УЗ, где $\circ = crd_e$, либо $\circ = crst_e$, либо $\circ = dr_e$. Обозначим имеющиеся табличные значения DoD , для которых определе-

ны $ACC_e(DoD, \circ)$, через D_d и соответствующие значения относительного уровня s заряда батареи через $s_d = 1 - D_d$, $d = 1, \dots, d^0$, где d^0 – число равных подынтервалов отрезка $[0, 1]$, $1 \geq D_1 \geq D_2 \geq \dots \geq D_{d^0} \geq 0$. Обозначим $\Delta_d^s = s_{d+1} - s_d = D_d - D_{d+1}$, $d = 1, \dots, d^0 - 1$.

Для расчета значений $W_e(s_d, \circ)$, $d = 1, \dots, d^0$, функции $W_e(s, \circ)$ плотности используется система линейных уравнений

$$2 \cdot ACC_e(D_d, \circ) \sum_{v=d}^{d^0} W_e(s_v, \circ) \Delta_v^s = cst_b_e, \quad d = 1, \dots, d^0.$$

Определим функции $\tilde{C}_e(s, \circ)$ кумулятивной стоимости износа батареи ЭБ типа e при изменении УЗ от 0 до s , основываясь на кусочно-постоянной функции $W_e(s, \circ)$ плотности стоимости ее износа и учитывая, что по определению $\tilde{C}_e(s, \circ) = \int_0^s W_e(\xi, \circ) d\xi$, $0 \leq s \leq s_{d^0}$.

С учетом кусочно-постоянной на отрезке $[0, s_{d^0}]$ функции $W_e(s, \circ)$ ее кумулятивный аналог $\tilde{C}_e(s, \circ)$ будет кусочно-линейной функцией по s : $\tilde{C}_e(s, \circ) = W_e(0, \circ) (s - 0)$ для $s \in [0, s_1)$, $\tilde{C}_e(s, \circ) = W_e(0, \circ) (s_1 - 0) + W_e(s_1, \circ) (s - s_1)$ для $s \in [s_1, s_2)$ и т. д.

В табл. 3 приведены зависимости достижимого числа циклов $ACC_e(DoD, \circ)$ за период службы литиевых батарей типа LFP от глубины разряда DoD для нескольких значений скорости зарядки в процессе их функционирования, а также значения $W_d = W_e(s_d, \circ)$ функций $W_e(s, \circ)$ и значения $C_d = \tilde{C}_e(s_d, \circ)$ функций $\tilde{C}_e(s, \circ)$. Данные зависимостей $ACC_e(DoD, \circ)$ табл. 3 для $dr_e = 1C$ и $crst_e = 6C$ получены методом экстраполяции на базе данных для скоростей зарядки 0,25 и 0,5C, представленных ОАО «Белкоммунмаш». На рис. 4 и 5 изображены соответствующие значениям $crd_e = 0,5C$, $crst_e = 6C$, $dr_e = 1C$ графики функций $ACC_e(DoD, \circ)$ и $\tilde{C}_e(s, \circ)$ для батарей типа LFP. Следует заметить, что функции $ACC_e(DoD, \circ)$ являются выпуклыми убывающими, а функции $\tilde{C}_e(s, \circ)$ – вогнутыми возрастающими в допустимых диапазонах изменения УЗ.

Таблица 3
 Зависимости достижимого числа циклов зарядки (разрядки), плотности стоимости износа и кумулятивной стоимости износа батареи LFP для различных скоростей зарядки (разрядки)

Table 3
 Dependencies of an attainable number of charging (discharging) cycles, the density of the wear cost and the cumulative cost of wear of the LFP battery for different charging (discharging) rates

DoD	УЗ	$crd_e = 0,5C$			$crst_e = 6C$			$dr_e = 1C$		
D_d	s_d	$ACC_e(D_d)$	W_d	C_d	$ACC_e(D_d)$	W_d	C_d	$ACC_e(D_d)$	W_d	C_d
0	1	–	–	3,686	–	–	16,125	–	–	5,292
0,1	0,9	550 000	0,188	3,667	210 000	0,491	16,076	430 000	0,240	5,268
0,2	0,8	170 000	0,419	3,625	88 000	0,681	16,008	145 000	0,472	5,221
0,3	0,7	63 000	1,031	3,522	24 000	3,127	15,695	50 000	1,352	5,086
0,4	0,6	30 000	1,802	3,342	9 600	6,450	15,050	22 000	2,627	4,823
0,5	0,5	17 000	2,631	3,079	6 000	6,450	14,405	13 000	3,248	4,498
0,6	0,4	9 700	4,569	2,622	3 000	17,200	12,685	7 200	6,395	3,859
0,7	0,3	6 000	6,561	1,966	1 700	26,306	10,054	4 400	9,121	2,947
0,8	0,2	4 000	8,600	1,106	900	53,961	4,658	2 750	14,073	1,540
0,9	0,1	3 100	7,490	0,357	700	32,762	1,382	2 100	11,616	0,378
1	0	2 800	3,569	0	640	13,821	0	1 950	3,780	0

Отличие рассматриваемой модели расчета стоимости износа батареи от модели, предложенной в работе [6] и уточненной в работе [10], заключается в учете зависимости функции плотности стоимости износа от скорости зарядки (разрядки) батареи.

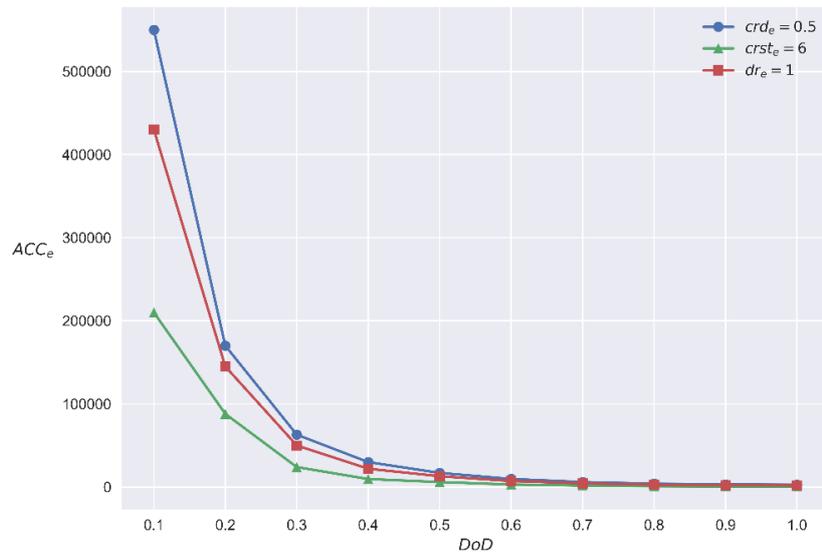


Рис. 4. Зависимости достижимого числа циклов зарядки (разрядки) батарей LFP от глубины разряда DoD

Fig. 4. Dependencies of an attainable number of charging (discharging) cycles of LFP batteries on the depth of discharge DoD

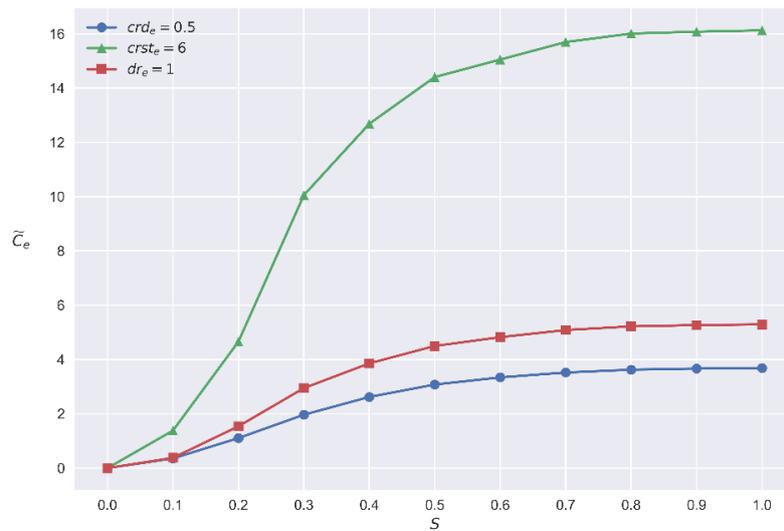


Рис. 5. Зависимости $\tilde{C}_e(s, \circ)$ для батарей LFP

Fig. 5. Dependencies $\tilde{C}_e(s, \circ)$ for LFP batteries

Формулировка задачи оптимизации зарядной инфраструктуры. Учитывая взаимно-однозначное соответствие между ЭБ j , их типами $e(j)$ и обслуживаемыми ими маршрутами $r(j)$, введем следующие обозначения: $E_j = E_{e(j)}$, $\Theta(j) = \Theta(r(j))$, $v_s(j) = v_s(r(j))$, $v_p(j) = v_p(r(j))$, $v_f(j) = v_f(r(j))$, $\underline{s}_j = \underline{s}_{e(j)}$, $\bar{s}_j = \bar{s}_{e(j)}$, $\tau_j^{\max} = \tau_{e(j)}^{\max}$, $t_j^{\max} = t_{e(j)}^{\max}$.

Контролируемые и зависимые переменные. Контролируемыми переменными модели являются: длительности зарядки ЭБ парка в депо и на конечных остановках, отношения порядка событий зарядки ЭБ на зарядных станциях депо в ночное время и на остановках в выделенном периоде, а также величины задержки начала зарядки ЭБ относительно моментов их прихода в место зарядки.

Зависимыми от контролируемых переменных являются УЗ ЭБ в контрольных точках маршрутов и количества станций зарядки в вершинах транспортной сети. Контрольные точки маршрутов расположены при выходе из (приходе в) депо и при приходе на (выходе с) конечную остановку.

Таким образом, к контролируемым переменным для депо относятся:

- длительности $\tau_j = \tau_{e(j),r(j)}$ зарядки в депо в ночное время для каждого ЭБ $j \in J$ типа $e(j)$, обслуживающего маршрут $r(j)$;
- задержки ud_j начала события зарядки ЭБ $j \in J$ в депо относительно момента ta_j прихода в депо в конце рабочего дня;
- индикаторы $zd_{ms}, zd_{ms} = 1$, если и только если событие s зарядки в депо выполняется после события зарядки m для любых $s, m \in Ud, s \neq m$.

Зависимые переменные для депо включают: УЗ sd_j в момент выхода в депо, $j \in J$; УЗ sa_j в момент прихода в депо, $j \in J$; число Kd зарядных станций типа cd .

Для описания контролируемых переменных множества Fst^v событий зарядки ЭБ $j \in J^v$ на конечной остановке $v \in \Theta$ в течение представительного периода с учетом гарантирующего расписания введем отображение $\rho = \omega^v(j, q): Fst^v \rightarrow \{1, 2, \dots, |Fst^v|\}$ множества Fst^v на их номера. (Далее будем отождествлять события зарядки с их номерами ρ .) При этом каждому событию зарядки $\rho \in Fst^v$ в диапазоне $[\gamma st, \delta st]$ представительного периода соответствует пара $(j(\rho), q(\rho))$, $j(\rho) \in J^v, q(\rho) \in \{\mu_j, \dots, g_j\}$, для любого $\rho = \omega^v(j, q) \in Fst^v$ выполняются условия $\gamma st \leq \tilde{a}_{jq}^v < \tilde{d}_{jq}^v \leq \delta st$.

Положим $\omega^{-1v}(\rho) = (j(\rho), q(\rho))$, $j(\rho) \in J^v, q(\rho) \in \{\mu_j, \dots, g_j\}$. Соответственно получаем:

- момент a_ρ^v $q(\rho)$ -го прихода ЭБ $j(\rho)$ на остановку $v \in \Theta(j)$, $\rho \in Fst^v$;
- момент d_ρ^v $q(\rho)$ -го выхода ЭБ $j(\rho)$ с остановки $v \in \Theta(j)$, $\rho \in Fst^v$;
- задержку u_ρ^v начала ρ -го события зарядки на остановке v относительно момента a_ρ^v прихода ЭБ $j(\rho)$ на эту остановку согласно гарантирующего расписания в течение представительного периода, $\rho \in Fst^v$.

К контролируемым переменным для конечной остановки $v \in \Theta$ при заданном гарантирующем расписании моментов $\tilde{a}_{jq}^v, \tilde{d}_{jq}^v$ прибытия и отправления $j \in J^v$ в циклах $q = \mu_j, \dots, g_j$ относятся:

- длительность $t_j^v = t_{e(j),r(j)}^v$ зарядки для каждого ЭБ j типа $e(j)$ на конечной остановке $v \in \Theta(j)$ маршрута $r(j)$, одинаковая для любого цикла $q = 1, \dots, k_j$;
- задержка u_s^v относительно момента \tilde{a}_{jq}^v прибытия ЭБ $j = j(s)$ на остановку $v \in \Theta(j)$ в цикле $q \in \{\mu_j, \dots, g_j\}$, где $(j(s), q(s)) = \omega^{-1v}(s), s \in Fst^v$;
- индикаторы $zst_{sp}^v, zst_{sp}^v = 1$, если и только если событие зарядки p на остановке v выполняется после события зарядки s для любых $s, p \in Ust^v, s \neq p$.

Зависимые переменные для конечной остановки $v \in \Theta$ включают:

- УЗ sa_{jq}^v при прибытии $j \in J^v$ в q -м цикле на остановку $v \in \Theta(j)$, $q = 1, \dots, k_j$;
- УЗ sd_{jq}^v при выходе $j \in J^v$ в q -м цикле с остановки $v \in \Theta(j)$, $q = 1, \dots, k_j$;
- число Kst^v станций зарядки типа c .

Система ограничений задачи. Система ограничений, задающая множество допустимых значений контролируемых переменных задачи, включает: условия длительности зарядки ЭБ на зарядных станциях соответствующих вершин сети, условия порядка выполнения зарядки для ЭБ на каждой станции зарядки, ограничения на УЗ каждого ЭБ парка в процессе функционирования.

Запишем условия для длительностей и порядка выполнения событий зарядки в депо:

$$ta_j + ud_j + \tau_j \leq ta_i + ud_i + M(1 - zd_{ji}), i, j \in J, i \neq j, \quad (1)$$

$$\sum_{j \in \Omega d, i \neq j} zd_{ij} - \sum_{j \in \Omega d^0, i \neq j} zd_{ji} = 0 \text{ для любых } i \in J, \quad (2)$$

$$\sum_{i \in \Omega d^0, i \neq j} z d_{ij} = 1 \text{ для любых } j \in J, \quad (3)$$

$$\sum_{i \in \Omega d, i \neq j} z d_{ji} = 1 \text{ для любых } j \in J, \quad (4)$$

$$t a_j + u d_j + \tau_j \leq t d_j + 24, j \in J. \quad (5)$$

Условия (1) обеспечивают отсутствие пересечения различных событий зарядки разных ЭБ по времени на одной зарядной станции в депо. Условия (2) являются потоковыми для сети событий зарядки в депо. Условия (3) обеспечивают для каждого ЭБ одно предшествующее событие зарядки, в то время как условия (4) – одно последующее событие зарядки в депо. Условия (5) обеспечивают завершение зарядки любого ЭБ в депо до момента выхода его на маршрут на следующий день.

Далее для краткости изложения введем дополнительное обозначение $t_\eta^v = t_{j(\eta)}^v$ для любой остановки $v \in \Theta(j)$, где $\omega^{-1v}(\eta) = (j(\eta), q(\eta))$, $j(\eta) \in J$, $q(\eta) \in \{\mu, \dots, g_j\}$, – отображение, обратное для отображения $\eta = \omega^v(j, q) \in Fst^v$.

Приведем ограничения на длительности и порядок выполнения событий зарядки на остановке $v \in \Theta$:

$$a_\eta^v + u_\eta^v + t_\eta^v \leq a_\mu^v + u_\mu^v + M(1 - zst_{\eta\mu}^v), \eta, \mu \in Fst^v, \eta \neq \mu, \quad (6)$$

$$\sum_{\eta \in \Omega st^v, \mu \neq \eta} zst_{\mu\eta}^v - \sum_{\eta \in \Omega st^v, \mu \neq \eta} zst_{\eta\mu}^v = 0 \text{ для любых } \mu \in Fst^v, \quad (7)$$

$$\sum_{\eta \in \Omega st^v, \mu \neq \eta} zst_{\eta\mu}^v = 1 \text{ для любых } \mu \in Fst^v, \quad (8)$$

$$\sum_{\eta \in \Omega st^v, \mu \neq \eta} zst_{\mu\eta}^v = 1 \text{ для любых } \mu \in Fst^v, \quad (9)$$

$$a_\eta^v + u_\eta^v + t_\eta^v \leq d_\eta^v \text{ для любых } \eta \in Fst^v. \quad (10)$$

Условия (6) обеспечивают отсутствие пересечения разных событий зарядки различных ЭБ по времени на одной зарядной станции остановки v . Условия (7) являются потоковыми для сети событий зарядки остановки v . Соотношения (8) обеспечивают для каждого ЭБ ровно одно предшествующее событие зарядки, в то время как условие (9) – только одно последующее событие зарядки остановки v . Ограничения (10) обуславливают завершение зарядки любого ЭБ на остановке v до заданного момента выхода на маршрут.

Введем дополнительные обозначения. Нулевым (начальным) циклом будем называть вспомогательный рейс из депо dep до стартовой конечной остановки $v_s(j)$ перед первым выходом ЭБ на маршрут, (k_j+1) -м циклом – вспомогательный рейс возвращения ЭБ в депо с остановки $v_f(j)$ после дневной смены, где $f \in \{s, p\}$.

Величины sl_{j0} , sl_{j, k_j+1} , $sl_{jq}(\xi, \zeta)$ потери заряда ЭБ j во вспомогательных рейсах при $q = 0, k_j+1$, а также в сервисных рейсах (ξ, ζ) , $\xi, \zeta \in \Theta(j)$ циклов $q = 1, \dots, k_j$ маршрута $r(j)$ для случая двух конечных остановок определяются следующим образом:

$$sl_{j0} = sl_{e(j)} L_{dep, v_s(j)};$$

$$sl_j(v_s(j), v_p(j)) = sl_{jq}(v_s(j), v_p(j)) = sl_{e(j)} L_{v_s(j), v_p(j)} \text{ для рейса } (v_s(j), v_p(j)) \text{ в любом цикле } q \in \{1, \dots, k_j\};$$

$$sl_j(v_p(j), v_s(j)) = sl_{jq}(v_p(j), v_s(j)) = sl_{e(j)} L_{v_p(j), v_s(j)} \text{ для рейса } (v_p(j), v_s(j)) \text{ в любом цикле } q \in \{1, \dots, k_j\};$$

$$sl_{j, k_j+1} = sl_{e(j)} L_{v_f(j), dep}.$$

Потеря заряда sl_{jq} ЭБ j в цикле $q = 0, 1, \dots, k_j+1$ для случая одной конечной остановки $v_s(j)$ маршрута $r(j)$: $sl_{j0} = sl_{e(j)} L_{dep, v_s(j)}$, $sl_j = sl_{jq} = sl_{jq} = sl_{e(j)} L_{v_s(j), v_s(j)}$, $q = 1, \dots, k_j$, $sl_{j, k_j+1} = sl_{e(j)} L_{v_s(j), dep}$.

Ограничения на УЗ для ЭБ $j \in J^v$ типа $e(j)$ на остановке $v = v_\sigma(j) \in \Theta(j)$, $\sigma \in \{s, p\}$:

$$sd_{jq}^v = sa_{jq}^v + crst_{e(j)} \cdot t_j^v, \quad q = 1, \dots, k_j, j \in J^v, \quad (11)$$

$$sa_{jq}^{v_s(j)} = sd_j - \frac{1}{E_j} (sl_{j0} + (q-1)(sl_j(v_s(j), v_p(j)) + sl_j(v_p(j), v_s(j))) + crst_{e(j)} \cdot (q-1)(t_j^{v_s(j)} + t_j^{v_p(j)}),$$

$$q = 1, \dots, k_j, j \in J^{v_s(j)}, \quad (12)$$

$$sa_{jq}^{v_p(j)} = sd_j - \frac{1}{E_j} (sl_{j0} + q \cdot sl_j(v_s(j), v_p(j)) + (q-1)sl_j(v_p(j), v_s(j))) + crst_{e(j)} [q t_j^{v_s(j)} + (q-1)t_j^{v_p(j)}],$$

$$q = 1, \dots, k_j, j \in J^{v_p(j)}, \quad (13)$$

$$\underline{s}_j \leq sa_{jq}^v \leq \bar{s}_j, \quad q = 1, \dots, k_j, j \in J^v, \quad (14)$$

$$\underline{s}_j \leq sd_{jq}^v \leq \bar{s}_j, \quad q = 1, \dots, k_j, j \in J^v. \quad (15)$$

Условия (11) устанавливают УЗ батареи любого ЭБ при выходе с остановки v в цикле q его обслуживания равным ее УЗ в момент прихода в этом цикле на остановку, увеличенным на долю УЗ, полученную при однократной зарядке на ней. Условия (12), (13) устанавливают величину УЗ батареи любого ЭБ в момент прихода на остановку $v_s(j)$ (либо $v_p(j)$) $\in \Theta(j)$ в цикле $q = 1, \dots, k_j$ равной УЗ в момент выхода ЭБ из депо за вычетом доли УЗ, израсходованной на маршруте за q циклов и увеличенной на долю УЗ, полученную в результате зарядок в этих q циклах на конечных остановках маршрута. Ограничения (14) обеспечивают для каждого ЭБ допустимость УЗ при приходе на остановку в цикле $q = 1, \dots, k_j$, в то время как ограничения (15) – допустимость УЗ при выходе с остановки в цикле $q = 1, \dots, k_j$.

Ограничения на УЗ для ЭБ $j \in J$ в депо:

$$sd_j = sa_j + crd_{e(j)} \cdot \tau_j, \quad j \in J, \quad (16)$$

$$sa_j = sd_j - \frac{1}{E_j} (sl_{j0} + sl_{j, k_j+1} + k_j (sl_j(v_s(j), v_p(j)) + sl_j(v_p(j), v_s(j))) + crst_{e(j)} \cdot k_j (t_j^{v_s(j)} + t_j^{v_p(j)}), \quad j \in J, \text{ при } f = s, \quad (17)$$

$$sa_j = sd_j - \frac{1}{E_j} (sl_{j0} + sl_{j, k_j+1} + k_j \cdot sl_j(v_s(j), v_p(j)) + (k_j - 1) \cdot sl_j(v_p(j), v_s(j))) + crst_{e(j)} \cdot ((k_j - 1) \cdot t_j^{v_s(j)} + k_j t_j^{v_p(j)}), \quad j \in J, \text{ при } f = p, \quad (18)$$

$$\underline{s}_j \leq sa_j \leq \bar{s}_j, \quad j \in J, e(j) \in EB, \quad (19)$$

$$\underline{s}_j \leq sd_j \leq \bar{s}_j, \quad j \in J, e(j) \in EB. \quad (20)$$

Согласно условиям (16) величина УЗ батареи любого ЭБ при выходе из депо должна равняться УЗ батареи в момент прихода в депо, увеличенному на величину УЗ, которая получается при его зарядке в депо. Условия (17) устанавливают величину УЗ батареи любого ЭБ в момент прихода в депо с конечной остановки $v_f(j) = v_s(j)$ равной величине УЗ при выходе из депо за вычетом доли УЗ, израсходованной на маршруте и увеличенной на долю УЗ, которая получается в результате всех зарядок ЭБ в течение дня на конечных остановках маршрута. В соответствии с условием (18) величина УЗ батареи любого ЭБ в момент прихода в депо с конечной остановки $v_f(j) = v_p(j)$ равна величине УЗ при выходе из депо за вычетом доли УЗ, израсходованной на маршруте и увеличенной на долю УЗ, которая получается в результате всех зарядок ЭБ

в течение дня на конечных остановках маршрута. Ограничения (19) обеспечивают для каждого ЭБ допустимость УЗ при приходе в депо, в то время как ограничения (20) – допустимость УЗ при выходе из депо.

Ограничения неотрицательности и целочисленности переменных:

$$0 \leq ud_j, j \in J; \quad (21)$$

$$0 \leq u_\eta^v, \eta \in Fst^v, v \in \Theta; \quad (22)$$

$$0 \leq \tau_j \leq \tau_j^{\max}, j \in J; \quad (23)$$

$$0 \leq t_j^v \leq t_j^{\max}, j \in J; \quad (24)$$

$$zd_{ij} \in \{0, 1\}, i, j \in J, i \neq j; \quad (25)$$

$$zst_{\eta\mu}^v \in \{0, 1\}, \eta, \mu \in Fst^v, \eta \neq \mu; \quad (26)$$

$$zst_{\eta\mu}^v = 0 \text{ при } d_\mu^v \leq a_\eta^v, \eta, \mu \in Fst^v, \eta \neq \mu; \quad (27)$$

$$sd_j = \bar{s}_j, j \in J, \quad (28)$$

где $M \gg ta_j, j \in J, M \gg a_\eta^v, \eta \in Fst^v, v \in \Theta$.

Ограничения (21) и (22) устанавливают неотрицательность значений переменных ud_j ($j \in J$) и u_η^v ($\eta \in Fst^v, v \in \Theta$) задержки событий зарядки ЭБ в депо и на конечных остановках соответственно. Ограничения (23) и (24) устанавливают допустимые диапазоны значений длительностей зарядки $\tau_j, t_j^v, j \in J$, ЭБ в депо и на остановках соответственно. Ограничения (25) и (26) определяют диапазоны значений бинарных переменных zd_{ij} и $zst_{\eta\mu}^v$, обозначающих, упорядочены ли события зарядки i, j в депо и события зарядки η, μ на остановке v соответственно. Ограничения (27) устанавливают отсутствие предшествования события зарядки η событию μ на остановке v , если время a_η^v прибытия ЭБ, соответствующего событию η , на остановку превосходит время отправления ЭБ, соответствующего событию μ . Условия (28) обеспечивают восстановление уровня заряда любого ЭБ в депо перед выходом на маршрут на следующий день.

Ограничения для случая одной конечной остановки имеют более простой вид, чем ограничения (6)–(28), и в настоящей статье опущены.

Обозначим для краткости:

$$\boldsymbol{\tau} = (\tau_j | j \in J), \boldsymbol{t} = (t_j^v | j \in J, v \in \Theta), \boldsymbol{z}d = (zd_{ij} | i, j \in J, i \neq j), \boldsymbol{z}st = (zst_{\eta\mu}^v | \eta, \mu \in Fst^v, \eta \neq \mu, v \in \Theta), \boldsymbol{u}d = (ud_j | j \in J),$$

$$\boldsymbol{u}st = (u_\eta^v | \eta \in Fst^v, v \in \Theta), \boldsymbol{s}d = (sd_j, sd_{jq}^v | q = 1, \dots, k_j + 1, j \in J, v \in \Theta(j)), \boldsymbol{s}a = (sa_j, sa_{jq}^v | q = 0, \dots, k_j, j \in J, v \in \Theta(j)).$$

Целевая функция. Целевая функция представляет собой суммарную суточную стоимость следующих компонент:

– электроэнергии, расходуемой в течение дня на станциях быстрой зарядки конечных остановок,

$$cst_est \left[\sum_{j \in J} \sum_{v_s(j)=v_f(j), v_p(j) \in \Theta(j)} E_j crst_{e(j)} ((k_j + 1)t_j^{v_s(j)} + k_j t_j^{v_p(j)}) + \right.$$

$$\left. + \sum_{j \in J} \sum_{v_s(j), v_p(j)=v_f(j) \in \Theta(j)} E_j crst_{e(j)} (k_j + 1)(t_j^{v_s(j)} + t_j^{v_p(j)}) \right],$$

где первое слагаемое представляет собой стоимость электроэнергии, потраченной на зарядку ЭБ на конечных остановках маршрутов, которые содержат только полные сервисные циклы ($f = s$), а второе слагаемое – стоимость электроэнергии, потраченной на зарядку ЭБ на конечных остановках маршрутов с неполным последним сервисным циклом ($f = p$);

– электроэнергии, расходуемой в ночное время в депо на станциях медленной зарядки,
 $cst_ed \sum_{j \in J} crd_{e(j)} \tau_j$;

– использованных зарядных станций в депо и на остановках $\frac{cst_cd \sum_{m \in J} zd_{0d,m}}{365} + \frac{cst_c \sum_{v \in \Theta} \sum_{m \in Fst^v} zst_{0st_v,m}}{365}$;

– износа батарей парка ЭБ

$$\begin{aligned}
 FB(\mathbf{sd}, \mathbf{sa}) = & \sum_{j \in J} (\tilde{C}_e(crd_{e(j)}, sd_j) - \tilde{C}_e(crd_{e(j)}, sa_j)) + \\
 & + \sum_{j \in J} \sum_{v \in \Theta(j)} \sum_{q=1}^{k_j+1} (\tilde{C}_e(crst_{e(j)}, sd_{jq}^v) - \tilde{C}_e(crst_{e(j)}, sa_{j,q-1}^v)) + \sum_{j \in J} \sum_{v \in \Theta(j)} \sum_{q=1}^{k_j} (\tilde{C}_e(dr_{e(j)}, sd_{jq}^v) - \tilde{C}_e(dr_{e(j)}, sa_{jq}^v)) + \\
 & + \sum_{j \in J} (\tilde{C}_e(dr_{e(j)}, sd_j) - \tilde{C}_e(dr_{e(j)}, sa_{j1}^{v_s(j)})) + \tilde{C}_e(dr_{e(j)}, sd_{j,k_j+1}^{v_f(j)}) - \tilde{C}_e(dr_{e(j)}, sa_j),
 \end{aligned}$$

где первое слагаемое представляет суточную долю стоимости износа батарей при зарядке всех ЭБ в депо, второе слагаемое – суточную долю стоимости износа батарей при зарядке всех ЭБ на конечных остановках обслуживаемых ими маршрутов, третье слагаемое – суточную долю стоимости износа батарей при расходе энергии в сервисных рейсах, а четвертое слагаемое – суточную долю стоимости износа батарей ЭБ парка во вспомогательных рейсах при выходе и возвращении ЭБ в депо;

– штрафов за задержку начала зарядки ЭБ после прибытия в депо и на остановки $\phi \sum_{j \in J} ud_j + \phi \sum_{v \in \Theta} \sum_{\eta \in Fst^v} u_\eta^v$, где ϕ – коэффициент штрафа.

Таким образом, целевая функция задачи оптимизации зарядной инфраструктуры имеет вид

$$\begin{aligned}
 \Phi(\boldsymbol{\tau}, \mathbf{t}, \mathbf{zd}, \mathbf{zst}, \mathbf{ud}, \mathbf{ust}) = & FB(\mathbf{sd}(\boldsymbol{\tau}, \mathbf{t}), \mathbf{sa}(\boldsymbol{\tau}, \mathbf{t})) + cst_ed \sum_{j \in J} crd_{e(j)} \tau_j + \\
 & + cst_est [\sum_{j \in J} \sum_{v_s(j)=v_f(j), v_p(j) \in \Theta(j)} E_j crst_{e(j)} ((k_j+1)t_j^{v_s(j)} + k_j t_j^{v_p(j)}) + \\
 & + \sum_{j \in J} \sum_{v_s(j), v_p(j)=v_f(j) \in \Theta(j)} E_j crst_{e(j)} (k_j+1)(t_j^{v_s(j)} + t_j^{v_p(j)})] + \frac{cst_cd \sum_{m \in J} zd_{0d,m}}{365} + \frac{cst_c \sum_{v \in \Theta} \sum_{m \in Fst^v} zst_{0st_v,m}}{365} + \\
 & + \phi \sum_{j \in J} ud_j + \phi \sum_{v \in \Theta} \sum_{\eta \in Fst^v} u_\eta^v.
 \end{aligned}$$

С учетом вышеизложенного формулировка рассматриваемой задачи оптимизации конфигурации зарядной инфраструктуры определяется выражением

$$\min \Phi(\boldsymbol{\tau}, \mathbf{t}, \mathbf{zd}, \mathbf{zst}, \mathbf{ud}, \mathbf{ust}) \quad (29)$$

при выполнении ограничений (1)–(28).

Задачу (1)–(29) будем обозначать задачей **A**. Учитывая, что $FB(\mathbf{sd}, \mathbf{sa})$ является кусочно-линейной функцией переменных векторов \mathbf{sd} и \mathbf{sa} , путем введения дополнительных переменных задача **A** может быть сведена к задаче целочисленного линейного программирования. Для решения задачи **A** может быть использовано стандартное ПО типа IBM ILOG CPLEX.

Закключение. Разработана математическая модель задачи оптимизации смешанной зарядной инфраструктуры для парка ЭБ нескольких типов. Смешанная зарядная инфраструктура совмещает технологию медленной зарядки батарей ЭБ в депо при использовании небольшой мощности зарядки с технологией быстрой их подзарядки на конечных остановках маршрутов при использовании большой мощности зарядки. Особенности предложенной математической модели являются применение принципа гарантированного результата для оценки условий функционирования зарядной инфраструктуры парка ЭБ в течение дня и учет в целевой функции наряду со стоимостями зарядной инфраструктуры и потребляемой электроэнергии также стоимости износа батареи каждого ЭБ парка. Применение принципа гарантированного результата сводится к выделению представительного периода дня с наиболее интенсивными пассажиропотоком

и расходом ЭБ электроэнергии, распространению характеристик этого периода на весь день, а также использованию гарантирующего расписания прихода ЭБ в представительный период на каждую конечную остановку с наибольшей потребностью в количестве параллельных зарядных станций. Условия функционирования парка ЭБ и его зарядной инфраструктуры ужесточаются таким образом, чтобы обеспечивать работоспособность ЭБ в любой другой период с менее интенсивными условиями.

В дальнейшем целесообразно исследовать влияние на стоимость зарядной инфраструктуры учета различных по интенсивности расхода электроэнергии режимов эксплуатации ЭБ в течение дня. Кроме того, представляют интерес для исследования возможности выбора емкостей батарей ЭБ и величин мощности их зарядки в депо и на остановках маршрутов из заданных множеств вариантов, а также вопросы устойчивости решения задачи к возмущениям в исходных данных.

Вклад авторов. *Б. М. Розин* предложил структуру и состав компонент математической модели оптимизации смешанной зарядной инфраструктуры парка ЭБ, *И. А. Шатерник* принял участие в обсуждении, уточнении и верификации компонент модели, а также в подготовке иллюстративного материала.

References

1. Gao Z., Lin Z., LaClair T. J., Liu C., Li J.-M., ..., Ward J. Battery capacity and recharging needs for electric buses in city transit service. *Energy*, 2017, vol. 122, pp. 588–600.
2. Olsson O., Grauers A., Pettersson S. Method to analyze cost effectiveness of different electric bus systems. *EVS29 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, Montreal, Quebec, Canada, June 2016*. Montreal, 2016, pp. 1–12.
3. Lin Z. Optimizing and diversifying electric vehicle driving range for US drivers. *Transportation Science*, 2014, vol. 48(4), pp. 635–650.
4. Pelletier S., Jabali O., Mendoza J. E., Laporte G. The electric bus fleet transition problem. *Transportation Research Part C: Emerging Technologies*, 2019, vol. 109, pp. 174–193.
5. Goehlich D., Fay T.-A., Park S. Conceptual design of urban e-bus systems with special focus on battery technology. *Proceedings of the 22nd International Conference on Engineering Design (ICED19), Delft, The Netherlands, 5–8 August 2019*. Delft, 2019. <https://doi.org/10.1017/dsi.2019.289>
6. Han S. K. A practical battery wear model for electric vehicles charging applications. *Applied Energy*, 2014, vol. 113, pp. 1100–1108.
7. Guschinsky N., Kovalyov M., Rozin B., Brauner N. Fleet and charging infrastructure decisions for fast-charging city electric bus service. *Computers and Operations Research*, 2021, vol. 135, p. 105449. <https://doi.org/10.1016/j.cor.2021.105449>
8. Rogge M., van der Hurk E., Larsen A., Sauer D. U. Electric bus fleet size and mix problem with optimization of charging infrastructure. *Applied Energy*, 2018, vol. 211, pp. 282–295.
9. Millner A. Modeling lithium ion battery degradation in electric vehicles. *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*. Waltham, MA, 2010, pp. 349–356.
10. Pelletier S., Jabali O., Laporte G. Charge scheduling for electric freight vehicles. *Transportation Research Part B*, 2018, vol. 115, pp. 246–269.

Информация об авторах

Розин Борис Матвеевич, кандидат технических наук, заведующий сектором, Объединенный институт проблем информатики Национальной академии наук Беларуси.

<http://orcid.org/0000-0003-3814-6569>

E-mail: rozin@newman.bas-net.by

Шатерник Илья Андрианович, инженер-программист, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: shatoby.i@gmail.com

Information about the authors

Boris M. Rozin, Ph. D. (Eng.), Head of the Sector, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

<http://orcid.org/0000-0003-3814-6569>

E-mail: rozin@newman.bas-net.by

Ilya A. Shatarnik, Engineer-programmer, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: shatoby.i@gmail.com

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

INFORMATION TECHNOLOGIES



УДК 519.711
<https://doi.org/10.37661/1816-0301-2022-19-2-85-99>

Оригинальная статья
Original Paper

Информационная система для онтологического моделирования предметных областей

М. Н. Бухаров

*Федеральное государственное бюджетное учреждение науки
«Институт радиотехники и электроники им. В. А. Котельникова
Российской академии наук»,
ул. Моховая, 11, Москва, 125009, Россия
E-mail: oberonco@mail.ru*

Аннотация

Цели. Рассмотрено создание онтологий предметных областей. Целью исследования явилась разработка математической модели онтологии и на ее основе – информационной системы для онтологического моделирования. Поставлена задача снижения трудоемкости онтологического моделирования.

Методы. В качестве методов исследования использовались теория систем гибридного интеллекта, теория множеств, элементы математической логики, методы разработки информационных систем, сравнительный анализ информационных систем, неформальный анализ практического использования информационной системы.

Результаты. Разработана математическая модель онтологии, использующая понятие метаобъекта. Онтологическое моделирование на основе этой модели включает спецификацию, концептуализацию и формализацию. На стадии спецификации построен глоссарий терминов. На стадии концептуализации определены объекты предметной области и их иерархия, затем выявлены связи между объектами. На стадии формализации определены метаобъекты и связи между ними, которые соответствуют объектам и связям между объектами. Это и есть онтология предметной области. На стадии актуализации определены параметры объектов предметной области и их значения, классы, подклассы и экземпляры классов. Параметры, значения параметров, классы, подклассы и экземпляры классов реализуются в онтологии в виде метаобъектов соответствующих типов. Разработана информационная система с уникальной архитектурой, а именно система гибридного интеллекта для автоматизации онтологического моделирования.

Заключение. Проведен сравнительный анализ разработанной информационной системы и применяемых сегодня систем для создания онтологий. Анализ показал, что разработанная автором информационная система по большинству параметров не уступает рассмотренным системам и является одновременно более простой при использовании и расширении. Предложенные математическая модель онтологии и информационная система для онтологического моделирования предметных областей апробированы при практическом создании онтологии по экологии. На основе проведенного сравнительного анализа и неформального анализа практического использования сделан вывод о том, что онтологическое моделирование с помощью разработанной информационной системы снижает трудоемкость и ускоряет сроки создания онтологий.

Ключевые слова: онтология, база знаний, онтологическое моделирование, система гибридного интеллекта, экология

Для цитирования. Бухаров, М. Н. Информационная система для онтологического моделирования предметных областей / М. Н. Бухаров // Информатика. – 2022. – Т. 19, № 2. – С. 85–99.
<https://doi.org/10.37661/1816-0301-2022-19-2-85-99>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 28.02.2022

Подписана в печать | Accepted 04.04.2022

Опубликована | Published 29.06.2022

Information system for ontological modelling the subject areas

Mikhail N. Bukharov

*Federal State Budgetary Institution of Science
"Kotelnikov Institute of Radioengineering and Electronics
of Russian Academy of Sciences",
st. Mokhovaya, 11, Moscow, 125009, Russia
E-mail: oberonco@mail.ru*

Abstract

Objectives. The creation of ontologies of subject areas is considered. The goal is to develop a mathematical model of ontology and the information system for ontological modeling. The task is to reduce the complexity of ontological modeling.

Methods. As research methods, the theory of hybrid intelligence systems, the theory of sets, elements of mathematical logic, methods for developing the information systems, comparative analysis of information systems, informal analysis of information system were used.

Results. Mathematical model of ontology using the concept of metaobject is developed. Ontological modelling based on this model involves specification, conceptualization and formalization. A glossary of terms is being built at the specification stage. At the conceptualization stage, objects in the subject area and their hierarchy are defined, and then connections between objects are identified. In the formalization stage, the metaobjects and the relationships between metaobjects that correspond to objects and the relationships between objects were defined. This is considered as the ontology of the subject area. During the actualization stage, the parameters of subject area objects and their values, classes, subclasses, and instances of classes were defined. Parameters, parameter values, classes, subclasses, and instances of classes are implemented in ontology as metaobjects of relative types. An information system with a unique architecture has been developed, namely a hybrid intelligence system for the automation of ontological modelling.

Conclusion. The article conducts a comparative analysis of the developed information system with the systems used today for creation of ontologies. The analysis showed that the information system developed by paper author in most parameters is not inferior to considered systems and at the same time easier to use and expand. The mathematical model of ontology and the information system for ontological modeling of subject areas, developed by author, are tested in practical creation of ontology on ecology. On the basis of the conducted comparative analysis and informal analysis of practical use, it is concluded that ontological modeling with the help of the information system developed by author reduces the labor intensity and decreases the time of ontologies creation.

Keywords: ontology, knowledge base, ontological modeling, hybrid intelligence system, ecology

For citation. Bukharov M. N. *Information system for ontological modelling the subject areas*. Informatika [Informatics], 2022, vol. 19, no. 2, pp. 85–99 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-2-85-99>

Conflict of interest. The author declare of no conflict of interest.

Введение. В настоящее время онтологическое моделирование предметных областей активно развивается и «вширь» [1–4], и «вглубь» [5–7], между тем трудоемкость онтологического моделирования остается высокой. В работе ставилась задача снизить трудоемкость онтологического моделирования предметных областей за счет применения систем гибридного интеллекта.

В статье приведены основные положения разработанной ранее теории систем гибридного интеллекта (книги [8, 9], статьи [10–12]) и предложена разработанная автором математическая модель онтологии и базы знаний. Деятельность по онтологическому моделированию предметной области представлена как система гибридного интеллекта. Описаны основные функциональные подсистемы разработанной системы гибридного интеллекта «Онтологическое моделирование предметной области», выполняющие сбор, регистрацию, хранение и использование информации, которая необходима для создания и применения онтологии. Онтологическое моделирование предметной области включает спецификацию, концептуализацию и формализацию. Результатом онтологического моделирования является онтология предметной области. Для превращения онтологии в базу знаний необходимо осуществить ее актуализацию. После проведения работ на стадии актуализации онтология становится базой знаний, однако «жесткой» границы между онтологией и базой знаний нет.

Рассмотрено онтологическое моделирование экологии с использованием разработанной автором системы гибридного интеллекта.

1. Основные положения теории систем гибридного интеллекта. Система гибридного интеллекта (СГИ) – это информационная система с уникальной архитектурой, которая создается для управления некоторой сложной деятельностью. Для описания архитектуры СГИ воспользуемся понятием «функционально-структурная схема информационной системы»¹, введенным автором в 1987 г. Функционально-структурная схема информационной системы – это перечень подсистем и модулей, из которых состоит система, с описанием выполняемых ими функций и взаимодействия между ними в основных режимах работы системы. Автор использует данное понятие для определения архитектуры системы гибридного интеллекта на основе процессов².

СГИ на основе процессов создается как иерархическая многоуровневая система. В качестве основы на первом уровне используются библиотеки готовых программ, реализующие исполнительную среду для работы системы. На втором уровне СГИ представляется состояниями процессов, аккумулирующими логику работы системы, на третьем – совокупностью взаимодействующих процессов, а на четвертом – функциональными подсистемами, реализующими стратегию и тактику управления деятельностью. Основным режим работы СГИ – управление деятельностью. В этом режиме сотрудники и их автоматизированные рабочие места (АРМ) передают информацию о состоянии деятельности в процессы; процессы в зависимости от поступившей информации выдают те или иные команды в АРМ, которые при необходимости транслируют эти команды сотрудникам; сотрудники выполняют команды и передают ответы через АРМ в процессы; процессы в зависимости от ответов, поступивших от АРМ, выдают в АРМ новые команды и т. д.

Структура системы гибридного интеллекта S имеет четыре уровня:

$$S = \langle S_1, S_2, S_3, S_4 \rangle, \quad (1)$$

где S_1 – библиотеки готовых программ, реализующие исполнительную среду для работы системы; S_2 – состояния процессов, аккумулирующие логику работы системы; S_3 – совокупность взаимодействующих процессов; S_4 – функциональные подсистемы, реализующие стратегию и тактику управления деятельностью.

Обозначим библиотеки готовых программ, реализующие исполнительную среду для работы системы гибридного интеллекта, как $S_1 = \{d_{i_d}\}$, где d_{i_d} – i_d -я библиотека готовых программ; состояния процессов, аккумулирующие логику работы системы гибридного интеллекта, как $S_2 = \{\tau_{i_s, j_p}\}$, где τ_{i_s, j_p} – состояние с номером i_s процесса с номером j_p ; совокупность взаимо-

¹Бухаров, М. Н. Технология программирования для систем автоматизации экспедиционных радиофизических экспериментов : дис. ... канд. техн. наук / М. Н. Бухаров. – М., 1987. – 237 с.

²Бухаров, М. Н. Управление человеко-машинными комплексами на основе гибридного интеллекта : дис. ... д-ра техн. наук : 05.13.10 / М. Н. Бухаров. – М., 2012. – 356 с.

действующих процессов как $S_3 = \{p_{n,m}\}$, где $p_{n,m}$ – процесс с номером m функциональной подсистемы с номером n ; функциональные подсистемы как $S_4 = \{s_{i_s}\}$, где s_{i_s} – i_s -я функциональная подсистема.

В компьютере состояния процессов – это подпрограммы, а процессы – программы, которые при выполнении на компьютере вызывают данные подпрограммы. Состояния процесса выполняются последовательно. В каждый момент времени в процессе выполняется только одно состояние, которое называется текущим состоянием процесса. Все текущие состояния всех процессов выполняются одновременно. Таким образом, работа СГИ на компьютере (в локальной вычислительной сети, в глобальной сети Интернет) – это параллельное выполнение последовательных процессов.

Для наглядности представим процесс графически в виде ориентированного графа. Вершинами графа являются состояния процесса, а дугами – переходы между состояниями. Если из какого-либо состояния процесса возможны переходы в два или более состояния, то они обозначаются пунктирной линией (рис. 1).

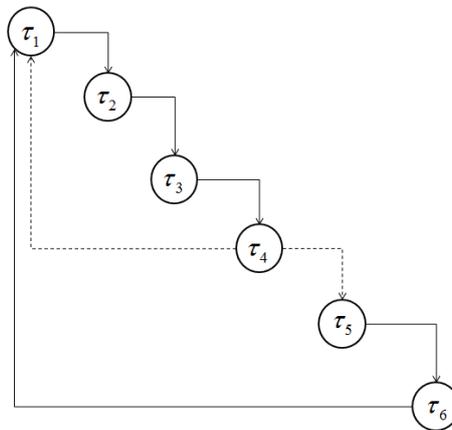


Рис. 1. Графическое представление процесса:

$\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$ – состояния процесса

Fig. 1. Process Graphic: $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$ – process states

Функциональная подсистема в общем случае имеет сложную иерархическую структуру. На k -м уровне i -я функциональная подсистема s_i^k включает подсистему управления, состоящую из нескольких процессов $\{p_{i,j_p}^k\}$; несколько функциональных подсистем $\{s_{i,j_s}^k\}$; несколько функциональных модулей $\{f_{i,j_f}^k\}$ и множество сообщений $\{m_{i,j_m}^k\}$:

$$s_i^k = \langle \{p_{i,j_p}^k\}, \{s_{i,j_s}^k\}, \{f_{i,j_f}^k\}, \{m_{i,j_m}^k\} \rangle, \quad (2)$$

где p_{i,j_p}^k – j_p -й процесс; s_{i,j_s}^k – j_s -я функциональная подсистема; f_{i,j_f}^k – j_f -й функциональный модуль; m_{i,j_m}^k – j_m -е сообщение (индексы i и k указывают на i -ю подсистему рассматриваемого k -го уровня).

Функциональный модуль может быть АРМ участника деятельности. В этом случае сам участник деятельности (сотрудник, контрагент) также является частью функциональной подсистемы (одним из ее функциональных модулей).

Подсистема управления реализует протокол взаимодействия функциональных подсистем одного уровня с помощью множества процессов. Процессы обмениваются сообщениями

с функциональными подсистемами через внутренний интерфейс. В то же время подсистема управления получает задания от внешней функциональной подсистемы и отправляет ей ответы через внешний интерфейс. Регламенты этого «вертикального» взаимодействия функциональных подсистем реализуются процессами подсистем управления смежных по вертикали уровней (верхний индекс k).

Таким образом, более детально на уровне k каждая i -я функциональная подсистема s_i^k представляется как кортеж из подсистемы управления (множество процессов $\{p_{i,j_p}^k\}$), нескольких функциональных подсистем следующего нижнего уровня $\{s_{i,j_s}^k\}$, внутреннего $\{m_{i,j_m}^k\}$ и внешнего $\{m_{i,j_m}^k\}$ интерфейсов и множества функциональных модулей $\{f_{i,j_f}^k\}$:

$$s_i^k = \langle \{p_{i,j_p}^k\}, \{s_{i,j_s}^k\}, \{m_{i,j_m}^k\}, \{m_{i,j_m}^k\}, \{f_{i,j_f}^k\} \rangle. \quad (3)$$

Внутренний интерфейс $\{m_{i,j_m}^k\}$ подсистемы s_i^k в то же время является внешним интерфейсом для функциональных подсистем следующего нижнего уровня $\{s_{i,j_s}^k\}$ и т. д.

Для облегчения создания систем гибридного интеллекта ранее автором был разработан специальный инструментальный комплекс, состоящий из нескольких компонентов [8, 9, 12].

2. Математическая модель онтологии. Онтология – это формальная спецификация концептуализации в некотором контексте предметной области [13]. Создаваемые нами онтологии и базы знаний являются множеством метаобъектов и связей между ними. Метаобъект – это некоторый текст, представляющий собой определение, понятие или какое-либо другое описание. Обозначим множество метаобъектов как $B = \{b_i\}$. Метаобъект может быть связан с одним или несколькими объектами из множества объектов A . Связь метаобъекта и объекта имеет тип. Обозначим множество всех типов связей метаобъекта и объекта как $T^{(b,a)} = \{t_i^{(b,a)}\}$. Конкретную связь метаобъекта с объектом будем называть связкой и обозначать как кортеж вида $\langle b_i, t_j^{(b,a)}, a_k \rangle$. Метаобъект имеет тип, множество типов метаобъектов обозначим $T^{(b)} = \{t_i^{(b)}\}$.

Между метаобъектами также могут быть установлены связи. Связь между двумя метаобъектами имеет тип. Обозначим множество типов всех связей между метаобъектами как $T^{(b,b)} = \{t_i^{(b,b)}\}$. Конкретную связь между двумя метаобъектами будем называть связкой и обозначать кортежем вида

$$\langle b_i, t_j^{(b,b)}, b_k \rangle. \quad (4)$$

Метаобъект может иметь одно или несколько ключевых слов. Ключевое слово – это одно слово или словосочетание в виде строки текста. Обозначим множество всех ключевых слов онтологии как $W = \{w_i\}$. Ключевые слова одного метаобъекта являются подмножеством всех ключевых слов онтологии. Множество из n ключевых слов метаобъекта b_i обозначим $W^{(b_i)} = \{w_{i_1}, w_{i_2}, w_{i_3}, \dots, w_{i_{n-1}}, w_{i_n}\}$, $W^{(b_i)} \subseteq W$. Ключевое слово может входить в метаобъект по одному из типов вхождения из множества типов вхождения: $V = \{v_i\}$. Множество V состоит из трех элементов: v_1 : ИЛИ (может быть), v_2 : И (обязательно есть), v_3 : НЕ (отсутствует).

Вхождение ключевого слова w_i в метаобъект b_k по типу вхождения v_j обозначим как кортеж

$$\langle w_i, v_j, b_k \rangle. \quad (5)$$

Поисковый запрос представляет собой множество кортежей, как в формуле (5), где на месте метаобъекта стоит символ « » (подчеркивание), показывающий, что на этом месте может быть любой метаобъект, удовлетворяющий условию вхождения в него ключевого слова w_i по типу вхождения v_j :

$$Z = \{ \langle w_{i_1}, v_{j_1}, - \rangle, \langle w_{i_2}, v_{j_2}, - \rangle, \langle w_{i_3}, v_{j_3}, - \rangle, \dots, \langle w_{i_n}, v_{j_n}, - \rangle \}. \quad (6)$$

Результат поиска – это множество метаобъектов, удовлетворяющих поисковому запросу из формулы (6). Обозначим результат поиска как множество $X = Z(B)$. Результат поиска X является не собственным подмножеством множества всех метаобъектов B : $X \subseteq B$. Поисковый запрос из формулы (6) может осуществляться последовательно или параллельно-последовательно.

При параллельно-последовательном выполнении поискового запроса выделяются следующие последовательно выполняемые группы параллельно выполняемых задач (поисковых подзапросов): поиск метаобъектов (множество X_1), удовлетворяющих условию вхождения ключевых слов по типу вхождения v_1 ; поиск метаобъектов из множества X_1 (множество X_2), удовлетворяющих условию вхождения ключевых слов по типу вхождения v_2 , и поиск метаобъектов из множества X_2 (множество X), удовлетворяющих условию вхождения ключевых слов по типу вхождения v_3 . Параллельно выполняемые задачи в этих группах бывают двух типов: задачи, выделяемые в отдельный поисковый подзапрос по принципу одновременного поиска одного или нескольких ключевых слов, и задачи, выделяемые в отдельный поисковый запрос по принципу поисковой базы (подмножество $B_i \subset B$ или все множество B).

Для облегчения онтологического моделирования предметной области, в частности для автоматического установления отдельных связей между метаобъектами, используем набор аксиом (законов и правил, которые описывают принципы существования отобранных концептов). Аксиома – это два утверждения о метаобъектах, связанные знаком следования (вывода). Утверждение о метаобъектах представляет собой предикат $P = f(b_1, b_2, \dots, b_n)$. В частности, предикат P может быть связкой, как в формуле (4), но на месте одного или обоих метаобъектов стоит символ « » (подчеркивание), показывающий, что на этом месте может быть любой метаобъект.

В разработанной автором информационной системе для онтологического моделирования предметных областей имеются три вида аксиом:

$$\langle b_i, t_j^{(b,b)}, - \rangle \Rightarrow \langle b_i, t_k^{(b,b)}, - \rangle, \quad (7)$$

$$\langle -, t_i^{(b,b)}, b_j \rangle \Rightarrow \langle -, t_k^{(b,b)}, b_j \rangle, \quad (8)$$

$$\langle -, t_i^{(b,b)}, - \rangle \Rightarrow \langle -, t_j^{(b,b)}, - \rangle. \quad (9)$$

Аксиома (7) означает, что если между метаобъектом b_i и некоторым метаобъектом имеется связь типа $t_j^{(b,b)}$, то между этими двумя метаобъектами также должна быть связь типа $t_k^{(b,b)}$. Аксиома (8) означает, что если между некоторым метаобъектом и метаобъектом b_j имеется связь типа $t_i^{(b,b)}$, то между этими двумя метаобъектами также должна быть связь типа $t_k^{(b,b)}$. Аксиома (9) означает, что если между двумя метаобъектами имеется связь типа $t_i^{(b,b)}$, то между этими двумя метаобъектами также должна быть связь типа $t_j^{(b,b)}$.

В информационной системе аксиомы задаются вручную в соответствующих диалогах. Затем можно запустить применение аксиом в автоматическом или полуавтоматическом режимах.

В автоматическом режиме осуществляется сканирование всех метаобъектов и проверка для каждого метаобъекта выполнения утверждения в левой части аксиомы. Если утверждение истинно, то проверяется наличие связи из утверждения, находящегося в правой части аксиомы.

Если связь имеется, то процесс применения аксиом продолжает проверку далее. Если же указанная в правой части аксиомы связь между метаобъектами не обнаружена, то такая связь устанавливается. И так делается для всех метаобъектов онтологии. Затем процесс применения аксиом продолжает проверку для следующей аксиомы. Она выполняется системой последовательно: сначала для одной аксиомы, затем для другой, пока не закончатся все аксиомы. Процесс применения одной аксиомы заканчивается тогда, когда будут проверены утверждения из левой части аксиомы для всех метаобъектов онтологии.

Необходимо отметить, что если в результате использования аксиомы множество всех связей метаобъектов онтологии было изменено, то в общем случае необходимо еще раз применить все аксиомы. Поэтому поочередное применение аксиом выполняется до тех пор, пока за очередной цикл применения не будет достигнуто отсутствие изменения множества всех связей метаобъектов онтологии. В случае добавления в онтологию нового метаобъекта и установления связи этого метаобъекта с одним из уже существующих в онтологии метаобъектов (создания новой связи из двух метаобъектов) при включенном режиме автоматического применения аксиом для этой новой связи метаобъектов поочередно применяются все аксиомы.

В полуавтоматическом режиме выполняется запрос пользователю на выполнение очередного цикла использования всех аксиом и каждой аксиомы в отдельности. Также в полуавтоматическом режиме информационная система перед каждым созданием новой связи между метаобъектами делает запрос пользователю и создает связь между метаобъектами только при положительном ответе.

3. Автоматизация онтологического моделирования предметной области с помощью гибридного интеллекта. Онтологическое моделирование предметной области включает спецификацию, концептуализацию и формализацию. Результатом онтологического моделирования является онтология предметной области. Для превращения онтологии в базу знаний необходимо провести ее актуализацию.

На стадии спецификации строится глоссарий терминов, включающий все важные для предметной области термины и их описания. На стадии концептуализации выявляются важные объекты предметной области. На этой стадии также определяются иерархия и связи между объектами предметной области. На стадии формализации создаются метаобъекты и связи между метаобъектами, которые соответствуют объектам и связям между объектами предметной области. В результате будет получена онтология предметной области.

На стадии актуализации определяются параметры объектов и их домены (домен параметра – это область допустимых значений параметра), а также значения, классы, подклассы и экземпляры классов. Параметры, домены параметров, значения параметров, классы, подклассы и экземпляры классов реализуются в виде метаобъектов соответствующих типов. После проведения работ на стадии актуализации онтология превращается в базу знаний. Однако «жесткой» границы между онтологией и базой знаний нет.

Для автоматизации онтологического моделирования предметных областей была разработана СГИ (см. формулы (1)–(3)), получившая название «Онтологическое моделирование предметной области» (СГИ ОМПО). В состав СГИ ОМПО вошли три функциональные подсистемы (см. формулу (2)): «Администрирование», «Экспертиза» и «Работа с пользователями».

В соответствии с формулой (3) каждая функциональная подсистема содержит процессы для взаимодействия с внешней средой и процессы для управления компонентами подсистемы. Взаимодействие с внешней средой осуществляется с помощью внешнего интерфейса, а управление внутренними компонентами подсистемы – с помощью внутреннего интерфейса. В соответствии с формулой (1) процессы содержат состояния. В статье для краткости изложения деление процессов на состояния опущено.

1. Функциональная подсистема «Администрирование» содержит следующие процессы:

- Получение информации из внешней среды;
- Передача информации во внешнюю среду;
- Подключение нового пользователя;
- Подключение нового эксперта;
- Аутентификация пользователя;

Аутентификация эксперта;

Размещение новой версии онтологии на Сервере;

Создание резервной копии онтологии;

Восстановление онтологии по резервной копии при сбоях в работе Сервера.

Функциональная подсистема «Администрирование» включает следующие внутренние компоненты:

Администратор;

Программист;

АРМ Администратора;

АРМ Программиста;

Функциональный модуль «Сервер» для размещения онтологии и Рабочих кабинетов пользователей онтологии;

Функциональные модули, используемые Программистом при настройке Рабочих кабинетов пользователей онтологии.

2. Функциональная подсистема «Экспертиза» содержит следующие процессы:

Получение информации из внешней среды;

Передача информации во внешнюю среду;

Проверка новой информации экспертами;

Редактирование Технологом онтологии на основе поступившей и проверенной экспертами новой информации;

Анализ и обобщение замечаний, полученных от Секретаря;

Подготовка Технологом новой версии онтологии.

Функциональная подсистема «Экспертиза» включает следующие внутренние компоненты:

Технолог;

Эксперт;

АРМ Технолога;

АРМ Эксперта;

Функциональный модуль «Журнал экспертиз» для размещения заключений экспертов по поступающей новой информации для размещения в онтологии;

Функциональные модули, используемые Технологом при редактировании онтологии и при проверке новых версий онтологии.

3. Функциональная подсистема «Работа с пользователями» содержит следующие процессы:

Получение информации из внешней среды;

Передача информации во внешнюю среду;

Регистрация запросов пользователей Секретарем;

Подготовка Секретарем рейтингов по использованию отдельных фрагментов онтологии и передача этих рейтингов пользователям;

Регистрация Секретарем замечаний пользователей;

Передача Секретарем поступивших от пользователей замечаний Технологу.

Функциональной подсистемы «Работа с пользователями» включает следующие внутренние компоненты:

Пользователь;

Секретарь;

АРМ Пользователя;

АРМ Секретаря;

Функциональный модуль «Журнал запросов пользователей» для размещения запросов пользователей и результатов их выполнения;

Функциональные модули, используемые Секретарем при выполнении запросов пользователей онтологии;

Функциональный модуль «Журнал замечаний пользователей» для размещения замечаний пользователей.

Для всех трех функциональных подсистем применяется единый внешний интерфейс. Внутренний интерфейс для каждой функциональной подсистемы используется свой. Внешний ин-

терфейс и все внутренние интерфейсы реализованы как распределенные базы данных и функциональные модули для размещения в базах данных информации и чтения из баз данных поступившей информации. Внешним интерфейсом пользуются только два процесса в каждой функциональной подсистеме: Получение информации из внешней среды и Передача информации во внешнюю среду. Внутренним интерфейсом в каждой функциональной подсистеме пользуются все процессы и функциональные модули этой функциональной подсистемы (в том числе и АРМ).

АРМ Администратора, Программиста, Эксперта, Технолога и Секретаря реализованы как Рабочие кабинеты на интернет-портале ROS (URL: <http://www.ros.iicenter.ru>). Скриншот рабочего места Администратора Базы знаний по экологии на интернет-портале ROS показан на рис. 2.

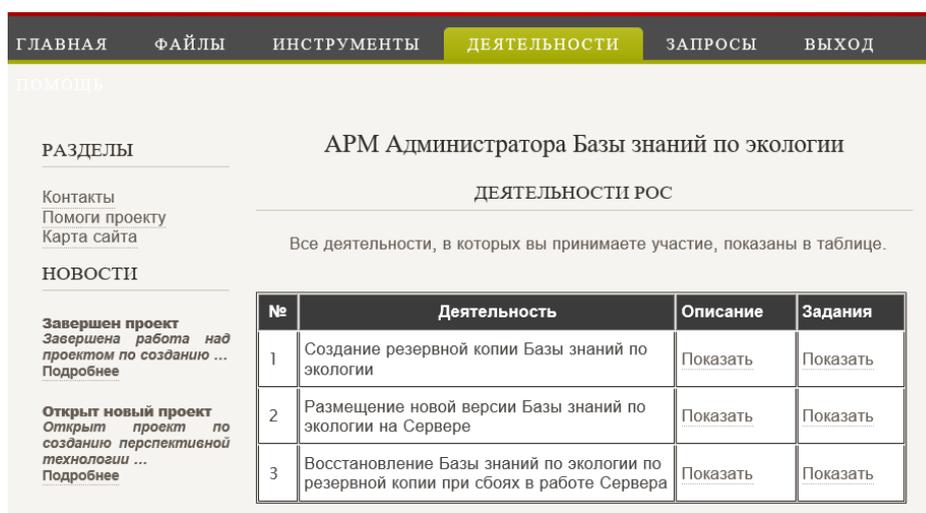


Рис. 2. Рабочее место Администратора Базы знаний по экологии на интернет-портале ROS

Fig. 2. Environmental Knowledge Base Administrator Workplace on the ROS Portal

Автоматизированные рабочие места пользователей реализованы как Рабочие кабинеты на интернет-портале WKL (URL: <http://www.wkl.iicenter.ru>). Скриншот рабочего места Пользователя Базы знаний по экологии на интернет-портале WKL изображен на рис. 3.

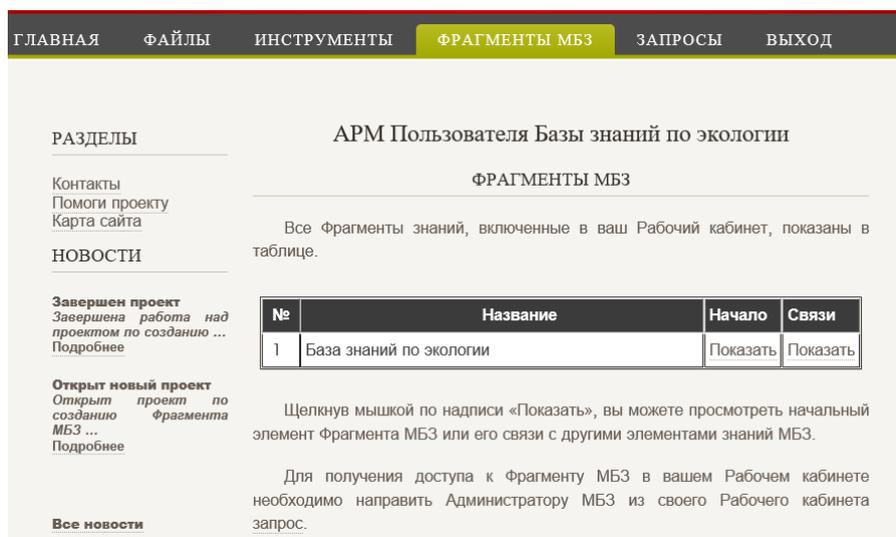


Рис. 3. Рабочее место Пользователя Базы знаний по экологии на интернет-портале WKL

Fig. 3. Environmental Knowledge Base User Workplace on WKL Internet Portal

4. Онтологическое моделирование экологии. Рассмотрим создание с помощью разработанной информационной системы СГИ ОМПО онтологическое моделирование предметной области «экология». Содержательно экология состоит из экологических объектов, понятий, теорем, гипотез, теорий, экспериментов, изобретений, открытий и других объектов, относящихся к данной предметной области. В предметной области «экология» выделены следующие разделы:

1. Основные понятия и определения.
2. Научные проблемы и задачи.
3. Научные гипотезы.
4. Научные теории.
5. Научные методы и методики исследований.
6. Математические модели.
7. Компьютерные модели.

Деятельность по сбору информации для онтологии и базы знаний по экологии, проверке и экспертизе этой информации, ее хранению и использованию автор рассматривает как систему гибридного интеллекта (книги [8, 9], статьи [10–12]). Реализация системы выполнена на основе платформы ROS и интернет-портала WKL.

Пользователями онтологии и базы знаний по экологии являются специалисты, работающие в области экологии и смежных областях, преподаватели и студенты, которым интересно все, что связано с этой наукой. Пользователи получают доступ к онтологии и базе знаний по экологии в своих Рабочих кабинетах на интернет-портале WKL.

Для иллюстрации работы СГИ ОМПО рассмотрим добавление в онтологию по экологии информации об экологических факторах.

Пусть функциональная подсистема «Работа с пользователями» СГИ ОМПО получила от одного из пользователей информацию об экологических факторах в виде следующего текста:

Экологические факторы среды оказывают на живые организмы различные воздействия, то есть могут влиять как раздражители, вызывающие приспособительные изменения физиологических и биохимических функций; как ограничители, обуславливающие невозможность существования в данных условиях; как модификаторы, вызывающие морфологические и анатомические изменения организмов; как сигналы, свидетельствующие об изменениях других факторов среды.

Один и тот же фактор среды может быть ресурсом для одних и условием для других видов. Например, свет – жизненно необходимый энергетический ресурс для растений, а для обладающих зрением животных – условие зрительной ориентации. Вода для многих организмов может быть и условием жизни, и ресурсом.

Секретарь в своем Рабочем кабинете с помощью функционального модуля «Журнал запросов пользователей» помещает данный запрос в журнал. Затем процесс «Передача информации во внешнюю среду» (функциональная подсистема «Работа с пользователями») передает его во внешний интерфейс. Процесс «Получение информации из внешней среды» (функциональная подсистема «Экспертиза») получает запрос из внешнего интерфейса и передает в АРМ Технолога. Технолог рассматривает полученную информацию, добавляет необходимую для экспертов дополнительную информацию и передает сформированную им информацию на экспертизу одновременно нескольким экспертам. При этом будут задействованы следующие процессы функциональной подсистемы «Экспертиза»:

Проверка новой информации экспертами;

Редактирование Технологом онтологии на основе поступившей и проверенной экспертами новой информации;

Подготовка Технологом новой версии онтологии.

В результате Технолог сформирует новую версию онтологии по экологии и передаст ее Администратору для размещения на Сервере. После размещения новой версии онтологии на Сервере Администратор оповестит Секретаря о выпуске новой версии онтологии. Секретарь в свою очередь известит об этом всех зарегистрированных пользователей.

Для краткости изложения задействованные здесь функциональные подсистемы, процессы, АРМ и функциональные модули не указаны.

В результате в новую версию онтологии по экологии будут добавлены следующие термины и законы:

- экологические факторы;
- абиотические экологические факторы;
- биотические экологические факторы;
- антропогенные экологические факторы;
- изменчивость во времени экологических факторов;
- закон оптимума;
- адаптация биологических организмов;
- регулярно-периодическая изменчивость во времени экологических факторов;
- нерегулярная изменчивость во времени экологических факторов;
- изменчивость экологических факторов на протяжении известных отрезков времени;
- способность переносить изменения экологических факторов;
- аккламация к экологическому фактору;
- узкая специализация в переносимости изменений экологических факторов;
- широкая выносливость в переносимости изменений экологических факторов.

Между добавленными элементами онтологии и уже имеющимися будут установлены связи следующих типов:

- включает;
- содержится в;
- является разновидностью;
- имеет разновидность;
- является характеристикой;
- имеет характеристику;
- имеет механизм;
- является механизмом;
- подчиняется закону;
- является законом для.

Установка связей между метаобъектами осуществляется вручную и автоматически с помощью аксиом (см. формулы (7)–(9)).

Фрагмент среза онтологии по экологии по типам связей «включает», «подчиняется закону», «имеет характеристику», «имеет разновидность» выглядит следующим образом:

Среда обитания – *включает* – Естественная среда обитания

Среда обитания – *включает* – Искусственная (созданная человеком) среда обитания

Естественная среда обитания – *включает* – Естественная водная среда обитания

Естественная среда обитания – *включает* – Естественная наземно-воздушная среда обитания

Естественная среда обитания – *включает* – Естественная почвенная среда обитания

Естественная среда обитания – *включает* – Непосредственно живые организмы как естественная среда обитания

Естественная среда обитания – *включает* – Абиотическая естественная среда обитания

Естественная среда обитания – *включает* – Биотическая естественная среда обитания

Абиотическая естественная среда обитания – *включает* – Естественная водная среда обитания

Абиотическая естественная среда обитания – *включает* – Естественная наземно-воздушная среда обитания

Абиотическая естественная среда обитания – *включает* – Естественная почвенная среда обитания

Биотическая естественная среда обитания – *включает* – Непосредственно живые организмы как естественная среда обитания

Экологические факторы – *включает* – Абиотические экологические факторы

Экологические факторы – *включает* – Биотические экологические факторы

Экологические факторы – *включает* – Антропогенные экологические факторы

Экологические факторы – *включает* – Ресурсы окружающей среды

Экологические факторы – *включает* – Условия окружающей среды

Экологические факторы – *подчиняется закону* – Закон оптимума

Экологические факторы – *имеет характеристику* – Изменчивость во времени экологических факторов

Экологические факторы – *имеет характеристику* – Способность переносить изменения экологических факторов

Изменчивость во времени экологических факторов – *имеет разновидность* – Регулярно-периодическая изменчивость во времени экологических факторов

Изменчивость во времени экологических факторов – *имеет разновидность* – Нерегулярная изменчивость во времени экологических факторов

Изменчивость во времени экологических факторов – *имеет разновидность* – Изменчивость экологических факторов на протяжении известных отрезков времени

Способность переносить изменения экологических факторов – *имеет разновидность* – Узкая специализация в переносимости изменений экологических факторов

Способность переносить изменения экологических факторов – *имеет разновидность* – Широкая выносливость в переносимости изменений экологических факторов.

В данном фрагменте курсивом выделены типы связей между элементами онтологии.

В СГИ ОМПО для наглядности фрагменты онтологии можно представить в графической форме (рис. 4).

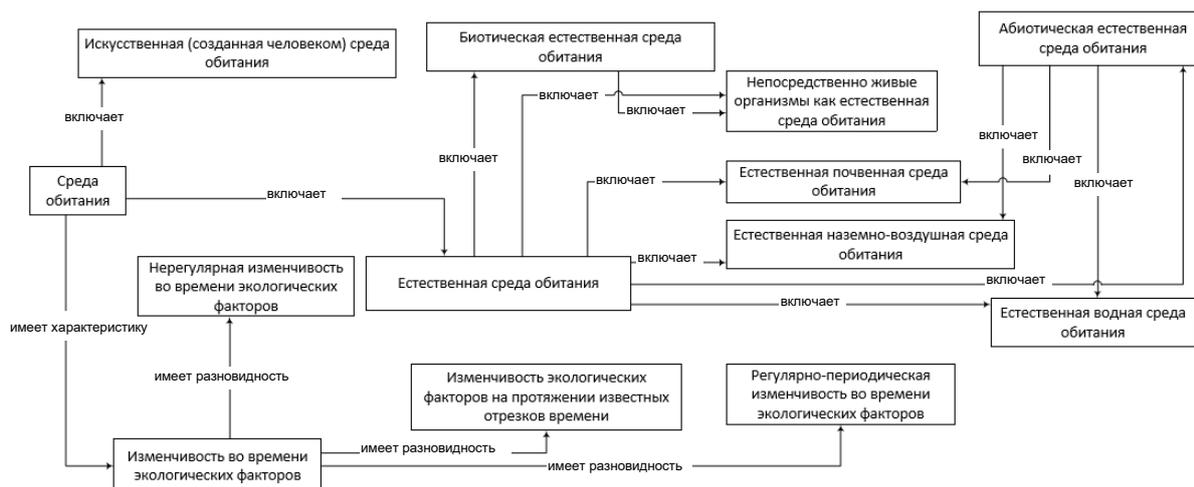


Рис. 4. Фрагмент среза онтологии по экологии по типам связей «включает», «имеет характеристику», «имеет разновидность» в графической форме

Fig. 4. A slice of ontology on ecology by types of links "includes", "has characteristic", "has variety" in graphic form

Пользователи онтологии и базы знаний по экологии – это специалисты в области экологии и смежных областях, преподаватели и студенты. Они применяют информацию из онтологии и базы знаний по экологии в своей работе и учебе.

В СГИ ОМПО (функциональная подсистема «Работа с пользователями») всем пользователям предлагается давать отзывы об использованных материалах. На основе этих отзывов определяется рейтинг каждого материала (процесс «Подготовка Секретарем рейтингов по использованию отдельных фрагментов онтологии и передача этих рейтингов пользователям»). Эти рейтинги могут применять специалисты, преподаватели и студенты при подборе материалов для своей работы и учебы.

Пользователи регистрируются на портале WKL для применения онтологии и базы знаний по экологии. После регистрации функциональная подсистема «Администрирование» выделяет каждому пользователю Рабочий кабинет и в нем открывает доступ к материалам онтологии и базы знаний по экологии, в частности к информации об экологических факторах, добавлен-

ной в очередную версию онтологии по экологии. Теперь пользователи могут получать эту новую информацию в своих Рабочих кабинетах на портале WKL и применять ее в работе и учебе.

Создание онтологии по экологии сначала выполнялось вручную, что было медленно и сопровождалось большим количеством ошибок, которые выявлялись и исправлялись также вручную. В настоящее время продолжено развитие онтологии по экологии с помощью специально разработанной для этих целей СГИ ОМПО (см. разд. 3). Используется также разработанная информационная система для онтологического моделирования других предметных областей.

Применение СГИ ОМПО позволило:

1) снизить долю рутинных работ при подборе и назначении экспертов, переписке с экспертами, архивировании информации, протоколировании всех технологических операций, автоматическом резервном копировании и восстановлении информации в случае аварийных ситуаций и др.;

2) обеспечить более четкое и оперативное выполнение таких технологических процессов, как вычитка текстов, редактирование текстов, поиск дублирующей информации, устранение дублирующей информации и др.

Неформальный анализ практического использования системы показал, что онтологическое моделирование с помощью СГИ ОМПО снижает трудоемкость и ускоряет сроки создания онтологий и баз знаний.

Проведем сравнительный анализ разработанной информационной системы с применяемыми на сегодняшний день системами для создания онтологий:

Protégé (URL: <http://protege.stanford.edu>);

Ontolingua (URL: <http://www.ksl.stanford.edu/software/ontolingua/>);

OntoStudio (панель OntoEdit) (URL: <https://www.semafora-systems.com/ontobroker-and-ontostudio-x>);

Fluent Editor (URL: <http://cognitum.eu/semantics/fluenteditor/>).

Сравнение осуществим по следующим параметрам: сложность использования, хранение информации, расширение, наличие аксиом и правил, автоматизация пополнения (таблица).

Сравнительный анализ систем для создания онтологий

Comparative analysis of systems for creating ontologies

Параметр <i>Parameter</i>	Система <i>System</i>				
	Protégé	Ontolingua	OntoStudio	Fluent Editor	СГИ ОМПО <i>SGI OMPO</i>
Сложность использования	Средняя	Высокая	Средняя	Низкая	Низкая
Хранение информации	Файлы	Серверная база данных	Файлы	Файлы	Серверная база данных
Расширение	Возможно	Невозможно	Возможно	Невозможно	Возможно
Наличие аксиом и правил	Имеется	Имеется	Имеется	Имеется	Имеется
Автоматизация пополнения	Отсутствует	Отсутствует	Отсутствует	Отсутствует	Имеется

Анализ показал, что разработанная информационная система по большинству параметров не уступает рассмотренным системам и в то же время является более простой при использовании и расширении.

Заключение. В работе представлена математическая модель онтологии и базы знаний. На основе этой модели и теории систем гибридного интеллекта (книги [8, 9], статьи [10–12]) автором реализована система гибридного интеллекта для онтологического моделирования

предметных областей. Кратко описаны основные функциональные подсистемы этой системы. Рассмотрено практическое применение разработанной информационной системы на примере онтологического моделирования экологии и проведен его неформальный анализ. Также выполнен сравнительный анализ представленной информационной системы с применяемыми на сегодняшний день системами для создания онтологий.

На основе проведенного сравнительного анализа и неформального анализа практического применения сделан вывод о том, что онтологическое моделирование с помощью разработанной информационной системы снижает трудоемкость и ускоряет сроки создания онтологий и баз знаний.

Список использованных источников

1. Грегер, С. Э. Построение онтологии архитектуры информационной системы / С. Э. Грегер, С. В. Поршневы // *Фундаментальные исследования*. – 2013. – № 10. – С. 2405–2409.
2. Kumar, A. The Ontology of Blood Pressure: A Case Study in Creating Ontological Partitions in Biomedicine / A. Kumar, B. Smith. – 2015. – Режим доступа: <http://ontology.buffalo.edu/medo/BPO.pdf>. – Дата доступа: 14.01.2021.
3. Пальчунов, Д. Е. Применение теоретико-модельных методов и онтологического моделирования для автоматизации диагностирования заболеваний / Д. Е. Пальчунов, Г. Э. Яхьяева, О. В. Ясинская // *Вестник НГУ. Сер. «Информационные технологии»*. – 2015. – Т. 13(3). – С. 42–51.
4. Онтология медицинской диагностики для интеллектуальных систем поддержки принятия решений / В. В. Грибова [и др.] // *Онтология проектирования*. – 2018. – Т. 8, № 1(27). – С. 58–73.
5. Ландэ, Д. В. Подход к созданию терминологических онтологий / Д. В. Ландэ, А. А. Снарский // *Онтология проектирования*. – 2014. – № 2(12). – С. 83–91.
6. Sowa, J. Building, Sharing and Merging Ontologies / J. Sowa. – 2015. – Режим доступа: <http://www.jfsowa.com/ontology/ontoshar.htm>. – Дата доступа: 14.01.2021.
7. Самойлов, Д. Е. Анализ неполных данных в задачах построения формальных онтологий / Д. Е. Самойлов, В. А. Семенова, С. В. Смирнов // *Онтология проектирования*. – 2016. – Т. 6, № 3(21). – С. 317–339. <http://doi.org/10.18287/2223-9537-2016-6-3-317-339>
8. Бухаров, М. Н. Системы гибридного интеллекта / М. Н. Бухаров. – М. : Научтехлитиздат, 2005. – 352 с.
9. Бухаров, М. Н. Теория систем гибридного интеллекта. Проектирование, стандартизация, моделирование и оптимизация : монография / М. Н. Бухаров. – М. : ГОУ ВПО МГУЛ, 2008. – 214 с.
10. Бухаров, М. Н. Управление сложными системами на основе гибридного интеллекта / М. Н. Бухаров // *Спецтехника и связь*. – 2015. – № 3. – С. 43–55.
11. Бухаров, М. Н. Управление сложными научно-техническими системами на основе гибридного интеллекта / М. Н. Бухаров // *Информационно-технологический вестник*. – 2015. – № 4. – С. 72–98.
12. Бухаров, М. Н. Инструментальные средства для создания систем гибридного интеллекта / М. Н. Бухаров // *Вестник РосНОУ. Сер. «Сложные системы: модели, анализ и управление»*. – 2018. – № 1. – С. 98–105.
13. Gruber, T. R. A Translation Approach to Portable Ontology Specification / T. R. Gruber // *Knowledge Acquisition*. – 1993. – Vol. 5, iss. 2. – P. 199–220.

References

1. Greger S. E., Porshnev S. V. *Construction of ontology of information system architecture*. *Fundamental'nye issledovaniya [Fundamental Research]*, 2013, no. 10, pp. 2405–2409 (In Russ.).
2. Kumar A., Smith B. *The Ontology of Blood Pressure: A Case Study in Creating Ontological Partitions in Biomedicine*, 2015. Available at: <http://ontology.buffalo.edu/medo/BPO.pdf> (accessed 14.01.2021).
3. Palchunov D. E., Yakhyaeva G. E., Yasinskaya O. V. *Application of theoretical and model methods and ontological modeling for automation of disease diagnosis*. *Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya "Informacionnye tehnologii" [Bulletin of the Novosibirsk State University. Series "Information Technology"]*, 2015, vol. 13(3), pp. 42–51 (In Russ.).
4. Gribova V. V., Petryaeva M. V., Okun D. B., Shalfeeva E. A. *Ontology of medical diagnostics for intelligent decision support systems*. *Ontologija proektirovanija [Design Ontology]*, 2018, vol. 8, no. 1(27), pp. 58–73 (In Russ.).

5. Lande D. V., Snarsky A. A. *Approach to the creation of terminological ontologies*. Ontologija proektirovanija [*Design Ontology*], 2014, no. 2(12), pp. 83–91 (In Russ.).
6. Sowa J. *Building, Sharing and Merging Ontologies*, 2015. Available at: <http://www.jfsowa.com/ontology/ontoshar.htm> (accessed 14.01.2021).
7. Samoilov D. E., Semenova V. A., Smirnov S. V. *Analysis of incomplete data in the problems of building formal ontologies*. Ontologija proektirovanija [*Design Ontology*], 2016, vol. 6, no. 3(21), pp. 317–339. (In Russ.). <http://doi.org/10.18287/2223-9537-2016-6-3-317-339>
8. Bukharov M. N. *Sistemy gibridnogo intellekta. Hybrid Intelligence Systems*. Moscow, Nauchtehlitizdat, 2005, 352 p. (In Russ.).
9. Bukharov M. N. *Teorija sistem gibridnogo intellekta. Proektirovanie, standartizacija, modelirovanie i optimizacija. Theory of Hybrid Intelligence Systems. Design, Standardization, Modeling and Optimization*. Moscow, Gosudarstvennoe obrazovatel'noe uchrezhdenie vysshego professional'nogo obrazovanija "Moskovskij gosudarstvennyj universitet lesa", 2008, 214 p. (In Russ.).
10. Bukharov M. N. *Management of Complex Systems Based on Hybrid Intelligence*. *Spectehnika i svjaz' [Special Machinery and Communications]*, 2015, no. 3, pp. 43–55 (In Russ.).
11. Bukharov M. N. *Management of Complex Scientific and Technical Systems Based on Hybrid Intelligence*. *Informacionno-tehnologicheskij vestnik [Information and Technological Bulletin]*, 2015, no. 4, pp. 72–98 (In Russ.).
12. Bukharov M. N. *Tools for Creating Hybrid Intelligence Systems*. *Vestnik Rossijskogo novogo universiteta. Serija "Slozhnye sistemy: modeli, analiz i upravlenie" [Bulletin of the Russian New University. Series "Complex Systems: Models, Analysis and Control"]*, 2018, no. 1, pp. 98–105 (In Russ.).
13. Gruber T. R. *A Translation Approach to Portable Ontology Specification*. *Knowledge Acquisition*, 1993, vol. 5, iss. 2, pp. 199–220.

Информация об авторе

Бухаров Михаил Николаевич, кандидат технических наук, старший научный сотрудник, доцент, федеральное государственное бюджетное учреждение науки «Институт радиотехники и электроники им. В. А. Котельникова Российской академии наук», Москва, Россия.
ORCID ID: <http://orcid.org/0000-0002-1420-179X>
E-mail: oberonco@mail.ru

Information about the author

Mikhail N. Bukharov, Ph. D. (Phys.-Math.), Senior Researcher, Associate Professor, Federal State Budgetary Institution of Science "Kotelnikov Institute of Radioengineering and Electronics of Russian Academy of Sciences", Moscow, Russia.
ORCID ID: <http://orcid.org/0000-0002-1420-179X>
E-mail: oberonco@mail.ru



УДК 004.75:004.42+004.94
<https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

Оригинальная статья
Original Paper

О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов

А. А. Андрушевич¹✉, И. С. Войтешенко², О. Ю. Емельянова²

¹Научно-инженерное республиканское унитарное предприятие «Межотраслевой научно-практический центр систем идентификации и электронных деловых операций», ул. Академическая, 15, к. 2, Минск, 220072, Беларусь

✉E-mail: andrushevich@bsu.by

²Белорусский государственный университет, пр. Независимости, 4, Минск, 220030, Беларусь

Аннотация

Цели. Целью проведенной аналитической и исследовательской работы являлись разработка и первоначальная оценка возможностей новой среды имитационного моделирования компонентов и приложений в Интернете вещей. Актуальность задачи связана с необходимостью упрощения исследования и тестирования таких приложений в связи с активным развитием и расширением Интернета вещей. При разработке среды имитационного моделирования преследовались следующие основные цели: построение математической модели, реализация программного обеспечения, в рамках которого могут быть проведены эксперименты с данной моделью, обеспечение возможности анализировать результаты и корректировать модель.

Методы. Использовались методы имитационного моделирования.

Результаты. Анализ актуальности и востребованности результатов исследований показал, что в качестве примера для демонстрации методов и средств решения задачи имитационного моделирования компонентов и приложений в Интернете вещей может быть выбрана реализация части модели приложения «умный дом», управляющей энергоэффективностью жилых помещений. Была разработана математическая модель дома, а также среда для моделирования на основе инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core.

Заключение. Разработана имитационная модель системы управления энергопотреблением «умного дома», включающая моделирование времени, условий внешней среды, теплопотерь, режимов работы отопительного оборудования и поведения жителей такого дома. На основе реализованной модели проведена первоначальная серия имитационных экспериментов, направленных на проверку базовой функциональности подготовленной среды имитационного моделирования и выбранного примера моделирования системы управления энергопотреблением жилых помещений.

Ключевые слова: Интернет вещей, имитационное моделирование, визуальное программирование, облачные вычисления, облачные технологии, управление энергопотреблением «умного дома»

Для цитирования. Андрушевич, А. А. О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов / А. А. Андрушевич, И. С. Войтешенко, О. Ю. Емельянова // Информатика. – 2022. – Т. 19, № 2. – С. 100–116. <https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.03.2022

Подписана в печать | Accepted 15.04.2022

Опубликована | Published 29.06.2022

About simulation environment for modeling the Internet of Things components and applications using visual flow-based programming tools and cloud services

Alexey A. Andrushevich^{1✉}, Iosif S. Vojteshenko², Olga Ju. Emelianova²

¹*Scientific Engineering Republican Unitary Enterprise "Interbranches Research & Development Centre for Identification Systems and e-Business Operations", st. Akademicheskaya, 15, building 2, Minsk, 220072, Belarus*

✉E-mail: andrushevich@bsu.by

²*Belarusian State University, av. Nezavisimosti, 4, Minsk, 220030, Belarus*

Abstract

Objectives. The purpose of the analytical and research work was to develop and perform an initial assessment of the capabilities of simulation environment for modeling the Internet of Things (IoT) components and applications. The relevance of the problem is associated with the need to simplify research and testing of such systems as the field is growing. In the implementation of the simulation environment, the following goals were pursued: building a mathematical model; implementation of software, capable of running experiments on that model; providing the user with the ability to analyze results and adjust the model.

Methods. Methods of simulation modeling were used.

Results. Analysis of the relevance and impact of the research results has led to an appropriate example for demonstrating methods and means of solving the problem of IoT subsystems, components, and applications simulation in the proposed environment. This example has been implemented in the part of the Smart Home application model responsible for the energy efficiency optimization in residential buildings enclosed in a simulation environment based on an integrated software package consisting of the Node-RED visual tool for flow-based programming and the Yandex Cloud / Yandex IoT Core cloud service.

Conclusion. A simulation model for managing energy consumption of a "smart home" was developed and implemented using the previously specified software package, including modeling time, environmental conditions, heat loss, operating modes of heating equipment and the behavior of house residents. Based on the implemented model, an initial series of simulation experiments were also carried out, on the one hand, aimed at checking some characteristics of the functionality of the developed simulation environment and the selected example for simulation of residential premises energy management. As a result of the initial experiments, the basic functionality of the integrated software package was proven and demonstrated based on the use of Node-RED and the Yandex Cloud / Yandex IoT Core cloud service for solving problems of simulation modeling of components, subsystems and applications of the Internet of things.

Keywords: Internet of Things, simulation, visual programming, cloud computing, cloud technologies, smart home energy management

For citation. Andrushevich A. A., Vojteshenko I. S., Emelianova O. Ju. *About simulation environment for modeling the Internet of Things components and applications using visual flow-based programming tools and cloud services*. *Informatika [Informatics]*, 2022, vol. 19, no. 2, pp. 100–116 (In Russ.).

<https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

Conflict of interest. The authors declare of no conflict of interest.

Введение. В настоящее время человечество переживает период активного развития Интернета вещей, выполняются теоретические работы и практическое внедрение новых технологий в различных его приложениях. В связи с этим возникают вопросы стандартизации используемых подходов и разработки общей информационно-коммуникационной среды. Решением данных вопросов занимаются национальные и международные организации по стандартизации, такие как Государственный комитет по стандартизации Республики Беларусь и W3C (URL: <https://www.w3.org/TR/wot-usecases/>). Настоящая статья посвящена подготовке среды имитационного моделирования, а также исследованию ее функциональных и нефункциональных характеристик на основе одного из инструментов, рекомендованных W3C для использования в сфере

Интернета вещей (URL: <https://www.w3.org/2020/04/pressrelease-wot-rec.html/>), а именно инструмента визуального программирования Node-RED (URL: <https://nodered.org/>) в интеграции с облачным сервисом Yandex Cloud (URL: <https://cloud.yandex.ru/docs/>).

Предложение интегрировать Node-RED с Yandex Cloud / Yandex IoT Core уже высказывалось разработчиками [1], однако адаптация и накопление опыта в результате использования этого инструментария в различных сферах является актуальной задачей.

Имитационное моделирование при построении и исследовании приложений Интернета вещей. Приложения Интернета вещей тесно связаны с окружающей средой, где устройства и его составные компоненты представляют собой мобильные гетерогенные объекты, оснащенные датчиками и манипуляторами. Каждое устройство может отличаться от других по функциональности и доступности вычислительных ресурсов, что, в свою очередь, определяется программным и аппаратным обеспечением. Для связи элементов используется коммуникативно-вычислительная инфраструктура Интернета, для которой требуется обеспечение корректности обработки данных и взаимодействия между различными частями рассматриваемой распределенной системы.

Таким образом, приложение Интернета вещей может быть представлено как надстройка над потоком данных об окружающей среде, которая принимает решения, воздействующие на эту среду (рис. 1).

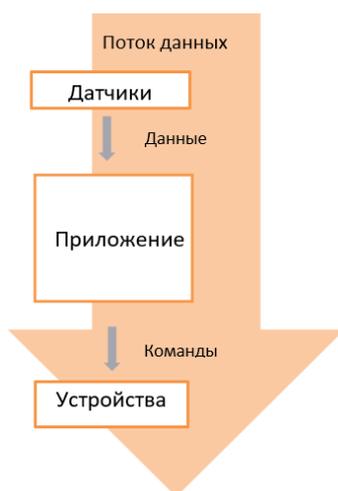


Рис. 1. Схематическое представление приложения в Интернете вещей

Fig. 1. Schematic representation of Internet of things application

Всесторонняя оценка работы программно-аппаратных продуктовых приложений в области Интернета вещей в среде их конечного развертывания требует большого количества ресурсов, в первую очередь финансовых и временных. Еще более сложным оказывается решение задачи исследования гипотез в отношении перспективных подходов. В этой связи представляет интерес разработка методов проверки правильности функционирования коммуникативно-вычислительной системы и ее соответствия заданным требованиям. При построении моделей распределенных систем традиционно используются различные языки: Unified Modeling Language (UML, URL: <https://www.uml.org/>), Very large scale Hardware Description Language (VHDL, URL: <http://www.eda-twiki.org/cgi-bin/view.cgi/P1076/WebHome/>), Triad.Net и др. При создании симуляторов компьютерных сетей часто применяется модульная компонентно-ориентированная C++ библиотека для дискретно-событийного моделирования OMNeT++ (URL: <https://omnetpp.org>) совместно с библиотекой сетевых моделей INET Framework (URL: <https://inet.omnetpp.org>). Также следует упомянуть библиотеки NS3 (URL: <https://www.nsnam.org>), GNS3 (URL: <https://www.gns3.com>) и Common Open Research Emulator (CORE, URL: <https://coreemu.github.io/core/>).

При моделировании компонентов и приложений в Интернете вещей важно иметь возможность отделять технические аспекты реализации, такие как выбор конкретных устройств или протоколов взаимодействия, чтобы сосредоточиться на обеспечении корректного поведения программной части. Иначе говоря, требуется разработать метод, который позволял бы в зависимости от конкретной ситуации проверять систему или ее прототип на соответствие заранее определенным критериям, давать оценку качества работы системы путем вычисления заданных показателей.

Имитационное моделирование – метод исследования, состоящий в создании и изучении модели, достаточно похожей на исходную систему. Распространенный пакет компьютерного имитационного моделирования динамических систем Simulink из числового пакета MATLAB (URL: <https://www.mathworks.com/products/simulink.html>) является типичным средством визуально-ориентированного программирования. Пользователь имеет возможность на основе готовых графических модулей (блоков) составлять структурные схемы (блок-схемы) исследуемой системы, выполнять имитацию происходящих в системе процессов и анализировать их течение во времени.

Несмотря на наличие визуального представления, применение Simulink требует владения определенными навыками. Кроме того, Simulink отличается относительно высокой стоимостью, в особенности для разработчиков или исследователей, не имеющих доступ в рамках некоторой организации. Поэтому вместо пакета Simulink желательно использовать программные средства с открытым исходным кодом или с оплатой только реально потребленных ресурсов. Что касается процесса разработки и тестирования, то модель, реализованная в Simulink, не может быть напрямую использована в приложениях Интернета вещей. Более того, для уже существующих приложений требуется разработка отдельной модели. Все эти факторы приводят к тому, что применение Simulink в некоторых случаях оказывается невозможным или неэффективным.

Среда имитационного моделирования должна позволять реализовывать широкий круг сценариев моделирования и эмулируемых устройств, а также рассчитывать оценки различных характеристик качества. При этом созданная система должна быть легко изменяемой и дополняемой. Кроме того, процесс разработки со стороны пользователя упрощается, когда существуют доступные компоненты для решения некоторых задач в формате библиотеки. В качестве дополнительного требования можно также выдвинуть возможность переноса разработанной системы между различными средами исполнения, в том числе возможность упрощенного перехода от прототипа к конечному продукту. Поэтому наиболее важным критерием выбора технологии является спектр ее возможностей.

Таким образом, возникает необходимость подготовки и настройки особой среды имитационного моделирования, в рамках которой могли бы проводиться имитационные эксперименты над компонентами и приложениями в Интернете вещей, т. е. сравнение различных сценариев их функционирования при разных условиях окружающей среды. При моделировании таких приложений требуется воспроизведение показателей, связанных со всеми относящимися к системе процессами, в течение длительного виртуального времени. В этой связи под моделированием понимается исследование поведения приложения косвенным путем – при помощи анализа вспомогательных объектов [2].

Наконец, отметим, что сложность может представлять разработка таких требований или критериев моделирования, которые бы учитывали несколько конфликтующих между собой целей. Зачастую задачи приложений Интернета вещей включают в себя несколько предписаний, между которыми требуется найти компромисс, причем на этапе разработки нередко само соотношение еще не определено. Таким образом, необходимо обеспечить достаточную гибкость среды имитационного моделирования, чтобы был возможен легкий переход между различными критериями и показателями качества.

Подводя итоги, можно предложить следующую схему для среды имитационного моделирования компонентов и приложений в Интернете вещей (рис. 2).

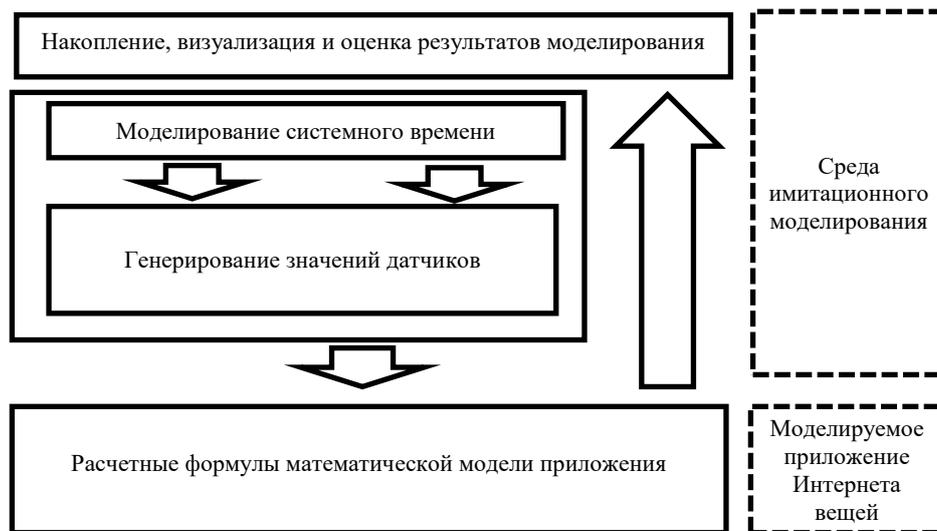


Рис. 2. Схематическое представление среды моделирования приложений в Интернете вещей

Fig. 2. Schematic representation of modeling environment for Internet of things application

Программный инструментарий (Node-RED и Yandex IoT Core) и подготовка среды моделирования. Node-RED – это визуальный инструмент разработки, основанный на парадигме потокового программирования. В Node-RED приложение представляется как сеть так называемых узлов (URL: <https://nodered.org/>), каждый из которых имеет определенную цель: на вход поступают некоторые данные, они обрабатываются внутри, после чего передаются дальше по сети. За перемещение данных между узлами отвечает сеть.

Приложение Интернета вещей представляется как отдельный поток Node-RED, но оно может быть реализовано независимо и подключено через сетевые узлы. В отдельном потоке или потоках реализуется программа для проведения имитационных экспериментов. В нее входят источники данных, генераторы различных показателей, в том числе и системного времени, а также узлы, отвечающие за коммуникацию с моделируемым приложением и анализ его функционирования.

Yandex IoT Core – основной сервис экосистемы Yandex Cloud для Интернета вещей. Он предназначен для двустороннего обмена данными между облаком и устройствами с использованием протокола MQTT. В этом протоколе для передачи данных используются именованные очереди, иначе называемые темами или топиками, в которые можно записывать данные. Подписавшись на события очереди, объекты могут асинхронно получать данные из нее. Основными элементами Yandex IoT Core являются устройство и реестр, которые обмениваются данными и командами через MQTT-брокер (URL: <https://cloud.yandex.ru/docs/>). Устройство может отправлять телеметрические данные и получать команды. Реестр – это набор логически связанных между собой устройств. Устройства в реестре могут взаимодействовать между собой, отправляя данные и получая команды. Реестр может читать данные и отправлять команды. MQTT-брокер отвечает за обмен MQTT-сообщениями между устройствами и реестрами: получает и обрабатывает сообщения, контролирует их доставку MQTT-клиентам (устройствам и реестрам). Обмен данными может происходить только через MQTT-брокер, который обеспечивает необходимый уровень качества обслуживания при обмене сообщениями. Чтобы устройства и реестры могли получать сообщения через MQTT-брокер, они должны быть подписаны на нужные топики.

Yandex IoT Core дополняет Node-RED, создавая возможность коммуникации с моделируемым приложением Интернета вещей при имитационном моделировании. Таким образом, приложения Интернета вещей состоят из двух частей: первая связана с передачей данных и команд между хранилищем и устройствами, а вторая осуществляет работу с полученными данными (рис. 3).

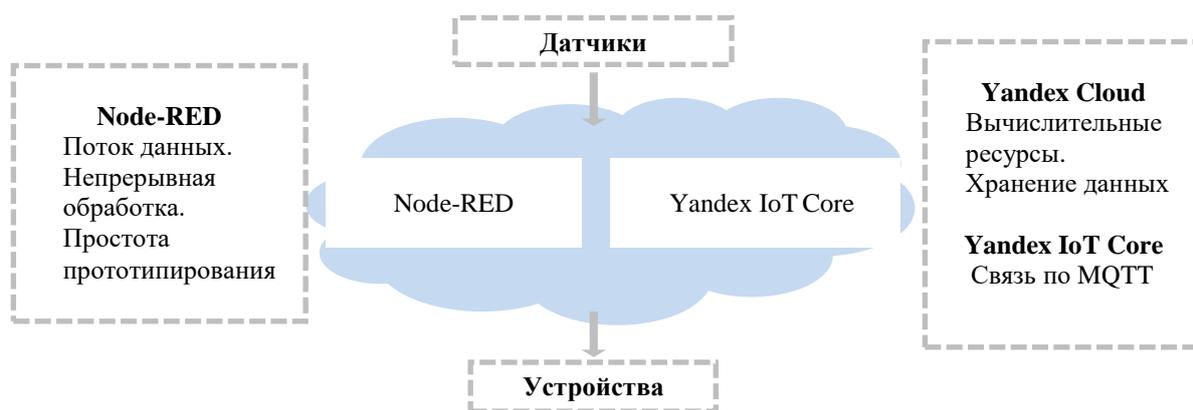


Рис. 3. Схематическое представление приложения в Интернете вещей на основе Node-RED и Yandex IoT Core
 Fig. 3. Schematic representation of Internet of things application based on Node-RED and Yandex IoT Core

Преимущества и ограничения Node-RED и Yandex IoT Core. При использовании в сфере Интернета вещей Node-RED предлагает несколько преимуществ по сравнению с другими вариантами. Во-первых, он является инструментом для визуального программирования, что позволяет облегчить создание прототипов, интегрировать все стадии разработки системы. При этом Node-RED достаточно гибок, чтобы решать большинство имеющихся в области Интернета вещей задач. Во-вторых, Node-RED можно запускать на большом количестве платформ. Использование Node-RED не накладывает ограничений на формат данных, кроме того, что он должен быть представлен как объект JavaScript. Таким образом, прототип можно использовать как основу для дальнейшей работы над продуктом. В-третьих, разработанные сценарии и правила не зависят от физической реализации системы, поскольку Node-RED предоставляет возможности для создания абстрактных устройств. При необходимости физические устройства можно позже привязать к системе через эти абстрактные устройства.

Вместе с тем использование Node-RED сопровождается дополнительными тратами на пересылку и обработку данных. Например, большую трудность представляют ограничения по частоте событий. На практике, если сигнал посылается чаще 10 раз в секунду, в приложении на основе Node-RED появляются ощутимые задержки. Кроме того, задачи некоторых приложений могут быть слишком сложными для функциональности, предлагаемой Node-RED, хотя это частично может решаться пользовательскими дополнениями.

Подготовка среды к проведению экспериментов. Использование Node-RED в сочетании с Yandex IoT Core позволяет решить все вышеописанные вопросы – от моделирования данных до выдачи результатов. Среда имитационного моделирования, включающая в себя генерацию датчиков, подключение системы, визуализацию информации и пользовательский интерфейс, реализована с помощью Node-RED. Для сбора и передачи данных был подключен Yandex IoT Core. Далее опишем ключевые детали проектирования и реализации.

Для исполнения среды Node-RED была создана виртуальная машина в облачном сервисе Yandex Compute Cloud. Этот сервис является частью Yandex Cloud и предоставляет масштабируемые вычислительные мощности для создания виртуальных машин и управления ими (URL: <https://cloud.yandex.ru/docs/>). Он предлагает широкий выбор настроек виртуальных машин – от различных операционных систем до тонкой настройки используемых ресурсов. Также был создан реестр системы в Yandex IoT Core, включающий в себя несколько устройств. Соединение с этими объектами было установлено с помощью узлов MQTT Input и MQTT Output.

Моделирование времени и значений датчиков. Немаловажным свойством приложений Интернета вещей и данных, с которыми они работают, является большая длительность обрабатываемых и управляемых процессов. К примеру, нагревание комнаты может занимать несколько часов, а значительные изменения температуры окружающей среды наблюдаются в масштабе от дня до года в зависимости от требуемой амплитуды. Под *внешним временем* будем понимать время, которое занимает моделирование с точки зрения внешнего наблюдателя. *Внутреннее,*

или *системное, время* – это время, которое проходит внутри модели. Моделируемое приложение работает только с системным временем. Разработка приложений Интернета вещей чаще всего требует моделирования внутрисистемного времени со значительным ускорением относительно внешнего времени. Таким образом, если за один шаг симуляции для внешнего наблюдателя проходит одна секунда, то для приложения может проходить одна минута, один час, одни сутки и т. д. Производительность обработки сигналов в Node-RED на практике ограничена 10 сигналами в секунду. Соответственно, при временном разрешении внутри системы в один час можно добиться ускорения по сравнению с реальным временем в 36 000 раз, а при временном разрешении в минуту – только в 600.

При реализации моделирования времени с помощью Node-RED интервал внешнего времени поддерживается с помощью узла Inject. Поскольку поток все время активен, то был добавлен узел, считающий шаги симуляции в пределах от нуля до заданного общего количества. По достижении последнего шага узел выдает сигнал об окончании эксперимента на отдельный выход. Для удобства демонстрации также использовался узел, отвечающий за получение внутрисистемного времени в часах.

Исходные данные при генерации показаний датчиков помещаются в отдельный узел для конфигурации. Неизменяемые данные задаются при развертывании потока, при этом есть возможность установить часть данных в начальное для эксперимента состояние через графический интерфейс.

Эмуляция работы датчика зависит от характера его показаний (дискретные или непрерывные). Примером дискретного датчика является датчик присутствия. В Node-RED такой датчик моделируется путем проверки условия, причем узел выдает сообщение, если условие оказывается истинным. Аналогичным образом можно действовать для дискретных датчиков, имеющих более одного значения. Примером непрерывного датчика может служить датчик температуры. Расчет температуры выполняется в нескольких узлах Node-RED, поскольку на ее значение влияют многие факторы. Некоторые из этих узлов относятся к обработке данных, полученных от моделируемого приложения. Отдельный узел был выделен для передачи данных о температуре на каждом шаге симуляции, поскольку предполагается, что приложение должно регулярно получать новые данные о температуре внутри подконтрольного ему помещения. Набор узлов, предназначенных для генерации показаний датчиков и управления временем, показан на рис. 4. Здесь и далее изображения взяты из визуального представления исходного текста программного обеспечения, созданного внутри среды Node-RED.

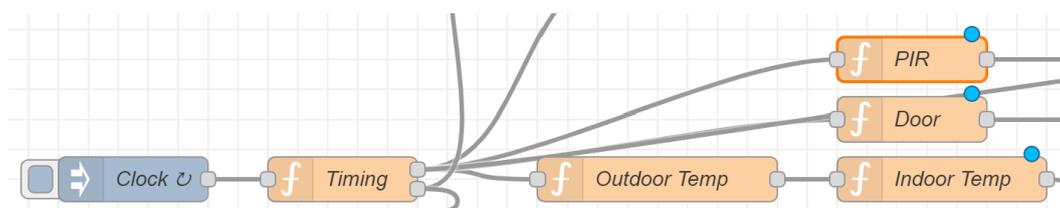


Рис. 4. Основные узлы генерации показаний датчиков

Fig. 4. Main sensor data generation nodes

Модель окружающего мира может быть в любой момент дополнена (при появлении необходимости в моделировании нового показателя) за счет добавления в систему нового узла, который соединяется с узлом, генерирующим время.

Пользовательский интерфейс. Возможности конфигурации среды пользователем. Чтобы среда имитационного моделирования была более наглядной и интуитивно понятной, желательно визуализировать процесс ее работы.

Для визуализации данных Node-RED предлагает особые графические узлы, позволяющие создавать элементы интерфейса, такие как текстовые поля, графики, на панели управления. Например, на графике можно показать изменение температуры или накопление стоимости

энергии, а с помощью текста – стоимость энергии или количество шагов моделирования. Для сбора данных использовались MQTT-узлы с подключением к Yandex IoT Core. Узлы, отвечающие за сбор и передачу данных, а также за их отображение на графике, показаны на рис. 5.

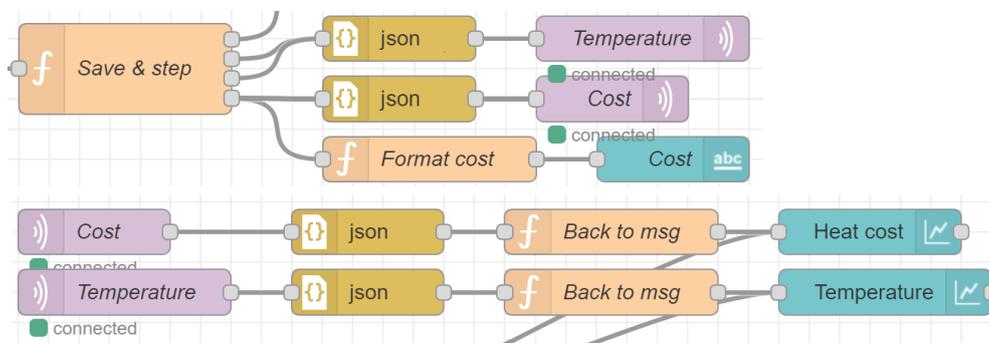


Рис. 5. Узлы для сбора, передачи и визуализации данных

Fig. 5. Nodes for data gathering, transmission and visualization

Помимо визуализации, собранные данные сохраняются в файл, который доступен через пользовательский интерфейс. Этот файл создается при запуске эксперимента и дополняется при поступлении новых данных. Тут же можно выгрузить настройки системы, чтобы учесть их при дальнейшем анализе. Узлы для создания, обновления и скачивания файлов изображены на рис. 6.

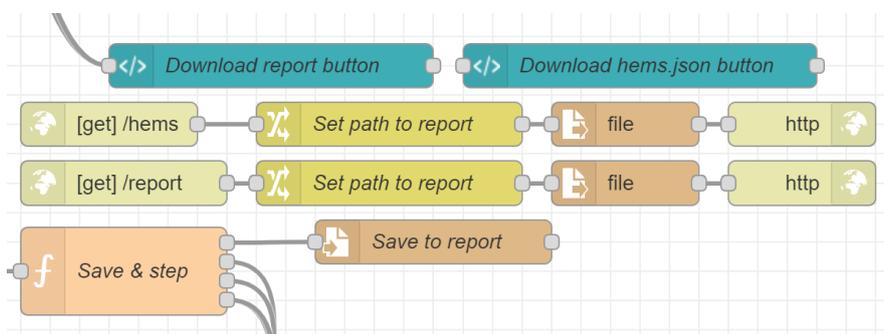


Рис. 6. Узлы для работы с файлами

Fig. 6. Nodes for file management

Таким образом, при запуске эксперимента пользователь может видеть динамическое изменение внешней температуры и температуры внутри дома, накопление общей стоимости затраченной энергии, количество пройденных шагов симуляции. По окончании эксперимента можно выгрузить сгенерированные данные для их анализа. Вкладка с элементами графического интерфейса, описанными выше, показана на рис. 7.

Пользователю также доступны возможности изменения конфигурации системы. Можно настроить две группы параметров: параметры среды моделирования и параметры моделируемого приложения. К первой группе относятся общее количество шагов и их частота, выраженная как количество шагов в одном часе по внутрисистемному времени. Вторая группа может быть разделена на параметры моделируемого приложения и настройки включаемых модулей. В частности, пользователь может отключать те или иные функции приложения, чтобы проверить реакцию приложения или сравнивать разные стратегии управления энергопотреблением. Под конфигурационную часть пользовательского интерфейса была разработана отдельная вкладка в панели управления (рис. 8).

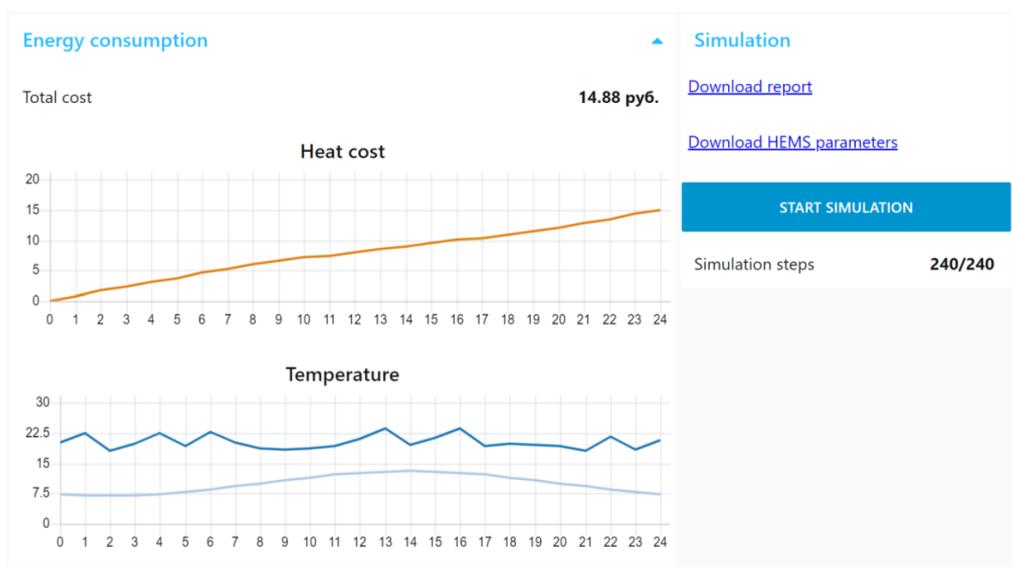


Рис. 7. Графический интерфейс пользователя для проведения симуляции

Fig. 7. Graphical user unterface for simulation execution

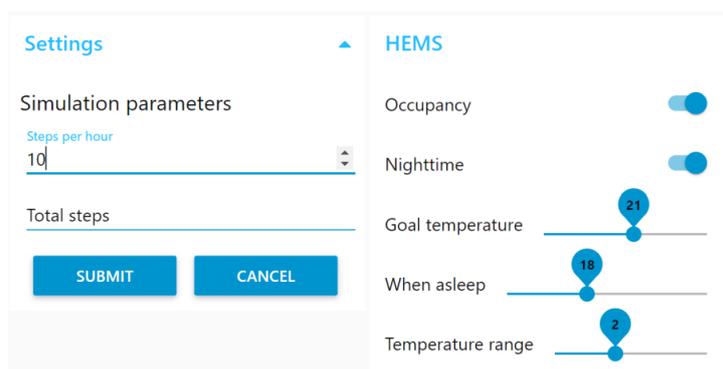


Рис. 8. Конфигурационная часть графического интерфейса пользователя

Fig. 8. Configuration part of graphical user unterface

Моделируемое приложение и математическая модель Интернета вещей. В качестве моделируемого приложения Интернета вещей будем рассматривать систему управления энергоэффективностью «умного дома» [3]. Домохозяйства вносят значительный вклад в общее энергопотребление. В структуре конечного потребления топливно-энергетических ресурсов в Республике Беларусь в 2019 г. жилищный сектор занимал почти 27 % [4]. Традиционно большинство решений в сфере управления энергоэффективностью являлись «статическими», т. е. воздействовать на их поведение после установки практически невозможно. В противоположность им в настоящее время набирают популярность интерактивные подходы, к которым относятся технологии, требующие настройки и взаимодействия с человеком после установки [5, 6].

Прежде чем приступить к планированию и проведению имитационного эксперимента, необходимо сформулировать тестируемую и базовую гипотезы, например, в виде сценария. От выбора базовой конфигурации системы будут зависеть рассчитанные преимущества и недостатки, поэтому важно определить ее корректно. В дальнейшем следует определить конфигурацию эксперимента, в частности специфицировать входные данные об окружающей среде в зависимости от требуемых для расчета показателей, выбрать длительность проведения эксперимента и интервал между шагами моделирования. В итоге необходимо подключить модель прототипа, провести имитационный эксперимент и проанализировать его результаты.

Моделирование условий внешней среды. В контексте обеспечения эффективного использования энергии для обогрева помещения наиболее важным показателем внешней среды является ее температура. Заметим, что этот показатель является внутренним по отношению к среде имитационного моделирования. На работу моделируемого приложения он оказывает влияние через расчет внутренней температуры контролируемого помещения, увеличивая или уменьшая потери тепла.

В случае коротких промежутков времени, применяемых для проверки корректности работы и оценки качества отдельных компонентов, простым способом моделирования является использование суммы средней температуры и суточного колебания:

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega t + \phi), \quad (1)$$

где $T_{\text{внеш}}$ – среднесуточная температура, A – амплитуда колебаний температуры, $\omega = 2\pi/24$ – частота колебаний, ϕ – фаза. Более интуитивно понятным является представление

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega(t - h_0)), \quad (2)$$

где h_0 – время суток, в которое температура должна быть максимальной.

На более длительных промежутках времени начинают играть роль годовичные колебания температуры, а также фактор случайности. Случайностью могут определяться как характер и амплитуда колебаний температуры в течение дня, так и ее среднее значение.

Моделирование тепловых потерь и работы устройств отопления, вентиляции и кондиционирования. Для расчета затрат на отопление были определены значения параметров, используемых в имитационном моделировании, в том числе параметры дома: $l_{\text{дома}} = 30$ м, $w_{\text{дома}} = 10$ м, $h_{\text{дома}} = 4$ м, $\alpha_{\text{крыши}} = 30^\circ$, $N_{\text{окон}} = 4$ ед., $h_{\text{окна}} = 1$ м, $w_{\text{окна}} = 1$ м, $k_{\text{стены}} = 0,038$ Вт/м·К, $L_{\text{стены}} = 0,2$ м, $k_{\text{окна}} = 0,78$ Вт/м·К, $L_{\text{окна}} = 0,1$ м, $\rho_{\text{воздуха}} = 1,225$ кг/м³ – и параметры нагревателя: $L = 1$ кг/с, $T_{\text{нагр}} = 50$ °С, $c = 1005,4$ Дж/кг·К, $p = 0,1616$ у.е./кВт·ч.

На основании значений указанных параметров при некоторых упрощениях были рассчитаны тепловые потери дома. Для этого использовались формулы для площади стен, площади окон, теплового сопротивления стен, теплового сопротивления окон, массы воздуха в доме, скорости теплового потока от обогревателя и скорости тепловых потерь. На основании этих величин определяется температура воздуха внутри помещения на следующем шаге симуляции:

$$T_{\text{внутр}}(t + \Delta t) = T_{\text{внутр}}(t) + cM_{\text{возд}}(Q_{\text{нагр}} - Q_{\text{потерь})\Delta t, \quad (3)$$

где t – внутрисистемное время симуляции, Δt – отрезок времени между шагами симуляции, c – удельная теплоемкость воздуха, $Q_{\text{нагр}}$ – скорость нагревания от обогревателя, $Q_{\text{потерь}}$ – скорость тепловых потерь, $M_{\text{возд}}$ – масса воздуха в помещении. Узлы расчета внутренней температуры представлены на рис. 9.

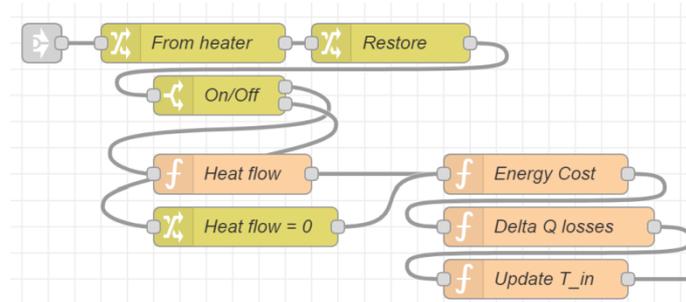


Рис. 9. Узлы расчета внутренней температуры помещения

Fig. 9. Nodes for internal room temperature calculation

Моделирование работы устройств отопления, вентиляции и кондиционирования и поведения жителей дома. Основным датчиком для работы с устройствами отопления, вентиляции и кондиционирования (ОВиК) в контексте управления энергопотреблением является датчик температуры внутри контролируемого помещения. В качестве базового сценария управления энергопотреблением была разработана система, включающая и выключающая отопление в зависимости от температуры внутри подконтрольного ей помещения. В этом сценарии отправляется команда на включение, когда разница между целевой и реальной температурой достигает заданного порога, и на выключение, когда температура вновь находится в нужных пределах. Возможные узлы для обработки температурных показаний изображены на рис. 10.

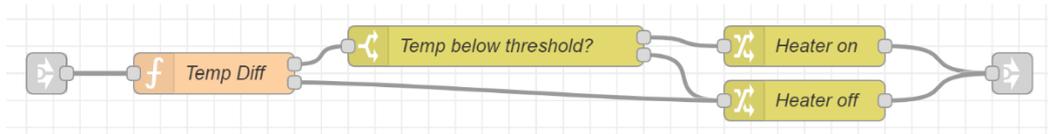


Рис. 10. Узлы для обработки показаний датчика температуры

Fig. 10. Nodes for temperature sensor data processing

Получаемый температурный режим в течение одних суток показан на рис. 11, где верхняя линия отражает температуру в помещении, нижняя – температуру снаружи, а отсутствием заливки отмечены желаемые пределы изменения температуры внутри помещения.

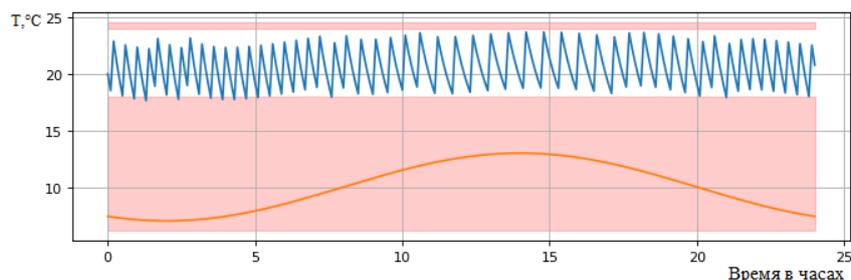


Рис. 11. Температурный режим базового сценария

Fig. 11. Temperature conditions of the base scenario

В течение дня могут возникать промежутки времени, когда дом пустует и нет определенных требований по температуре, что позволяет уменьшить потребление энергии. Такая ситуация фиксируется с помощью датчиков присутствия. Например, система может поддерживать несколько состояний в виде конечного автомата, переходы между которыми осуществляются по сообщениям от датчиков. Имеет смысл выделить три возможных состояния, связанных с поведением людей: «дом пуст» относится к случаю, когда внутри нет ни одного человека; «жители активны», когда хотя бы один человек находится в доме и бодрствует; «жители спят», когда нет ни одного бодрствующего человека в доме. Узлы системы, предназначенные для реагирования на сообщения от таких датчиков, изображены на рис. 12.

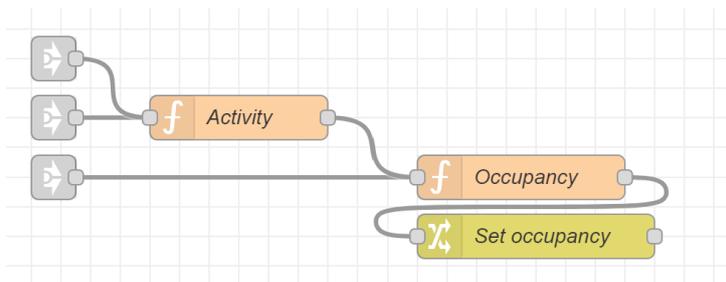


Рис. 12. Узлы обработки данных от датчиков движения

Fig. 12. Nodes for movement sensor data processing

Для сравнения с базовым сценарием были выбраны две стратегии уменьшения энергопотребления, основывающиеся на отслеживании поведения пользователей. Первая из них состоит в том, чтобы наблюдать за активностью людей, находящихся в доме, с помощью датчиков движения. Применение такой стратегии оправдано, поскольку воздействие холода не влияет на качество сна, в то время как излишне высокая температура его снижает [7]. Вторая стратегия использует датчик на входной двери, чтобы отслеживать моменты, когда люди покидают помещение или входят в него. В этом случае можно отключать устройства ОВиК, когда они не требуются.

Стратегия, основанная на активности, работает следующим образом. К системе подключается датчик движения. Пока он посылает сигналы, система находится в активном состоянии и поддерживает один заданный температурный уровень. После каждого сигнала система ждет условленное время, которое сбрасывает по приходу нового сигнала. Вместе с тем если за это условленное время не появилось новых сигналов, то система переходит в неактивное состояние, в котором поддерживает меньшую температуру (рис. 13).

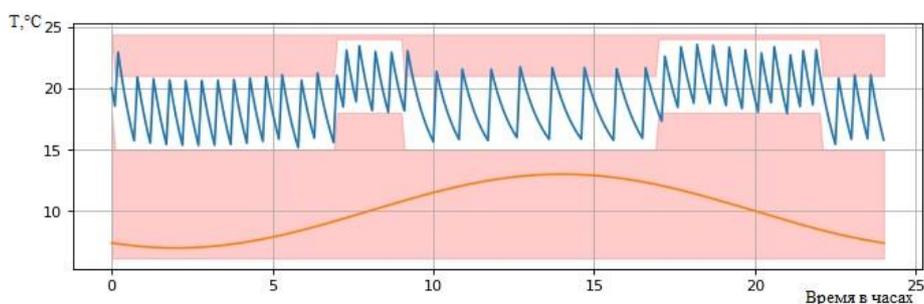


Рис. 13. Температурный режим с учетом активности пользователей

Fig. 13. Temperature regime taking into account the activity of users

Вторая стратегия позволяет отслеживать состояние «дом пуст», а не активность пользователей. Для этого к системе подключается датчик, реагирующий на использование входной двери. Таким образом, система отключает отопление в те моменты, когда в помещении никого нет. Возможный температурный режим изображен на рис. 14.

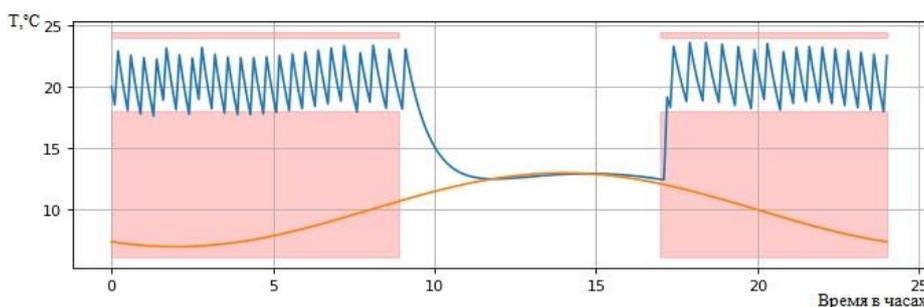


Рис. 14. Температурный режим с учетом наличия людей в помещении

Fig. 14. Temperature regime, taking into account the presence of people in the room

Можно заметить, что предложенные стратегии не противоречат друг другу, поэтому имеет смысл проверить их совместную работу. В этом случае система определяет состояние «дом пуст» согласно предыдущей стратегии. Если же такое состояние не зафиксировано, то система действует в зависимости от показаний датчиков движения. Соответствующий температурный режим в течение суток показан на рис. 15.

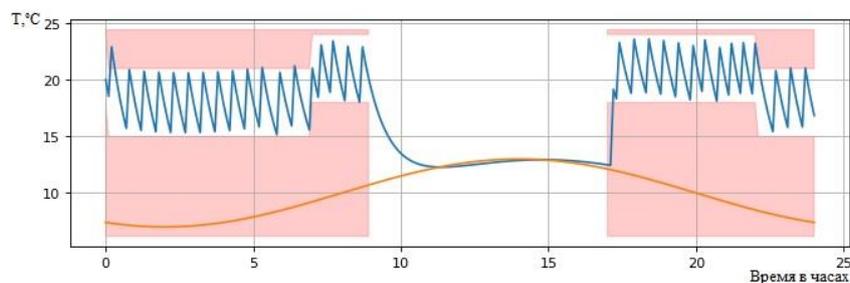


Рис. 15. Температурный режим при использовании обеих стратегий одновременно

Fig. 15. Temperature conditions when using both strategies

Проведение имитационных экспериментов и их результаты. Для демонстрации возможности проведения имитационных экспериментов, а также анализа и оценки использования для этого Node-RED и Yandex IoT Core была поставлена задача сравнить несколько сценариев управления устройствами ОВиК, отличающихся степенью и способом учета поведения пользователей приложения.

Основной метрикой функциональности систем управления энергопотреблением является общая стоимость затраченной энергии. Для оценки качества и перспективности сформулированных выше сценариев был поставлен эксперимент: система работает в течение 24 ч по виртуальному времени. При этом для моделирования выбран размер шага в 6 мин, т. е. 10 шагов в час. Формулы, включающие в себя время, используют именно внутрисистемное время, а не количество шагов, если не указано иначе. Температура окружающей среды изменяется согласно выражению

$$T_{\text{внеш}} = 10 + 3 \cos\left(\frac{\pi}{12}(t - 14)\right), \quad (4)$$

где t – текущий временной шаг в часах.

Поведение пользователя моделируется следующим образом: пользователь спит с 22:00 до 7:00, активен с 7:00 до 9:00 и с 17:00 до 22:00, остальное время находится вне контролируемого помещения. Целью пользователя является минимизация энергопотребления устройств ОВиК. Вместе с тем требования к комфорту, предъявляемые пользователями, зависят от ряда состояний: когда жители активны, температура не должна превышать 24 °C и не опускаться ниже 18 °C; когда жители спят, минимальная допустимая температура понижается до 15 °C; если дом пуст, то требований по температуре нет.

Гипотеза заключается в том, что базовый сценарий окажется наименее эффективным в контексте описанных выше критериев, самой же эффективной будет система управления, использующая обе стратегии уменьшения энергопотребления. В результате проведенного имитационного эксперимента были получены следующие результаты: общая стоимость затраченной энергии составила 14,87 условных единиц для базового сценария, для стратегии с учетом активности – 12,64, для стратегии с учетом наличия людей в помещении – 11,52, для объединенной стратегии – 9,88.

На рис. 16 можно видеть динамическое накопление общей стоимости энергии. В частности, заметим, что стратегии, включающие отслеживание наличия людей в помещении, выигрывают за счет длительных промежутков без употребления энергии. Таким образом, выдвинутая гипотеза подтвердилась.

Дополнительно оценим тестируемые стратегии оптимизации с точки зрения комфорта пользователя. Для того чтобы сравнить оценки при разной длительности эксперимента, рассчитаем среднее значение:

$$M = \frac{1}{N} \sum_{i=0}^N m(T, h_i), \quad (5)$$

где N – количество шагов моделирования, $m(T, h_i)$ – функция оценки при температуре T на шаге h_i .

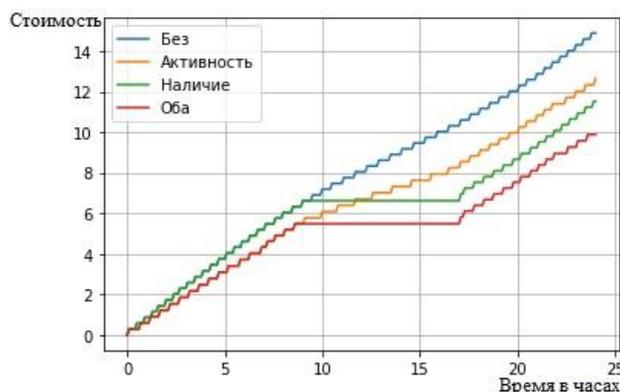


Рис. 16. Накопление потребления энергии

Fig. 16. Accumulation of energy consumption

Использовались два варианта расчета i -й оценки. При расчете по формуле (6) оценка дается в зависимости от количества единиц времени, в течение которых наблюдалась температура за пределами требуемых значений:

$$m(T,h) = \begin{cases} T_H(h) - T(h), & \text{если } T(h) < T_H(h); \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h); \\ T(h) - T_B(h), & \text{если } T(h) > T_B(h), \end{cases} \quad (6)$$

где $T_H(h)$ – нижняя граница допустимой температуры на шаге тестирования h , $T(h)$ – текущая температура на шаге симуляции h , $T_B(h)$ – верхняя граница допустимой температуры на шаге симуляции h . Другую оценку комфорта можно получить с помощью формулы, по которой рассчитывается среднее отклонение от целевой температурной области:

$$m(T,h) = \begin{cases} 1, & \text{если } T(h) < T_H(h) \text{ или } T(h) > T_B(h); \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h). \end{cases} \quad (7)$$

Полученные результаты представлены в табл. 1. Видно, что наилучшие результаты показал базовый (первый) сценарий, однако третий и четвертый сценарии по результатам незначительно отличаются от базового. В целом отметим, что значительных отклонений от требуемых значений не наблюдалось. Данный факт подтверждает возможность поддержки достаточного уровня комфорта всеми сценариями. Наиболее часто нарушения наблюдаются при переходе между состояниями (см. рис. 13–15). Это связано с тем, что помещение не успевает достаточно быстро нагреться или остыть. Возможным решением будет уточнение условий для управления обогревателем, чтобы система управления более плавно изменяла температуру.

Таблица 1
 Оценка комфорта пользователя для каждого сценария

Table 1
 Assessment of user comfort for each scenario

Сценарий Scenario	Доля времени вне заданных пределов Fraction of time outside the specified limits	Среднее отклонение от заданных пределов Average deviation from the specified limits
Первый (базовый)	0,0415	0,0072
Второй (активность жителей)	0,1112	0,0472
Третий (наличие жителей)	0,0456	0,0090
Четвертый (обе стратегии второго и третьего сценариев)	0,0539	0,0196

По итогам предварительного анализа эксперимента отметим, что в целях сравнения сценариев в вычислениях были приняты наиболее широкие границы. Таким образом, требования по комфорту были более свободными, чем в отдельных сценариях. Обратим внимание на то, что первый и третий сценарии поддерживают достаточно высокую температуру в ночное время, а это может быть нежелательным для некоторых пользователей.

Для проверки полученных результатов также был проведен более длительный эксперимент продолжительностью семь внутрисистемных дней. Показания датчиков моделировались аналогичным образом, данные представлены в табл. 2.

Таблица 2
Результаты длительного эксперимента

Table 2
Results of a long-term experiment

Сценарий <i>Scenario</i>	Потребление энергии <i>Power consumption</i>		Комфорт <i>Comfort</i>	
	Общее <i>General</i>	Среднее за день <i>Daily Average</i>	По времени <i>By time</i>	По отклонению <i>By deviation</i>
Первый	104,13	14,87	0,0464	0,0090
Второй	87,62	12,52	0,1201	0,0573
Третий	78,89	11,27	0,0500	0,0136
Четвертый	70,99	10,14	0,0577	0,0197

Из табл. 2 видно, что показатели изменились незначительно, причем если абсолютные значения и изменились, то отношения между рассматриваемыми моделями – нет. Это позволяет сделать вывод о справедливости результатов первого эксперимента.

Обсуждение результатов имитационных экспериментов и практические выводы. По одной из классификаций требования при разработке программного обеспечения (ПО) делятся на функциональные и нефункциональные. Под функциональностью будем понимать соответствие результатов работы ПО ожиданиям пользователя. Целью эксперимента являлась первоначальная оценка возможностей среды имитационного моделирования компонентов и приложений в Интернете вещей на основе инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core. В рассмотренном примере приложения Интернета вещей функциональные требования состояли в снижении стоимости энергопотребления и в нахождении показателей температурного комфорта жителей дома в заданных пределах. Результаты проведенных имитационных экспериментов говорят о том, что предлагаемая среда моделирования и выбранный пример приложения удовлетворяют обозначенным требованиям функциональности.

Среди нефункциональных требований важно выделить требования к надежности и эффективности. При всей ограниченности ресурсов компонентов и приложений в Интернете вещей их реакция на изменение окружающей среды должна быть своевременной. Нежелательным поведением приложения в выбранном примере могло бы быть постоянное нахождение обогревателя во включенном состоянии. В результате температура в помещении выйдет за пределы комфортных значений. Кроме того, такая система отопления была бы неэффективной в своем использовании энергии. Ни в одном из проведенных экспериментов не наблюдалось значительного выхода за пределы требуемых значений, что предварительно свидетельствует о надежности и эффективности разработанных прототипов.

Другие категории требований при разработке ПО бывает сложно оценить программным образом без участия человека. С точки зрения удобства использования отметим, что разработанное моделируемое приложение не требует участия человека для своей работы, за исключением установки требуемой температуры. Тем не менее существует риск неверного применения данного приложения при попытках ручного управления отоплением, что приведет к снижению эффективности приложения.

Можно сделать практический вывод, что предлагаемая среда моделирования и рассматриваемый пример моделируемого приложения удовлетворяют базовым требованиям при разработ-

ке ПО. Вместе с тем следует провести дополнительные эксперименты и дать более глубокий анализ их результатов, чтобы сформулировать рекомендации по выбору между рассмотренными сценариями. Дальнейшая исследовательская работа будет заключаться в разработке моделей, проведении экспериментов и анализе результатов для других приложений в Интернете вещей.

Заключение. В настоящей публикации авторы представили исследование по разработке и первоначальной оценке возможностей среды имитационного моделирования компонентов и приложений в Интернете вещей на основе интегрированного программного комплекса, состоящего из инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core. Для проверки работы данной среды в качестве примера приложения была выбрана система управления энергопотреблением жилых помещений. Для демонстрации ее функциональности проведена серия имитационных экспериментов, которые включили в себя моделирование времени, условий внешней среды, теплопотерь, режимов работы отопительного оборудования и поведения жителей дома. В результате первоначальных экспериментов была обоснована и подтверждена стратегия оптимизации расхода энергии отопительных приборов на основании показателей температурного комфорта и поведения жителей дома.

Вклад авторов. *А. А. Андрушевич* – обоснование концепции и анализ актуальности статьи, консультирование по вопросам создания среды имитационного моделирования на основе Node-RED и Yandex Cloud / Yandex IoT Core, формулирование выводов; *И. С. Войтешиенко* – сбор, анализ и интерпретация результатов работы, критический пересмотр содержания текста, утверждение окончательного варианта статьи для публикации; *О. Ю. Емельянова* – существенный вклад в реализацию работы, написание текста.

Список использованных источников

1. Сурков, А. Знакомство с Node-RED и потоковое программирование в Yandex IoT Core [Электронный ресурс] / А. Сурков. – Режим доступа: <https://habr.com/ru/company/yandex/blog/519600/>. – Дата доступа: 05.11.2021.
2. Акопов, А. С. Имитационное моделирование / А. С. Акопов. – М. : Юрайт, 2016. – 389 с.
3. Stojkoska, B. L. R. A review of Internet of Things for smart home: Challenges and solutions / B. L. R. Stojkoska, K. V. Trivodaliev // *J. of Cleaner Production*. – 2017. – Vol. 140. – P. 1454–1464.
4. Энергетический баланс Республики Беларусь : стат. сборник / Нац. Стат. Комитет Респ. Беларусь ; редкол. : И. В. Медведева (председатель) [и др.]. – Минск, 2020. – 152 с.
5. Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior / M. Pritoni [et al.] // *Energy Research & Social Science*. – 2015. – Vol. 58. – P. 190–197.
6. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments / J. Pan [et al.] // *IEEE Internet of Things J.* – 2015. – Vol. 2. – P. 527–537.
7. Okamoto-Mizuno, K. Effects of thermal environment on sleep and circadian rhythm / K. Okamoto-Mizuno, K. Mizuno // *J. of Physiological Anthropology*. – 2012. – Vol. 31. – P. 105–114.

References

1. Surkov A. Znakomstvo s Node-RED i potokovoe programmirovanie v Yandex IoT Core. *Introduction to Node-RED and Streaming Programming in Yandex IoT Core*. Available at: <https://habr.com/ru/company/yandex/blog/519600/> (accessed 05.11.2021) (In Russ.).
2. Akopov A. S. Imitacionnoe modelirovanie. *Simulation*. Moscow, Jurajt, 2016, 389 p. (In Russ.).
3. Stojkoska B. L. R., Trivodaliev K. V. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 2017, vol. 140, pp. 1454–1464.
4. Jenergeticheskij balans Respubliki Belarus': statisticheskij sbornik. *Energy Balance of the Republic of Belarus: Statistical Compendium*. Nacional'nyj statisticheskij komitet Respubliki Belarus'; In I. V. Medvedeva, I. S. Kangro, Zh. N. Vasilevskaja, O. A. Dovnar, E. I. Kuharevich, ..., A. S. Snetkov (eds.). Minsk, 2020, 152 p. (In Russ.).

5. Pritoni M., Meier A. K., Aragon C., Perry D., Peffer T. Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior. *Energy Research & Social Science*, 2015, vol. 8, pp. 190–197.

6. Pan J., Jain R., Vu T., Paul S. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments. *IEEE Internet of Things Journal*, 2015, vol. 2, pp. 527–537.

7. Okamoto-Mizuno K., Mizuno K. Effects of thermal environment on sleep and circadian rhythm. *Journal of Physiological Anthropology*, 2012, vol. 31, pp. 105–114.

Информация об авторах

Андрушевич Алексей Андреевич, младший научный сотрудник, научно-инженерное республиканское унитарное предприятие «Межотраслевой научно-практический центр систем идентификации и электронных деловых операций».

E-mail: andrushevich@bsu.by

Войтешенко Иосиф Станиславович, кандидат технических наук, доцент, доцент кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.

E-mail: voit@bsu.by

Емельянова Ольга Юрьевна, выпускница кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.

E-mail: volha.emelianova@gmail.com

Information about the authors

Alexey A. Andrushevich, Junior Researcher, Scientific Engineering Republican Unitary Enterprise "Interbranches Research & Development Centre for Identification Systems and e-Business Operations".

E-mail: andrushevich@bsu.by

Iosif S. Vojteshenko, Ph. D. (Eng.), Assoc. Prof., Assoc. Prof. of the Department of Programming Technologies of the Faculty of Applied Mathematics and Informatics, Belarusian State University.

E-mail: voit@bsu.by

Olga Ju. Emelianova, Graduate of the Department of Programming Technologies, Faculty of Applied Mathematics and Informatics, Belarusian State University.

E-mail: volha.emelianova@gmail.com

Правила для авторов

Редакция журнала «Информатика» просит авторов руководствоваться приведенными ниже правилами.

I. Статьи принимаются в редакцию через электронную систему подачи по адресу <http://inf.grid.by> в формате файлов текстовых редакторов Microsoft Word. Объем оригинальной статьи – от 8 до 16 стр., включая рисунки, таблицы и достаточное количество наиболее актуальных ссылок; объем обзорной статьи – от 16 до 32 стр., включая все основные ссылки. Текст набирается с переносами, шрифт Times New Roman 11 пт, интервал между строками – одинарный, абзацный отступ 0,5 см, поля по 2,5 см со всех сторон.

Материал статьи должен быть четко структурированным: Введение; основные разделы, в которых изложены цели и задачи, методы, результаты; Заключение (выводы).

II. Статьи о результатах работ, проведенных в научных учреждениях, должны иметь разрешение на публикацию (сопроводительное письмо за подписью руководителя или выписку из заседания ученого совета, отдела или кафедры, акт экспертизы).

III. Статьи в обязательном порядке должны включать аннотацию, ключевые слова, список литературы, информацию об авторах на русском и английском языках.

На титульной странице располагаются следующие метаданные:

1. Индекс по универсальной десятичной классификации (УДК); на русском и английском языках тип статьи (оригинальная или обзорная), название статьи, инициалы и фамилии всех авторов, полное наименование учреждений, где работают авторы, с указанием почтового адреса, при наличии указывается ученая степень и ORCID, e-mail ответственного лица.

2. Аннотация (Abstract) объемом 150–250 слов в оригинальной статье должна быть структурирована отдельными подразделами: Цели, Методы, Результаты, Заключение, а также максимально характеризовать содержательную часть рукописи. Сюда не следует включать впервые введенные термины, аббревиатуры (за исключением общеизвестных), ссылки на литературу.

3. Ключевые слова (Keywords) – наиболее значимые слова или словосочетания по теме работы, отражающие специфику темы, объекты и результаты исследования; перечень ключевых слов должен содержать 5–10 слов.

4. В разделе Благодарности (Acknowledgements) указываются все источники финансирования исследования, а также благодарности людям, которые участвовали в работе над статьей.

5. Автор обязан уведомить редакцию о реальном или потенциальном конфликте интересов, включив информацию в раздел Конфликт интересов (Conflict of interest).

6. Формулы, рисунки, таблицы в статье нумеруются в соответствии с порядком их упоминания в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Рисунки должны быть выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, и названия таблиц набираются шрифтом гарнитуры основного текста размером 9 пт. Перевод подрисуночной подписи и пояснений к рисунку, а также перевод названия таблицы, заголовки строк или столбцов располагаются курсивом после русскоязычной версии.

7. Набор формул выполняется в формульном редакторе Microsoft Equation или Math Type. Прямым шрифтом набираются: греческие и русские буквы; математические символы (\sin , \lg , ∞); символы химических элементов (C, Cl, CH₃); цифры (римские и арабские); индексы (верхние и нижние), являющиеся сокращениями слов. Курсивом набираются латинские буквы, символы физических величин (в том числе и в индексе).

8. Список использованной литературы оформляется в соответствии с требованиями Высшей аттестационной комиссии Республики Беларусь (ГОСТ 7.5–2008). Номер литературной ссылки в тексте дается порядковым номером в квадратных скобках. Ссылаться на неопубликованные работы не допускается.

10. Отдельно оформляется References со следующей структурой: авторы (транслитерация), транслитерированное название монографии, *Перевод названия монографии на английский язык*. Выходные данные с обозначениями на английском языке. От транслитераций названий статей можно отказаться.

Ссылки на учебно-методическую литературу, ГОСТы, авторефераты, статистические отчеты в список не включаются, а оформляются в виде сносок (с подробными рекомендациями можно ознакомиться на сайте журнала в разделе Правила для авторов).

11. В разделе Информация об авторах (Information about the authors) приводятся ФИО авторов полностью, ученая степень, звание, должность, название организации, ORCID (при наличии).

IV. Все поступающие в редакцию рукописи проходят предварительную проверку на соответствие Правилам для авторов. Статья может быть возвращена автору на доработку с просьбой устранить недостатки или дополнить информацию. После проверки на соответствие правилам статья направляется рецензенту с указанием сроков рецензирования.

V. При наличии замечаний рецензента автору предоставляется определенное время на доработку рукописи. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все замечания. Окончательное решение о публикации или отклонении рукописи принимается редколлегией журнала. При положительном заключении рецензента статья передается редактору для подготовки к печати. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

VI. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

VII. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее или принятых к публикации другими изданиями.

ИНДЕКСЫ

00827

для индивидуальных
подписчиков

008272

для предприятий
и организаций