

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

ИНФОРМАТИКА

INFORMATICS

20
лет

ЮБИЛЕЙ
ЖУРНАЛА

ТОМ 21
VOL. 21

3 | 2024

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 for authors 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 21, № 3, июль-сентябрь 2024

Ежеквартальный научный журнал

Издается с января 2004 г.

Учредитель и издатель – государственное научное учреждение «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, Центр «Квантовая оптика и квантовая информатика» Института физики им. Б. И. Степанова 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)

Скляр 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)

Черемисинова Людмила Дмитриевна, д-р техн. наук, проф., ОИПИ НАН Беларуси (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 21, № 3, июль-сентябрь 2024

Ответственный за выпуск *Мойсейчик Светлана Сергеевна*
Редактор *Гончаренко Галина Борисовна*
Компьютерная верстка *Бутевич Ольга Борисовна*

Сдано в набор 21.08.2024. Подписано в печать 19.09.2024. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография. Усл. печ. л. 12,5. Уч.-изд. л. 12,3. Тираж 40 экз. Заказ 5.

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 № 38200000016516 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. 21, no. 3, Jule-September 2024

Published quarterly

Issued since January 2004

Founder and publisher – State Scientific Institution "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, Center of Quantum Optics and Quantum
Information of B. I. Stepanov Institute of Physics 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. (Mil.Eng.), 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)

Ljudmila D. Cheremisinova, D. Sc. (Eng.), Prof., UIIP NASB (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. 21, no. 3, July-September 2024

Issue Head *Sviatlana S. Maiseichyk*

Editor *Halina B. Hancharenka*

Computer Imposition *Volha B. Butsevich*

Sent for press 21.08.2024. Output 19.09.2024. Format 60×84 1/8. Offset paper. Headset Times. Riesography. Printed sheets 12,5. Publisher's signatures 12,3. Circulation 40 copies. Order 5.

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. 38200000016516 dated 18.12.13.

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

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

СОДЕРЖАНИЕ

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

- Потгосин Ю. В.** Декомпозиция параллельного автомата в сеть последовательных автоматов и энергосберегающее кодирование их состояний при асинхронной реализации 7
- Черемисинов Д. И., Черемисинова Л. Д.** Извлечение логических сетей при декомпиляции описаний КМОП-схем на уровне транзисторов..... 23

ЗАЩИТА ИНФОРМАЦИИ И НАДЕЖНОСТЬ СИСТЕМ

- Янчевский В. И., Говорушко И. О., Матвеев Г. В.** Разделение секрета в специальной линейной группе 39

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Ким Т. Ю., Прокопович Г. А.** Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта..... 48
- Венгеренко В. В., Инютин А. В.** Гибридный контроль дефектов печатных плат 63

БИОИНФОРМАТИКА

- Воробьев Д. А., Карпенко А. Д., Тузиков А. В., Андрианов А. М.** Адаптация архитектуры нейронной сети REINVENT для генерации потенциальных ингибиторов проникновения ВИЧ-1 80

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Липницкий С. Ф., Степура Л. В.** Обработка результатов веб-поиска в системе информационной поддержки процессов принятия решений..... 94

УЧЕНЫЕ БЕЛАРУСИ

- Садыхов Рауф Хосровович (21.10.1944–15.03.2015)** 106

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

CONTENTS

LOGICAL DESIGN

- Pottosin Yu. V.** Decomposition of a parallel automaton into a net of sequential automata and low power state assignment of them at asynchronous implementation 7
- Cheremisinov D. I., Cheremisinova L. D.** Extraction of logical networks during decompiling transistor-level CMOS circuit descriptions 23

INFORMATION PROTECTION AND SYSTEM RELIABILITY

- Yanchevskii V. I., Havarushka I. A., Matveev G. V.** Secret sharing in a special linear group 39

INTELLIGENT SYSTEMS

- Kim T. Yu., Prakapovich R. A.** Development of an imitation learning method for a neural network system of mobile robot's movement on example of the maze solving 48
- Vengerenko V. V., Inyutin A. V.** Hybrid inspection of printed board defects 63

BIOINFORMATICS

- Varabyeu D. A., Karpenko A. D., Tuzikov A. V., Andrianov A. M.** Adaptation of the REINVENT neural network architecture to generate potential HIV-1 entry inhibitors 80

INFORMATION TECHNOLOGIES

- Lipnitsky S. F., Stepura L.V.** Web search results processing in an information support system for decision-making processes 94

SCIENTISTS OF BELARUS

- Sadykhov Rauf Khosrovovich (21.10.1944–15.03.2015)** 106

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.711
<https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

Оригинальная статья
Original Article

Декомпозиция параллельного автомата в сеть последовательных автоматов и энергосберегающее кодирование их состояний при асинхронной реализации

Ю. В. Поттосин

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

Аннотация

Цели. Рассматриваются задача декомпозиции параллельного автомата в сеть последовательных автоматов при асинхронной реализации и противогоночное энергосберегающее кодирование их состояний. Целью работы является исследование возможности применения приема декомпозиции при кодировании частичных состояний для снижения размерности задачи с учетом особенностей асинхронной реализации.

Методы. Заданный параллельный автомат разлагается в сеть последовательных асинхронных автоматов, состояния которых кодируются затем троичными векторами. Снижение энергопотребления проектируемого устройства достигается за счет уменьшения интенсивности переключений элементов памяти, которая оценивается с помощью вероятностей переходов между состояниями автомата. Кодирование состояний сводится к задаче минимального взвешенного покрытия. Вероятности переходов между состояниями вычисляются путем решения системы линейных уравнений согласно методу Чэпмена – Колмогорова.

Результаты. Описан способ построения сети из последовательных асинхронных автоматов, реализующей заданный параллельный автомат. Затронута задача минимизации межкомпонентных связей в сети.

Заключение. Использование декомпозиции параллельного автомата позволяет снизить размерность трудоемкой задачи кодирования состояний. Предлагаемый метод предназначен для применения в системах автоматизированного проектирования дискретных устройств.

Ключевые слова: параллельный автомат, частичное состояние, декомпозиция автоматов, кодирование состояний автомата, вероятность перехода между состояниями, взвешенное покрытие

Для цитирования. Поттосин, Ю. В. Декомпозиция параллельного автомата в сеть последовательных автоматов и энергосберегающее кодирование их состояний при асинхронной реализации / Ю. В. Поттосин // Информатика. – 2024. – Т. 21, № 3. – С. 7–22. <https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 10.06.2024

Подписана в печать | Accepted 27.06.2024

Опубликована | Published 30.09.2024

Decomposition of a parallel automaton into a net of sequential automata and low power state assignment of them at asynchronous implementation

Yuri V. Pottosin

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

Abstract

Objectives. The problems of decomposition of a parallel automaton into a net of sequential automata at synchronous realization and low power race free state assignment of them are considered. The objective of the paper is to investigate the possibilities of applying decomposition in state assignment of partial states in order to decrease the problem dimension taking into account the peculiar properties of the asynchronous realization.

Methods. The given parallel automaton is decomposed into a net of sequential asynchronous automata whose states are assigned then with ternary vectors. The power consumption lowering of the designed device is achieved by lowering the intensity of its memory elements switching that is appreciated by probabilities of transitions between the states of the automaton. The state assignment is reduced to the problem of minimal weighted cover. The probabilities of transitions between sets are calculated by means of solving a system of linear equations according to the Chapman – Kolmogorov method.

Results. A method to construct a net of sequential asynchronous automata that realizes the given parallel automaton is described. The paper touches upon the problem of minimization of interconnections in the net.

Conclusion. Applying parallel automaton decomposition allows decreasing the dimension of the laborious problem of state assignment. The proposed method is intended for application in computer aided systems for design of discrete devices.

Keywords: parallel automaton, partial state, decomposition of automata, state assignment of automata, probability of transition between states, weighted cover

For citation. Pottosin Yu. V. *Decomposition of a parallel automaton into a net of sequential automata and low power state assignment of them at asynchronous implementation*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 7–22 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

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

Введение. Одним из способов снижения размерности задач при синтезе дискретных устройств является разложение исходного устройства на блоки. Этой цели служит решение задачи декомпозиции автоматов. Параллельный автомат является функциональной моделью дискретного устройства управления с достаточной степенью удобства, представляющей параллелизм управляемых взаимодействующих процессов [1]. Такая модель близка к широко известной сети Петри [2]. В отличие от последовательного автомата, который в какой-то момент времени находится только в одном состоянии, параллельный автомат может одновременно находиться в нескольких частичных состояниях. Любая совокупность частичных состояний, в которых параллельный автомат может находиться одновременно, не находясь ни в каком другом состоянии, называется глобальным состоянием. Синтез логической структуры устройства, поведение которого представлено параллельным автоматом, лежит через кодирование его частичных состояний. Решение этой задачи можно заменить нахождением всех его глобальных состояний и последующим их кодированием каким-либо из известных методов, разработанных применительно к последовательным автоматам. Однако при сравнительно небольшом количестве частичных состояний параллельного автомата число его глобальных состояний, как показано в работе [3], может быть настолько велико, что задачу их кодирования невозможно решить за приемлемое время. Кроме того, при асинхронной реализации параллельный автомат не всегда моделируется последовательным автоматом [3], и указанный подход вообще не может быть

использован. Поэтому были разработаны методы специально для кодирования частичных состояний параллельного автомата [4–6]. Они используют приемы, отличающиеся от тех, которые применяются при кодировании состояний последовательных автоматов. В настоящей работе предлагается сводить кодирование частичных состояний параллельного автомата к совместному кодированию состояний компонентных автоматов сети, реализующей заданный параллельный автомат. Каждый из компонентных автоматов данной сети является последовательным асинхронным автоматом, и поэтому для кодирования его состояний можно использовать какой-либо из известных методов, предназначенных для таких автоматов.

Одним из важных критериев оптимизации при проектировании дискретных устройств является величина энергии, потребляемой проектируемым устройством. Это обусловлено, с одной стороны, стремлением увеличить время действия источника энергии в портативных приборах, а с другой – стремлением снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем.

Как отмечено в работах [7, 8], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений логических элементов и элементов памяти. Это дает возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться на этапе кодирования состояний автомата, т. е. когда абстрактным символам состояний приписываются булевы векторы, что необходимо для получения системы булевых функций, представляющей структурную модель проектируемого устройства. Для последовательных автоматов при синхронной и асинхронной реализации эта задача решалась в работах [9–14].

Метод энергосберегающего кодирования частичных состояний параллельного автомата при синхронной реализации описан в статье [15], где используется подход из работы [16], сводящий данную задачу к поиску покрытия графа его полными двудольными подграфами. Декомпозиционный подход к энергосберегающему кодированию состояний при синхронной реализации представлен в статье [17], в которой показано, как применять для совместного кодирования состояний компонентных автоматов сети, реализующей параллельный автомат, итеративный метод, описанный в работе [18].

Использование приема декомпозиции параллельного автомата, т. е. разложения его в сеть последовательных автоматов, позволяет снизить, как было сказано, размерность задачи. Кроме того, разложение проектируемого устройства на блоки дает возможность снизить энергопотребление путем блокировки работы некоторых блоков [19].

Описание модели. Параллельный автомат состоит из следующих объектов: $Q = \{q_1, q_2, \dots, q_l\}$ – множества *частичных состояний*, $X = \{x_1, x_2, \dots, x_n\}$ – множества входных булевых переменных, $Y = \{y_1, y_2, \dots, y_m\}$ – множества выходных булевых переменных и последовательности строк-переходов $T = \{\tau_1, \tau_2, \dots, \tau_l\}$, имеющих вид [3]

$$\tau_i = S_i : - K_i \rightarrow K_i' \rightarrow S_i', \quad (1)$$

где S_i и S_i' – подмножества множества Q ; K_i – элементарная конъюнкция переменных из множества X и K_i' – элементарная конъюнкция переменных из множества Y .

Все множество частичных состояний, в которых рассматриваемый параллельный автомат находится в некоторый момент времени, называется *глобальным состоянием*. Смысл выражения (1) заключается в следующем. Если автомат находится одновременно в состояниях, составляющих множество S_i , и булевы переменные приняли значения, обращающие конъюнкцию K_i в единицу, то конъюнкция K_i' приобретает значение 1 и автомат переходит из частичных состояний, составляющих множество S_i , в частичные состояния, составляющие множество S_i' . Другими словами, пусть $P = \{P_1, P_2, \dots, P_p\}$ – множество всех достижимых глобальных состояний заданного параллельного автомата. Тогда если $S_i \subseteq P_g$, где P_g – текущее глобальное состояние автомата, и моделируемое устройство принимает на входе двоичные сигналы, обращающие конъюнкцию K_i в единицу, то глобальным состоянием в следующий момент времени будет $P_h = (P_g \setminus S_i) \cup S_i'$, а на выходе устройство выдаст двоичные сигналы, обращающие конъюнкцию K_i' в единицу. Любая из конъюнкций K_i или K_i' может отсутствовать в приведенной строке.

Отсутствие K_i означает ее тождественное равенство единице. Отсутствие K_i' означает в зависимости от интерпретации данной модели либо то, что все переменные из множества Y обращаются в нуль, либо то, что значения сигналов на выходе не меняются. Так же, как для последовательного автомата, возможны синхронная и асинхронная реализации параллельного автомата. Далее будет рассматриваться асинхронная реализация. При такой реализации время такта функционирования автомата не фиксировано и определяется изменениями внешнего входного сигнала. При поступлении сигнала на вход автомат приходит в устойчивые частичные состояния, из которых он не уходит до очередного изменения входного сигнала. Это изменение не должно произойти раньше, чем установятся частичные состояния. В настоящей работе не рассматриваются выходные сигналы и конъюнкция K_i' будет опускаться. Проблема устранения «дребезжания» выходных сигналов, т. е. многократного изменения значений, во время переходов между состояниями решается на уровне схемы, как, например, в работе [20].

В описываемую модель внесены следующие ограничения и допущения:

1. Вводится начальное глобальное состояние в виде одноэлементного множества. Для определенности можно взять $\{q_1\}$.

2. Для двух различных строк, i -й и j -й, если $S_i \cap S_j \neq \emptyset$, то $S_i = S_j$.

3. При переходе из одного устойчивого частичного состояния в другое автомат может проходить некоторую последовательность неустойчивых состояний. Время пребывания в любом из них пренебрежимо мало.

Не будем, как сказано выше, обращать внимание на выходные сигналы, ограничимся рассмотрением переключений элементов памяти.

Примером параллельного автомата может служить следующая последовательность строк:

$$\begin{aligned}\tau_1 &= 1: -x_2 \rightarrow 10; \\ \tau_2 &= 10: -\bar{x}_2 \rightarrow 2.3.4; \\ \tau_3 &= 2: -x_1 \rightarrow 5.6; \\ \tau_4 &= 3.5: -x_2 \rightarrow 8; \\ \tau_5 &= 4: -x_2 \rightarrow 7; \\ \tau_6 &= 4: -\bar{x}_1 \rightarrow 9; \\ \tau_7 &= 7: -\bar{x}_2 \rightarrow 9; \\ \tau_8 &= 6.8.9: -x_1 \rightarrow 1.\end{aligned}$$

Данный параллельный автомат допускает асинхронную реализацию и моделирование асинхронным последовательным автоматом. Здесь $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ и $X = \{x_1, x_2\}$. За начальное глобальное состояние примем одноэлементное множество $\{1\}$. Первая строка (переход τ_1) означает, что, если $x_2 = 1$, то независимо от того, какое значение имеет x_1 , автомат уходит из состояния $\{1\}$ и входит в состояние $\{10\}$, которое также является глобальным. При $x_2 = 0$ автомат остается в состоянии $\{1\}$. Из состояния $\{10\}$ при $x_2 = 0$ автомат переходит в частичные состояния 2, 3 и 4, но при $x_1 = 0$ состояние 4 оказывается неустойчивым и меняется на частичное состояние 9 (переход τ_6). Устойчивым глобальным состоянием тогда является $\{2, 3, 9\}$. При $x_1 = 1$ состояние 2 оказывается неустойчивым, из него автомат переходит в частичные состояния 5 и 6 (переход τ_3) и оказывается в устойчивом глобальном состоянии $\{3, 4, 5, 6\}$. Проследив таким образом функционирование данного параллельного автомата, получим еще устойчивые глобальные состояния $\{3, 5, 6, 9\}$, $\{6, 7, 8\}$ и $\{6, 8, 9\}$. Далее будем рассматривать только устойчивые частичные и глобальные состояния.

Построение сети последовательных автоматов, реализующей заданный параллельный автомат. Решение данной задачи подробно представлено в статье [17]. Пусть задан некоторый параллельный автомат B , описанием которого является последовательность строк вида (1). Рассмотрим некоторую сеть $N = (X, A_1, A_2, \dots, A_n)$, где X – множество входных булевых переменных, тех же, что присутствуют в задании автомата B , и A_1, A_2, \dots, A_n – компонентные последовательные автоматы с множествами внутренних состояний Q_1, Q_2, \dots, Q_n . Каждый из автома-

тов A_j является автоматом без выходов, т. е. для него заданы только переходы, которые представим в форме, аналогичной выражению (1):

$$q^j : -\alpha_j(x_1, x_2, \dots, x_n, q^1, q^2, \dots, q^{j-1}, q^{j+1}, \dots, q^n) \rightarrow q^j, \quad (2)$$

где α_j – предикат над переменными из множества X и состояниями компонентных автоматов $A_1, A_2, \dots, A_{j-1}, A_{j+1}, \dots, A_n$, принимающий значение 1 при определенной комбинации значений некоторых входных переменных и определенной совокупности состояний некоторых компонентных автоматов. Если $\alpha_j = 1$, то автомат A_j переходит из состояния q^j в состояние q^j , в противном случае остается в состоянии q^j .

Сеть N реализует автомат B , если существует такое отображение φ множества $D \subseteq Q_1 \times Q_2 \times \dots \times Q_n$ в некоторую совокупность подмножеств множества Q , что для любого перехода (1) автомата B при $S_i \subseteq \varphi(q^1, q^2, \dots, q^n)$ имеем $S_i' \subseteq \varphi(q^1, q^2, \dots, q^n)$, если $\alpha_j = 1$, и $S_i' \subseteq \varphi(q^1, q^2, \dots, q^n)$, если $\alpha_j = 0$, согласно (2). Если в случае асинхронной реализации параллельный автомат B допускает моделирование последовательным автоматом, то можно построить последовательный автомат A , эквивалентный автомату B , с состояниями, соответствующими глобальным состояниям автомата B . Описанная сеть в этом случае реализует автомат A в смысле работ [21, 22].

Определим множество состояний Q_j компонентного автомата A_j ($j = 1, 2, \dots, n$) сети N следующим образом. Каждому состоянию $q^j \in Q_j$ поставим во взаимно однозначное соответствие некоторое частичное состояние q автомата B , причем среди всех частичных состояний, соответствующих состояниям из Q_j , не должно быть ни одной пары *параллельных* состояний, т. е. таких, в которых автомат B может находиться одновременно. Указанное соответствие обозначим $f_j(q^j) = q$. Пусть множество компонентных автоматов $\{A_1, A_2, \dots, A_n\}$ сети N достаточно для того, чтобы для каждого частичного состояния q автомата B существовала по крайней мере одна компонента A_j с таким состоянием q^j , что $f_j(q^j) = q$.

Переходы в компонентных автоматах определим согласно выражению (2) следующим образом. Каждой строке τ_i вида (1) из описания автомата B поставим в соответствие совокупность переходов в тех компонентных автоматах A_j , каждый из которых имеет такое состояние q^j , что $f_j(q^j) \in S_i \cup S_i'$. Для компонентного автомата A_j в качестве левой части выражения (2) возьмем то состояние q^j , для которого $f_j(q^j) \in S_i$. Таких состояний в Q_j ровно одно, поскольку по определению множества Q_j оно не содержит ни одной пары состояний, соответствующих параллельным частичным состояниям автомата B . Состояние q^j в выражении (2) определяем так, чтобы имело место $f_j(q^j) \in S_i'$. При этом полагаем, что $\alpha_j = 1$ тогда и только тогда, когда $K_i = 1$ и $\{f_1(q^1), f_2(q^2), \dots, f_{j-1}(q^{j-1}), f_{j+1}(q^{j+1}), \dots, f_n(q^n)\} \supseteq S_i'$.

Множество последовательных автоматов $\{A_1, A_2, \dots, A_n\}$, образующих сеть N , которая реализует заданный параллельный автомат B , строим следующим образом. Получим все максимальные множества взаимно непараллельных частичных состояний автомата B . Максимальных в том смысле, что любое частичное состояние, не принадлежащее какому-либо из этих множеств, оказывается параллельным хотя бы одному из принадлежащих ему. Далее следует получить кратчайшее покрытие (естественно потребовать минимум компонентных автоматов) этими множествами всех пар частичных состояний, связанных переходами. Каждое множество из полученного покрытия соответствует множеству состояний одного из компонентных последовательных автоматов, составляющих искомую сеть N .

В статье [23] описан алгоритм установления параллельности частичных состояний, который для параллельных автоматов рассматриваемого класса решает данную задачу за полиномиальное время. Как отмечается далее, в случае энергосберегающего кодирования приходится рассматривать глобальные состояния параллельного автомата и по ним легко установить параллельность частичных состояний. Для примера параллельного автомата, описание которого приведено выше и перечислены глобальные состояния, матрица отношения параллельности на множестве частичных состояний имеет вид

1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	2
0	1	0	1	1	1	0	0	1	0	3
0	0	1	0	1	1	0	0	0	0	4
0	0	1	1	0	1	0	0	1	0	5
0	0	1	1	1	0	1	1	1	0	6
0	0	0	0	0	1	0	1	0	0	7
0	0	0	0	0	1	1	0	1	0	8
0	1	1	0	1	1	0	1	0	0	9
0	0	0	0	0	0	0	0	0	0	10

Ее можно рассматривать как матрицу смежности графа, представляющего отношение параллельности. Упомянутые множества непараллельных частичных состояний соответствуют независимым множествам данного графа. Способы нахождения максимальных независимых множеств в графе описаны в работе [18]. В данном графе максимальными независимыми являются множества {1, 2, 4, 7, 10}, {1, 3, 7, 10}, {1, 2, 5, 7, 10}, {1, 2, 6, 10}, {1, 2, 4, 8, 10}, {1, 3, 8, 10}, {1, 2, 5, 8, 10} и {1, 4, 7, 9, 10}. Табл. 1 представляет собой таблицу покрытия, в которой строкам соответствуют максимальные множества взаимно непараллельных частичных состояний, а столбцам – пары частичных состояний, связанных переходами. Задача здесь решается тривиально: выбираются строки, содержащие единицу, единственную в каком-либо столбце, и они составляют искомое покрытие. В данном случае искомое покрытие составляют множества {1, 2, 6, 10}, {1, 3, 8, 10}, {1, 2, 5, 8, 10} и {1, 4, 7, 9, 10}.

Таблица 1
 Table 1

	1-6	1-8	1-9	1-10	2-5	2-6	2-10	3-8	3-10	4-7	4-9	4-10	5-8	7-9
1,2,4,7,10				1			1			1		1		
1,3,7,10				1					1					
1,2,5,7,10				1	1		1							
1,2,6,10	1			1		1	1							
1,2,4,8,10		1		1			1					1		
1,3,8,10		1		1				1	1					
1,2,5,8,10		1		1	1		1						1	
1,4,7,9,10			1	1						1	1	1		1

Состояния компонентных автоматов обозначим теми же символами, которые обозначают соответствующие им частичные состояния заданного параллельного автомата. Таким образом, искомую сеть составляют четыре автомата: A_1 с множеством состояний $Q_1 = \{1, 2, 5, 8, 10\}$, A_2 с множеством состояний $Q_2 = \{1, 2, 6, 10\}$, A_3 с множеством состояний $Q_3 = \{1, 3, 8, 10\}$ и A_4 с множеством состояний $Q_4 = \{1, 4, 7, 9, 10\}$. Переходы между состояниями определяются согласно выражению (2), как описано выше. Поведение получаемой сети N зададим в виде системы не полностью определенных функций $q^{1'}$, $q^{2'}$, $q^{3'}$ и $q^{4'}$, принимающих значения соответственно из множеств Q_1 , Q_2 , Q_3 и Q_4 . Их аргументами являются булевы переменные x_1 , x_2 и многозначные переменные q^1 , q^2 , q^3 и q^4 , принимающие значения из тех же множеств, что и указанные функции. Данную систему функций представляет табл. 2 – обычное табличное задание дискретных функций. В ней символ «–» обозначает безразличное значение переменной.

В результате кодирования состояний, т. е. замены каждой переменной q^j и, соответственно, $q^{j'}$ булевым вектором с компонентами z_i^j (соответственно $z_i^{j'}$), представляющими состояния двоичных элементов памяти, система функций, заданная табл. 2, преобразуется в систему булевых функций.

Метод противогоночного кодирования состояний асинхронного автомата. Для энерго-сберегающего противогоночного кодирования состояний асинхронного последовательного автомата используем способ, подробно описанный в статье [13].

Таблица 2
Table 2

Номер строки <i>Line number</i>	Внешний вход <i>External input</i>	Начальное состояние <i>Initial state</i>				Конечное состояние <i>Final state</i>				Номер строки <i>Line number</i>	Внешний вход <i>External input</i>	Начальное состояние <i>Initial state</i>				Конечное состояние <i>Final state</i>					
		x_1	x_2	q^1	q^2	q^3	q^4	$q^{1'}$	$q^{2'}$			$q^{3'}$	$q^{4'}$	q^1	q^2	q^3	q^4	$q^{1'}$	$q^{2'}$	$q^{3'}$	$q^{4'}$
1	- 0	1	1	1	1	1	1	1	1	12	1	0	5	6	3	4	5	6	3	4	
2	- 1	1	1	1	1	10	10	10	10	13	- 0	5	6	3	9	5	6	3	9		
3	- 1	10	10	10	10	10	10	10	10	14	0	1	5	6	3	9	8	6	8	9	
4	0	0	10	10	10	10	2	2	3	9	15	1	1	5	6	3	9	10	10	10	10
5	1	0	10	10	10	10	5	6	3	4	16	- 1	8	6	8	7	8	6	8	7	
6	0	-	2	2	3	9	2	2	3	9	17	1	0	8	6	8	7	1	1	1	1
7	1	0	2	2	3	9	5	6	3	9	18	0	0	8	6	8	7	8	6	8	9
8	1	1	2	2	3	9	10	10	10	10	19	0	-	8	6	8	9	8	6	8	9
9	0	0	5	6	3	4	5	6	3	9	20	1	1	8	6	8	9	10	10	10	10
10	0	1	5	6	3	4	8	6	8	9	21	1	0	8	6	8	9	1	1	1	1
11	1	1	5	6	3	4	8	6	8	7											

Явление одновременного переключения элементов памяти называется *состязаниями* или *гонками* элементов памяти [3]. Принято называть состязания *неопасными*, если все промежуточные состояния, в которых автомат может оказаться при переходе из одного состояния в другое под воздействием некоторого входного сигнала, являются неустойчивыми для этого сигнала, т. е. при любом порядке переключений элементов памяти автомат из некоторого состояния q_i переходит всегда в состояние q_j , определяемое функцией переходов. Если же при этом автомат может оказаться в некотором устойчивом состоянии q_k , отличном от q_j , то состязания называются *опасными*. Кодирование состояний, обеспечивающее отсутствие опасных состязаний (гонок), называется *противогоночным*.

Рассматриваются пары переходов между состояниями при одном и том же сигнале на входе. Например, при $x_1 = 0, x_2 = 1$ согласно строкам 2 и 10 табл. 2 у автоматов A_1, A_2, A_3 и A_4 имеются пары переходов $(1 \rightarrow 10, 5 \rightarrow 8), (1 \rightarrow 10, 6 \rightarrow 6), (1 \rightarrow 10, 3 \rightarrow 8)$ и $(1 \rightarrow 10, 4 \rightarrow 9)$ соответственно. Условие отсутствия опасных состязаний на паре переходов формулируется в виде троичного вектора, компонентам которого соответствуют состояния автомата, и компоненты, соответствующие состояниям одной пары, имеют значение 1, а другой пары – значение 0. Такими векторами для перечисленных пар переходов являются $(0 - 1 1 0), (0 - 1 0), (0 1 1 0)$ и $(0 1 - 1 0)$, где единицы и нули могут меняться местами. Все условия отсутствия опасных состязаний в виде описанных векторов составляют так называемую *матрицу условий*, в которой отсутствуют имплицитные строки [3]. Троичный вектор **a** имплицитно троичный вектор **b**, если **b** получается из **a** заменой некоторых нулей или единиц значением «-» и, возможно, инвертированием полученного результата. Например, вектор $(1 0 - - 1 0 1)$ имплицитно $(1 0 - - - 0 1)$ и $(0 1 - - - 1 -)$. Смысл этого отношения в том, что условие, представленное вектором **b**, автоматически выполняется при соблюдении условия, представленного вектором **a**. Для компонентных автоматов рассматриваемой сети матрицы условий имеют следующий вид:

$$\begin{matrix}
 \begin{matrix} 1 & 2 & 5 & 8 & 10 \\ \begin{bmatrix} 0 & 1 & - & - & 1 \\ 0 & - & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & - \\ 0 & 1 & - & - & 0 \\ 0 & - & 1 & 1 & 0 \\ - & 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 2 & 6 & 10 \\ \begin{bmatrix} 0 & 1 & - & 1 \\ 0 & - & 1 & 1 \\ 0 & 1 & 1 & - \\ 0 & 1 & - & 0 \\ 0 & - & 1 & 0 \\ - & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 3 & 8 & 10 \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 4 & 7 & 9 & 10 \\ \begin{bmatrix} 0 & - & - & 1 & 1 \\ 0 & 1 & 0 & - & 1 \\ 0 & 1 & - & 1 & 0 \\ 0 & - & 1 & 1 & - \\ 0 & 1 & 1 & - & 0 \\ 1 & 0 & - & 1 & 0 \\ - & 0 & 0 & 1 & 1 \\ - & 0 & 1 & 0 & - \\ 0 & - & 0 & 1 & - \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix}
 \end{matrix}
 \end{matrix}$$

Троичная матрица R *имплицирует* троичную матрицу S , если для каждой строки матрицы S в матрице R найдется имплицирующая ее строка. Задача противогоночного кодирования с минимизацией длины кода состояния сводится к нахождению матрицы с минимальным числом строк, имплицирующей матрицу условий и называемой *кратчайшей имплицирующей формой* матрицы условий. Столбцы этой матрицы будут представлять искомые коды состояний, а получаемая в результате ее транспонирования матрица называется *матрицей кодирования*. Строкам матрицы кодирования соответствуют состояния автомата, а столбцам – внутренние переменные. Строки этой матрицы представляют коды соответствующих состояний.

Кратчайшая имплицирующая форма матрицы условий находится следующим образом. Множество строк матрицы условий называется *совместимым*, если существует вектор, имплицирующий каждую строку этого множества. Совместимое множество называется *максимальным*, если оно не является собственным подмножеством другого совместимого множества. Необходимо найти кратчайшее покрытие множества строк матрицы условий максимальными совместимыми множествами. Каждому совместимому множеству соответствует вектор, имплицирующий все строки, принадлежащие этому множеству. Указанные векторы, соответствующие элементам полученного покрытия, в качестве строк составят кратчайшую имплицирующую форму заданной матрицы условий. Ниже приведены матрицы, строки которых представляют максимальные совместимые множества и соответствующие имплицирующие векторы. Справа каждой строки показаны номера имплицируемых ею строк матрицы условий.

$$\begin{array}{c}
 1 \ 2 \ 5 \ 8 \ 10 \\
 \left[\begin{array}{ccccc} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array} \right] \begin{array}{l} 1,2,3 \\ 1,6 \\ 3,4 \\ 4,5 \\ 5,6 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 2 \ 6 \ 10 \\
 \left[\begin{array}{cccc} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] \begin{array}{l} 1,2,3 \\ 1,6 \\ 3,4,5 \\ 5,6 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 3 \ 8 \ 10 \\
 \left[\begin{array}{ccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right] \begin{array}{l} 1, \\ 2 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 4 \ 7 \ 9 \ 10 \\
 \left[\begin{array}{ccccc} 0 & 1 & 0 & 1 & 1 \\ 0 & - & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right] \begin{array}{l} 1,2,8,9 \\ 1,4 \\ 1,7,9 \\ 2,6 \\ 3,4,5 \\ 3,8,9 \\ 5,7 \end{array} . \\
 \end{array}$$

При применении описанного подхода к решению задачи противогоночного кодирования состояний асинхронного автомата для снижения интенсивности переключений элементов памяти можно использовать следующие соображения. Каждому i -му столбцу матрицы кодирования можно поставить в соответствие множество переходов. Это множество составляют переходы, которыми связаны состояния автомата. В кодах состояний переменная z_i имеет различные значения, т. е. при таких переходах i -й триггер в реальной схеме, реализующей заданный автомат, меняет свое состояние. Очевидно, интенсивность переключений элементов памяти определяется вероятностями переходов между состояниями. Столбцу матрицы кодирования состояний ставится в соответствие вероятность события. Это событие заключается в том, что происходит некоторый переход, при котором меняется значение переменной, соответствующей данному столбцу. Поскольку переходы между состояниями автомата являются несовместимыми событиями, эта вероятность равна сумме вероятностей отдельных переходов из множества, соответствующего i -му столбцу.

Каждому совместимому множеству строк матрицы условий и, соответственно, вектору, имплицирующему все строки из данного множества, приписывается вес в виде величины, пропорциональной сумме вероятностей переходов, связанных с этим вектором. Искомое решение получается в виде покрытия множества строк матрицы условий максимальными совместимыми множествами, обладающего минимальным весом. Весом покрытия является сумма весов принадлежащих ему элементов.

Вычисление вероятностей переходов. Для подсчета вероятностей переходов между состояниями последовательного автомата принимаются следующие предположения: автомат является полностью определенным; все состояния являются взаимно достижимыми, т. е. для любых

двух состояний существует последовательность входных сигналов, переводящая автомат из одного состояния в другое; автомат работает достаточно долго; время пребывания автомата в устойчивом состоянии для всех состояний одно и то же.

Вероятность перехода последовательного автомата из состояния q_i в состояние q_j , вызываемого входным сигналом $\mathbf{x} = (x_1, x_2, \dots, x_n)$, равна вероятности прихода входного сигнала \mathbf{x} . Если имеется несколько входных сигналов, переводящих автомат из состояния q_i в состояние q_j , условная вероятность p'_{ij} такого перехода равна сумме вероятностей этих сигналов как вероятность несовместимых событий. Условием вероятности является то, что автомат находится в состоянии q_i . Абсолютная вероятность p_{ij} перехода из состояния q_i в состояние q_j в течение всего времени работы автомата равна произведению $p(q_i)p'_{ij}$, где $p(q_i)$ – вероятность того, что автомат находится в состоянии q_i (это событие и приход сигналов, меняющих состояние q_i на q_j , являются независимыми событиями).

Для вычисления вероятностей $p(q_i)$, $i = 1, 2, \dots, m$, где m – число состояний автомата, можно использовать уравнения Чэпмена – Колмогорова для дискретных Марковских цепей [24]. Подобно закону Кирхгоффа из электротехники, можно сказать, что сумма вероятностей переходов в некоторое состояние равна сумме вероятностей переходов из этого состояния. На основании приведенных соображений можно вывести следующую систему уравнений с неизвестными $p(q_i)$ ($i = 1, 2, \dots, m$):

$$\sum_{i=1}^m p(q_i)p'_{ij} = p(q_j), \quad j = 1, 2, \dots, m,$$
$$\sum_{i=1}^m p(q_i) = 1.$$

Вероятности p'_{ij} должны быть известны. Таким образом, решив эту систему уравнений, получим вероятности $p(q_i)$. Как было сказано раньше, абсолютная вероятность p_{ij} определяется как $p_{ij} = p(q_i)p'_{ij}$.

Для компонентного автомата в сети входной сигнал, кроме внешних сигналов сети, могут составлять сигналы о состояниях других компонентных автоматов. Благодаря внутренним связям то, что компонентный автомат находится в определенном состоянии, и приход некоторого входного для него сигнала нельзя считать независимыми событиями. В случае когда асинхронный параллельный автомат допускает моделирование его последовательным автоматом, предлагается для вычисления вероятностей переходов между состояниями компонентных автоматов использовать такое моделирование. Состояниями моделирующего асинхронного последовательного автомата являются устойчивые глобальные состояния заданного параллельного автомата. Вероятности переходов между глобальными состояниями определяются так, как показано выше для моделирующего последовательного автомата, а вероятности переходов между частичными состояниями и, соответственно, между состояниями компонентных автоматов определяются через вероятности переходов между глобальными состояниями. Вероятность перехода между частичными состояниями q_i и q_j равна сумме вероятностей переходов между теми глобальными состояниями P_s и P_t , для которых $q_i \in P_s$ и $q_j \in P_t$ или $q_i \in P_t$ и $q_j \in P_s$. Считается, что вероятность нахождения автомата в неустойчивом состоянии пренебрежимо мала и приравнивается нулю.

Таблица переходов между устойчивыми глобальными состояниями параллельного автомата из рассматриваемого примера представлена в виде табл. 3, где строки и столбцы соответствуют глобальным состояниям, и на пересечении строки и столбца стоит условие перехода из состояния, соответствующего строке, в состояние, соответствующее столбцу. Здесь и далее пустая клетка в таблице означает, что соответствующего перехода нет. По этой таблице легко определяются условные вероятности переходов, приведенные в табл. 4. Считается, что входные двоичные сигналы x_1 и x_2 независимы и равновероятны.

Таблица 3

Table 3

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	\bar{x}_2	x_2					
10		x_2	$\bar{x}_1 \bar{x}_2$	$x_1 \bar{x}_2$			
2.3.9		$x_1 x_2$	\bar{x}_1		$x_1 \bar{x}_2$		
3.4.5.6				$x_1 \bar{x}_2$	$\bar{x}_1 \bar{x}_2$	$x_1 x_2$	$\bar{x}_1 x_2$
3.5.6.9		$x_1 x_2$			\bar{x}_2		$\bar{x}_1 x_2$
6.7.8	$x_1 \bar{x}_2$					x_2	$\bar{x}_1 \bar{x}_2$
6.8.9	$x_1 \bar{x}_2$	$x_1 x_2$					\bar{x}_1

Таблица 4

Table 4

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	1/2	1/2					
10		1/2	1/4	1/4			
2.3.9		1/4	1/2		1/4		
3.4.5.6				1/4	1/4	1/4	1/4
3.5.6.9		1/4			1/2		1/4
6.7.8	1/4					1/2	1/4
6.8.9	1/4	1/4					1/2

Для вероятностей попадания рассматриваемого автомата в глобальные состояния $P_1 = \{1\}$, $P_2 = \{10\}$, $P_3 = \{2, 3, 9\}$, $P_4 = \{3, 4, 5, 6\}$, $P_5 = \{3, 5, 6, 9\}$, $P_6 = \{6, 7, 8\}$ и $P_7 = \{6, 8, 9\}$ по табл. 4 получим следующую систему уравнений:

$$\begin{aligned} p(1) &= 1/2 p(1) + 1/4 p(6.7.8) + 1/4 p(6.8.9); \\ p(10) &= 1/2 p(1) + 1/2 p(10) + 1/4 p(2.3.9) + 1/4 p(3.5.6.9) + 1/4 p(6.8.9); \\ p(2.3.9) &= 1/4 p(10) + 1/2 p(2.3.9); \\ p(3.4.5.6) &= 1/4 p(10) + 1/4 p(3.4.5.6); \\ p(3.5.6.9) &= 1/4 p(2.3.9) + 1/2 p(3.5.6.9) + 1/4 p(3.4.5.6); \\ p(6.7.8) &= 1/4 p(3.4.5.6) + 1/2 p(6.7.8); \\ p(6.8.9) &= 1/4 p(3.4.5.6) + 1/4 p(3.5.6.9) + 1/4 p(6.7.8) + 1/2 p(6.8.9); \\ p(1) + p(10) + p(2.3.9) + p(3.4.5.6) + p(3.5.6.9) + p(6.7.8) + p(6.8.9) &= 1, \end{aligned}$$

решением которой являются вероятности $p(1) = 15/153$, $p(10) = 48/153$, $p(2.3.9) = 24/153$, $p(3.4.5.6) = 16/153$, $p(3.5.6.9) = 20/153$, $p(6.7.8) = 8/153$, $p(6.8.9) = 22/153$ (удобно рассматривать эти дроби при одном и том же знаменателе).

Абсолютные вероятности переходов между глобальными состояниями, полученные по формуле $p_{ij} = p(q_i) p'_{ij}$, представлены в табл. 5.

Таблица 5

Table 5

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	15/306	15/306					
10		24/153	12/153	12/153			
2.3.9		6/153	12/153		6/153		
3.4.5.6				4/153	4/153	4/153	4/153
3.5.6.9		5/153			10/153		5/153
6.7.8	2/153					4/153	2/153
6.8.9	11/306	11/306					11/153

Как было сказано выше, вероятности переходов между частичными состояниями определяются через вероятности переходов между глобальными состояниями. Например, в задании ав-

томата частичное состояние 3 меняется на частичное состояние 8 (переход τ_4). Это происходит при переходах из глобального состояния $\{3, 4, 5, 6\}$ в глобальное состояние $\{6, 7, 8\}$, из того же состояния в глобальное состояние $\{6, 8, 9\}$ и из $\{3, 5, 6, 9\}$ в $\{6, 8, 9\}$. Суммой вероятностей таких несовместимых событий согласно табл. 5 является $13/153$, это вероятность перехода из частичного состояния 3 в частичное состояние 8. Табл. 6 представляет вероятности переходов между частичными состояниями (не важно, в каком направлении), где удалены пустые строки и столбцы, а дроби приведены к общему знаменателю.

Таблица 6
Table 6

	5	6	7	8	9	10
1		15/306	4/306	15/306	11/306	15/306
2	12/306	12/306				36/306
3			16/306	26/306		74/306
4			8/306		16/306	24/306
5				26/306		34/306
6						45/306
7					4/306	
8						11/306
9						45/306

Вероятности переходов между частичными состояниями параллельного автомата совпадают с вероятностями переходов между соответствующими состояниями компонентных автоматов в реализующей его сети.

Кодирование состояний компонентных автоматов и получение функций возбуждения элементов памяти. Ранее отмечалось, что столбцу z_i матрицы кодирования можно приписать вес в виде величины, пропорциональной сумме вероятностей переходов, при которых меняется значение z_i . В качестве этой величины возьмем числитель данной суммы при общем знаменателе всех рассматриваемых дробей. Таким же образом припишем веса максимальным совместимым множествам.

В настоящей работе не рассматривается какой-либо конкретный метод получения минимального взвешенного покрытия, так как эта задача подробно исследована (см., например, [25]). Для автоматов A_1 , A_2 и A_3 задача взвешенного кратчайшего покрытия строк матрицы условий максимальными совместимыми множествами решается просто: обязательным элементом покрытия для A_1 является множество $\{1, 2, 3\}$, для A_2 обязательными элементами покрытия являются $\{1, 2, 3\}$ и $\{3, 4, 5\}$, а у A_3 такое покрытие совпадает с совокупностью одноэлементных максимальных совместимых множеств. Для автомата A_1 искомое покрытие составляют множества $\{1, 2, 3\}$, $\{4, 5\}$, $\{5, 6\}$, для A_2 – множества $\{1, 2, 3\}$, $\{3, 4, 5\}$, $\{5, 6\}$ и для A_3 – $\{1\}$, $\{2\}$. Таблицей покрытия для автомата A_4 является табл. 7. Ее кратчайшим покрытием с минимальным весом составляют строки, соответствующие множествам $\{1, 2, 8, 9\}$, $\{2, 6\}$, $\{3, 4, 5\}$ и $\{5, 7\}$. Полученные покрытия определяют следующие матрицы кодирования состояний:

$$\begin{array}{l} \begin{array}{ccc} z_1^1 & z_2^1 & z_3^1 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 5 \\ 8 \\ 10 \end{array} \end{array} , \quad \begin{array}{ccc} z_1^2 & z_2^2 & z_3^2 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 6 \\ 10 \end{array} \end{array} , \quad \begin{array}{cc} z_1^3 & z_2^3 \\ \left[\begin{array}{cc} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{array} \right] \begin{array}{l} 1 \\ 3 \\ 8 \\ 10 \end{array} \end{array} , \quad \begin{array}{cccc} z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array} \right] \begin{array}{l} 1 \\ 4 \\ 7 \\ 9 \\ 10 \end{array} \end{array} \end{array}.$$

Качество кодирования состояний автомата можно оценить введенной в работе [11] суммой $D = \sum p_{ij}(d_{ij} - 1)$, где p_{ij} – вероятность перехода между состояниями q_i и q_j в любом направлении, d_{ij} – расстояние по Хэммингу между кодами состояний q_i и q_j , а суммирование ведется по всем парам состояний автомата. Очевидно, что чем эта сумма меньше, тем лучше решение, и $D = 0$, если любой переход между состояниями соответствует переключению только одного элемента

памяти в логической схеме, реализующей заданный автомат. Что касается автоматов A_1, A_2 и A_3 , то покрытия с минимальным весом для них либо совпадают с покрытиями, получаемыми без учета весов, либо мало отличаются от них. Для автомата A_4 $D = 86/306$. Кратчайшее покрытие, полученное по табл. 7 без учета весов, составляют множества $\{1, 2, 8, 9\}, \{1, 7, 9\}, \{2, 6\}$ и $\{3, 4, 5\}$. Соответственно, $D = 97/306$. Ясно, что предлагаемый метод снижает интенсивность переключения элементов памяти и вместе с тем потребление энергии в проектируемом устройстве.

Таблица 7

Table 7

	1	2	3	4	5	6	7	8	9	Вес Weight
1,2,8,9	1	1						1	1	38
1,4	1			1						30
1,7,9	1						1		1	70
2,6		1				1				84
3,4,5			1	1	1					84
3,8,9			1					1	1	92
5,7					1		1			48

Функции возбуждения элементов памяти компонентных автоматов получаются по табл. 2 заменой символов состояний их кодами. При этом функции определяются на интервалах пространства внутренних переменных, определяемых переходами, что необходимо для исключения опасных состязаний в асинхронных автоматах. Система полученных слабо определенных булевых функций возбуждения элементов памяти представлена в табл. 8, где функции заданы в интервальной форме.

Таблица 8

Table 8

	$x_1 x_2$	$z_1^1 z_2^1 z_3^1$	$z_1^2 z_2^2 z_3^2$	$z_1^3 z_2^3$	$z_1^4 z_2^4 z_3^4 z_4^4$	$z_1^{1'} z_2^{1'} z_3^{1'}$	$z_1^{2'} z_2^{2'} z_3^{2'}$	$z_1^{3'} z_2^{3'}$	$z_1^{4'} z_2^{4'} z_3^{4'} z_4^{4'}$
1	- 0	0 0 0	0 0 0	0 0	0 0 0 0	0 0 0	0 0 0	0 0	0 0 0 0
2	- 1	- 0 0	- 0 0	- 0	- - 0 0	1 0 0	1 0 0	1 0	1 1 0 0
3	- 1	1 0 0	1 0 0	1 0	1 1 0 0	1 0 0	1 0 0	1 0	1 1 0 0
4	0 0	1 - 0	1 - 0	1 -	1 - - 0	1 1 0	1 1 0	1 1	1 0 1 0
5	1 0	1 - -	1 - -	1 -	1 1 - -	1 1 1	1 1 1	1 1	1 1 1 1
6	0 -	1 1 0	1 1 0	1 1	1 0 1 0	1 1 0	1 1 0	1 1	1 0 1 0
7	1 0	1 1 -	1 1 -	1 1	1 0 1 0	1 1 1	1 1 1	1 1	1 0 1 0
8	1 1	1 - 0	1 - 0	1 -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
9	0 0	1 1 1	1 1 1	1 1	1 - 1 -	1 1 1	1 1 1	1 1	1 0 1 0
10	0 1	- 1 1	1 1 1	- 1	1 - 1 -	0 1 1	1 1 1	0 1	1 0 1 0
11	1 1	- 1 1	1 1 1	- 1	- - 1 1	0 1 1	1 1 1	0 1	0 0 1 1
12	1 0	1 1 1	1 1 1	1 1	1 1 1 1	1 1 1	1 1 1	1 1	1 1 1 1
13	- 0	1 1 1	1 1 1	1 1	1 0 1 0	1 1 1	1 1 1	1 1	1 0 1 0
14	0 1	- 1 1	1 1 1	- 1	1 0 1 0	0 1 1	1 1 1	0 1	1 0 1 0
15	1 1	1 - -	1 - -	1 -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
16	- 1	0 1 1	1 1 1	0 1	0 0 1 1	0 1 1	1 1 1	0 1	0 0 1 1
17	1 0	0 - -	- - -	0 -	0 0 1 1	0 0 0	0 0 0	0 0	0 0 0 0
18	0 0	0 1 1	1 1 1	0 1	- 0 1 -	0 1 1	1 1 1	0 1	1 0 1 0
19	0 -	0 1 1	1 1 1	0 1	1 0 1 0	0 1 1	1 1 1	0 1	1 0 1 0
20	1 1	- - -	1 - -	- -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
21	1 0	0 - -	- - -	0 -	- 0 - 0	0 0 0	0 0 0	0 0	0 0 0 0

Минимизация функций и установление межкомпонентных связей. Задача совместной минимизации функций возбуждения элементов памяти решается отдельно для каждой компоненты сети. На первом этапе производится максимальное расширение интервалов, на каждом из которых хотя бы одна функция имеет значение 1. Данная процедура выполняется с помощью построения матриц различий для каждого из интервалов и нахождения кратчайших столбцовых покрытий этих матриц, как описано в работе [3]. В результате получается система дизъюнктив-

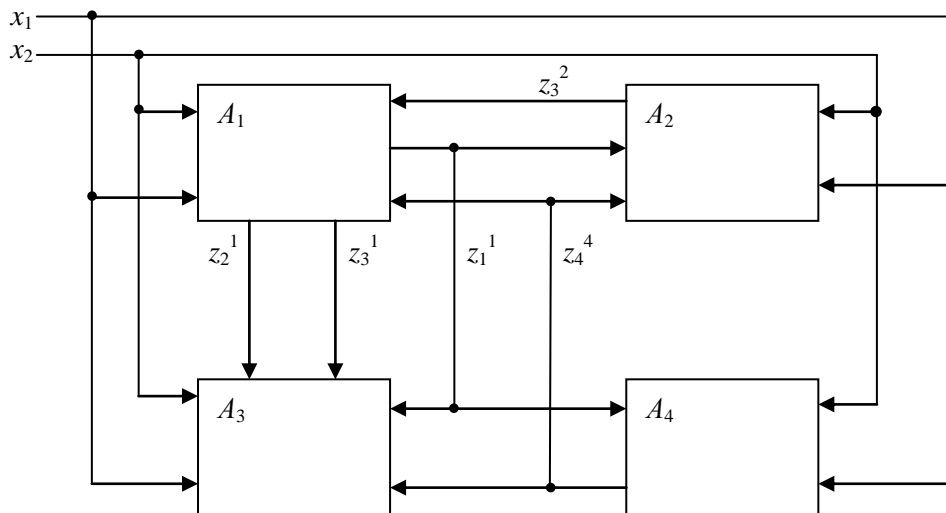
ных нормальных форм (ДНФ). При этом некоторые переменные оказываются несущественными аргументами и выбор среди них дает возможность минимизировать межкомпонентные связи, выбирая направление расширения интервала. В процессе минимизации ДНФ нельзя использовать обобщенное склеивание, так как в решение могут войти интервалы, при которых не устраняются опасные состязания. Поэтому минимизация заключается здесь в выявлении и удалении избыточных элементарных конъюнкций из полученных ДНФ. Эти конъюнкции выявляются с помощью способа, описанного в работах [18, 26], где для получения кратчайшего покрытия совокупности интервалов используется простая операция пересечения множеств.

В результате минимизации получаем следующие пары матриц, которые представляют системы ДНФ, реализующие соответственно автоматы A_1, A_2, A_3 и A_4 :

$$\begin{matrix} \begin{matrix} x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & 0 & - & - & - \\ 0 & 0 & 1 & - & 0 & - & - \\ 1 & 0 & 1 & - & - & - & - \\ 0 & - & - & 1 & - & 0 & - \\ - & 0 & 1 & - & 1 & - & - \\ - & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 \\ 0 & - & 0 & - & - & - & - \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & , & \begin{matrix} z_1^1 & z_2^1 & z_3^1 \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix} \end{matrix} & ; & \begin{matrix} x_1 & x_2 & z_1^1 & z_1^2 & z_2^2 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & - & 0 & - & - \\ 0 & 0 & - & 1 & - & 0 & - \\ 1 & 0 & 1 & - & - & - & - \\ 0 & - & - & - & - & 1 & - \\ - & 1 & - & - & - & - & 1 \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & , & \begin{matrix} z_1^2 & z_2^2 & z_3^1 \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix} \end{matrix} \end{matrix}$$

$$\begin{matrix} \begin{matrix} x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & 0 & - & - & - \\ - & 0 & 1 & - & - & - & - \\ 0 & - & - & - & 0 & 1 & - \\ 1 & 1 & - & - & 0 & - & - \\ 0 & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 \\ 0 & - & 0 & - & - & 1 & - \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & , & \begin{matrix} z_1^3 & z_2^3 \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix} \end{matrix} & ; & \begin{matrix} x_1 & x_2 & z_1^1 & z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \begin{bmatrix} - & 1 & - & - & - & 0 & 0 \\ 0 & 0 & - & 1 & - & - & - \\ 1 & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & 1 & - \\ - & 0 & 1 & - & 0 & - & - \\ 1 & 1 & - & - & - & - & 0 \\ 1 & 1 & - & - & - & - & 1 \\ - & 1 & - & 0 & - & - & 1 \end{bmatrix} & , & \begin{matrix} z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix} \end{matrix} \end{matrix}$$

Существенные аргументы функций показывают связи между компонентами сети. Полученная сеть изображена на рисунке.



Сеть последовательных автоматов
Net of sequential automata

Заключение. Описанный метод декомпозиции параллельного автомата в сеть последовательных автоматов может быть применен при синтезе распределенных систем управления. В таких системах, управляющих некоторой совокупностью отдаленных друг от друга объектов, все блоки связаны информационно в единую сеть, а каждый блок находится непосредственно на объекте управления. Здесь же возникает задача распределения выходных сигналов по компонентам сети. В данной статье выходные сигналы не рассматривались, все внимание было сосредоточено на функции возбуждения элементов памяти. Предлагаемый метод энергосберегающего кодирования частичных состояний параллельного автомата рассчитан на использование его в автоматизированной системе логического проектирования, когда заданный параллельный автомат допускает моделирование последовательным асинхронным автоматом. Сравнение результатов применения изложенного метода с результатами кодирования частичных состояний без учета интенсивности переключений элементов памяти показывает, что применение данного метода дает лучший результат. Использование декомпозиции параллельного автомата позволяет снизить размерность трудоемкой задачи кодирования состояний.

Список использованных источников

1. Закревский, А. Д. Параллельные алгоритмы логического управления / А. Д. Закревский. – М. : УРСС, 2003. – 304 с.
2. Питерсон, Дж. Теория сетей Петри и моделирование систем : пер. с англ. / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
3. Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.
4. Закревский, А. Д. Блочное кодирование частичных состояний у автоматов, реализующих параллельные алгоритмы логического управления / А. Д. Закревский // Изв. АН СССР. Техническая кибернетика. – 1983. – № 5. – С. 3–11.
5. Черемисинова, Л. Д. Реализация асинхронными автоматами параллельных алгоритмов логического управления / Л. Д. Черемисинова // Автоматика и вычислительная техника. – 1985. – № 2. – С. 65–69.
6. Поттосин, Ю. В. Декомпозиционный метод кодирования состояний параллельного автомата / Ю. В. Поттосин // Автоматика и вычислительная техника. – 1987. – № 1. – С. 84–91.
7. Мурога, С. Системное проектирование сверхбольших интегральных схем : в 2 кн. / С. Мурога. – М. : Мир, 1985. – Кн. 1. – 288 с.
8. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. – 1996. – Vol. 1. – P. 3–56.
9. Kashirova, L. State assignment of finite state machine for decrease of power dissipation / L. Kashirova, A. Keevallik, M. Meshkov // Second Intern. Conf. Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, 12–14 Nov. 1997. – Minsk : Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 1997. – Vol. 1. – P. 60–67.
10. Sudnitson, A. Partition search for FSM low power synthesis / A. Sudnitson // Fourth Intern. Conf. Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, 14–16 Nov. 2001. – Minsk : Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 2001. – Vol. 1. – P. 44–49.
11. Закревский, А. Д. Алгоритмы энергосберегающего кодирования состояний автомата / А. Д. Закревский // Информатика. – 2011. – № 1(29). – С. 68–78.
12. Поттосин, Ю. В. Кодирование состояний дискретного автомата, ориентированное на уменьшение энергопотребления реализующей схемы / Ю. В. Поттосин // Прикладная дискретная математика. – 2011. – № 4(14). – С. 62–71.
13. Поттосин, Ю. В. Энергосберегающее противогоночное кодирование состояний асинхронного автомата / Ю. В. Поттосин // Информатика. – 2015. – № 2(46). – С. 94–101.
14. Pottosin, Yu. Race-free state assignment for low power asynchronous automaton / Yu. Pottosin // Further Improvements in the Boolean Domain / ed. B. Steinbach. – Cambridge Scholars Publishing, 2018. – P. 253–267.
15. Pottosin, Yu. V. Low power assignment of partial states of a parallel automaton / Yu. V. Pottosin // Прикладная дискретная математика. – 2022. – № 56. – С. 113–122.
16. Pottosin, Yu. Optimal state assignment of synchronous parallel automata / Yu. Pottosin // Design of Embedded Control Systems. – N. Y. : Springer, 2005. – P. 111–124.

17. Поттосин, Ю. В. Совместное энергосберегающее кодирование состояний последовательных автоматов сети, реализующей параллельный автомат / Ю. В. Поттосин // Информатика. – 2023. – Т. 20, № 1. – С. 75–90.
18. Поттосин, Ю. В. Комбинаторные задачи в логическом проектировании дискретных устройств / Ю. В. Поттосин. – Минск : Беларуская навука, 2021. – 175 с.
19. Piguët, C. Low-power and low-voltage CMOS digital design / C. Piguët // *Microelectronic Engineering*. – 1997. – No. 39. – P. 179–208.
20. Якубайтис, Э. А. Асинхронные логические автоматы / Э. А. Якубайтис. – Рига : Зинатне, 1966. – 380 с.
21. Hartmanis, J. Algebraic structure theory of sequential machines / J. Hartmanis, R.E. Stearns. – Englewood Cliffs, N. J. : Prentice Hall Inc., 1966. – 208 p.
22. Кээваллик, А. Э. Теорема декомпозиции конечных автоматов / А. Э. Кээваллик // *Автоматика и вычислительная техника*. – 1974. – № 1. – С. 17–24.
23. Ковалев, А. В. О нахождении отношения параллельности на множестве мест одного подкласса сетей Петри / А. В. Ковалев // *Вес. Акад. навук Беларускай ССР. Сер. фіз.-мат. навук*. – 1989. – № 2. – С. 106–110.
24. Macii, E. High-level power modeling, estimation and optimization / E. Macii, M. Pedram, F. Somenzi // *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*. – 1998. – Vol. 17, no. 11. – P. 1061–1079.
25. Закревский, А. Д. Оптимизация покрытий множеств / А. Д. Закревский // *Логический язык для представления алгоритмов синтеза релейных устройств*. – М. : Наука, 1966. – С. 136–148.
26. 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.

References

1. Zakrevskij A. D. Parallel'nye algoritmy logicheskogo upravleniya. *Parallel Algorithms for Logical Control*. Moscow, URSS, 2003, 304 p. (In Russ.).
2. Peterson J. *Petri Net Theory and the Modeling of Systems*; first edition. Prentice Hall, 1981, 290 p.
3. Zakrevskij A. D., Pottosin Yu. V., Cheremisinova L. D. Logicheskie osnovy proektirovaniya diskretnyh ustrojstv. *Logical Fundamentals of Discrete Devices Design*. Moscow, Fizmatlit, 2007, 592 p. (In Russ.).
4. Zakrevskij A. D. *Block partial state assignment of automata that implement parallel algorithms for logical control*. *Izvestija Akademii nauk SSSR. Tehnicheskaja kibernetika [Proceedings of the Academy of Sciences of the USSR. Engineering Cybernetics]*, 1983, no. 5, pp. 3–11 (In Russ.).
5. Cheremisinova L. D. *Implementation of parallel algorithms for logical control by asynchronous automata*. *Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering]*, 1985, no. 2, pp. 65–69 (In Russ.).
6. Pottosin Yu. V. *A decomposition method for state assignment of a parallel automaton*. *Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering]*, 1987, no. 1, pp. 84–91 (In Russ.).
7. Muroga C. *Sistemnoye proektirovanie sverhbol'shih integral'nyh shem : v dvuh knigah. System Design of Super Large Integrated Circuits : In 2 Issues*. Moscow, Mir, 1985, iss. 1, 288 p. (In Russ.).
8. Pedram M. Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems*, 1996, vol. 1, pp. 3–56.
9. Kashirova L., Keevallik A., Meshkov A. M. State assignment of finite state machine for decrease of power dissipation. *Second International Conference Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, 12–14 November 1997*. Minsk, Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 1997, vol. 1, pp. 60–67.
10. Sudnitson A. Partition search for FSM low power synthesis. *Fourth International Conference Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, 14–16 November 2001*. Minsk, Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 2001, vol. 1, pp. 44–49.
11. Zakrevskij A. D. *Algorithms for low power state assignment of an automaton*. *Informatika [Informatics]*, 2011, no. 1(29), pp. 68–78 (In Russ.).
12. Pottosin Yu. V. *State assignment of a discrete automaton to decrease power consumption of the implementing circuit*. *Prikladnaya diskretnaya matematika [Discrete Applied Mathematics]*, 2011, no. 4(14), pp. 62–71 (In Russ.).

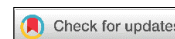
13. Pottosin Yu. V. *Low power race-free state assignment of an asynchronous automaton*. Informatika [Informatics], 2015, no. 2(46), pp. 94–101 (In Russ.).
14. Pottosin Yu. Race-free state assignment for low power asynchronous automaton. *Further Improvements in the Boolean Domain*. Ed. B. Steinbach. Cambridge Scholars Publishing, 2018, pp. 253–267.
15. Pottosin Yu. V. Low power assignment of partial states of a parallel automaton. *Prikladnaya diskretnaya matematika [Discrete Applied Mathematics]*, 2022, no. 56, pp. 113–122.
16. Pottosin Yu. Optimal state assignment of synchronous parallel automata. *Design of Embedded Control Systems*. New York, Springer, 2005, pp. 111–124.
17. Pottosin Yu. V. Joint low power state assignment of sequential automata that form a net implementing a parallel automaton. Informatika [Informatics], 2023, vol. 20, no. 1, pp. 75–90 (In Russ.).
18. Pottosin Yu. V. *Kombinatornye zadachi v logicheskom proektirovanii diskretnyh ustrojstv. Combinatorial Problems in Logical Design of Discrete Devices*. Minsk, Belaruskaya navuka, 2021, 175 p. (In Russ.).
19. Piguet C. Low-power and low-voltage CMOS digital design. *Microelectronic Engineering*, 1997, no. 39, pp. 179–208.
20. Yakubajtis E. A. *Asinhronnye logicheskie avtomaty. Asynchronous Logical Automata*. Riga, Zinatne, 1966, 380 p. (In Russ.).
21. Hartmanis J., Stearns R. E. *Algebraic Structure Theory of Sequential Machines*. Englewood Cliffs, New Jersey, Prentice Hall Inc., 1966, 208 p.
22. Keevallik A. *Decomposition theorem of finite automata*. Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering], 1974, no. 1, pp. 17–24 (In Russ.).
23. Kovalyov A. V. *About finding parallelism relation on the set of positions of a subclass of Petri nets*. Vesci Akademii nauk Belaruskaj SSR. Seria fizika-matematichnyh navuk [Proceedings of the Academy of Sciences of Byelorussian SSR. Series of Physico-Mathematical Sciences], 1989, no. 2, pp. 106–110 (In Russ.).
24. Macii E., Pedram M., Somenzi F. High-level power modeling, estimation and optimization. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 1998, vol. 17, no. 11, pp. 1061–1079.
25. Zakrevskij A. D. *Optimization of set covers*. Logicheskij yazyk dlya predstavlenia algoritmov sinteza relejnyh ustrojstv [Logical Language for Presenting Algorithms for Synthesis of Relay Devices], Moscow, Nauka, 1966, pp. 136–148 (In Russ.).
26. Pottosin Yu. 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.).

Информация об авторе

Поттосин Юрий Васильевич, кандидат физико-математических наук, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: pott@newman.bas-net.by

Information about the author

Yuri V. Pottosin, Ph. D. (Phys.-Math.), Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: pott@newman.bas-net.by



УДК 519.714.5
<https://doi.org/10.37661/1816-0301-2024-21-3-23-38>

Оригинальная статья
Original Article

Извлечение логических сетей при декомпиляции описаний КМОП-схем на уровне транзисторов

Д. И. Черемисинов, Л. Д. Черемисинова[✉]

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

Аннотация

Цели. Рассматривается проблема восстановления функционального описания цифровых устройств СБИС, представленных на транзисторном уровне. Целью исследования является разработка метода и программных средств выделения блоков, представляющих логические сети, из двухуровневых описаний КМОП-схем на транзисторном уровне, которые были получены в результате распознавания (экстракции) подсхем, реализующих логические элементы.

Методы. Предлагаются графовые методы и программные средства извлечения связанных блоков, представляющих логические сети, из двухуровневых описаний транзисторных схем в формате SPICE. В графовой интерпретации задача сводится к построению помеченного ориентированного графа логической сети, исходя из помеченного неориентированного двудольного графа, задающего двухуровневое описание транзисторной схемы.

Результаты. Предложенный метод позволяет выделить лексикографически ранжируемые логические сети, от которых производится переход к логическим уравнениям, задающим функции, реализуемые на выходных полюсах полученных сетей. Разработаны программные средства, которые обеспечивают генерацию иерархического описания в формате SPICE, реализующего исходную схему на транзисторном уровне, а также описания выделенных логических сетей на языке SF иерархических структурно-функциональных описаний дискретных устройств и на языках высокого уровня (VHDL и Verilog).

Заключение. Разработанные программные средства включены в программу декомпиляции транзисторных КМОП-схем и протестированы в ее составе на практических примерах схем транзисторного уровня. В работе приведены примеры обратного инжиниринга некоторых практических транзисторных схем.

Ключевые слова: экстракция транзисторных подсхем, КМОП-схемы, формат SPICE, распознавание логических вентилях, логическая сеть, обратный инжиниринг

Для цитирования. Черемисинов, Д. И. Извлечение логических сетей при декомпиляции описаний КМОП-схем на уровне транзисторов / Д. И. Черемисинов, Л. Д. Черемисинова // Информатика. – 2024. – Т. 21, № 3. – С. 23–38. <https://doi.org/10.37661/1816-0301-2024-21-3-23-38>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 23.07.2024
Подписана в печать | Accepted 20.08.2024
Опубликована | Published 30.09.2024

Extraction of logical networks during decompiling transistor-level CMOS circuit descriptions

Dmitry I. Cheremisinov, Ljudmila D. Cheremisinova[✉]

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

Abstract

Objectives. The problem of restoring the functional description of digital VLSI devices presented at the transistor level is considered. The objective of the work is to develop means for extraction of blocks representing logical networks from two-level descriptions of CMOS circuits at the transistor level, which were obtained as a result of recognition (extraction) of subcircuits that implement logic elements.

Methods. Graph based methods and software tools are proposed for extracting a connected blocks representing a logical network from two-level descriptions of a transistor circuits in SPICE format. In the graph interpretation, the task is reduced to constructing a labeled directed graph of a logical network based on a labeled undirected bipartite graph specifying a two-level description of the transistor circuit.

Results. The proposed method makes it possible to identify lexicographically ranked logical networks, from which a transition is made to logical equations that specify the functions implemented at the outputs of the resulting networks. Software tools have been developed that provide the generation of a hierarchical description in SPICE format that implements the original circuit at the transistor level, as well as descriptions of found logical networks in the SF language of hierarchical structural and functional descriptions of discrete devices and in high-level languages (VHDL and Verilog).

Conclusion. The developed methods are implemented in C++, included in the program for decompiling transistor CMOS circuits and tested within it on practical examples of transistor-level circuits. The paper provides examples of reverse engineering of some practical transistor circuits.

Keywords: transistor subcircuit extraction, CMOS circuits, SPICE format, logical gates recognition, logical network, reverse engineering

For citation. Cheremisinov D. I., Cheremisinova L. D. *Extraction of logical networks during decompiling transistor-level CMOS circuit descriptions*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 23–38 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-23-38>

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

Введение. Мощным инструментом проектирования и тестирования топологии современных СБИС, содержащих миллионы транзисторов, является обратное проектирование, или обратный инжиниринг, СБИС (англ. hardware reverse engineering). Обратное проектирование широко используется для верификации [1, 2] и перепроектирования [3] интегральных схем на новый технологический базис. Кроме того, в последние годы появились работы, в которых предлагается использовать его и для обнаружения несанкционированных вкладок. Такие вкладышки могут быть внесены в топологию устройства при его изготовлении [4–6].

В процессе обратного инжиниринга по плоскому (одноуровневому) структурному описанию интегральной схемы на транзисторном уровне восстанавливается ее описание на уровне логических элементов. Эта операция в смысле направления преобразований является инверсной задаче проектирования СБИС, и если перевод с уровня логических элементов на уровень транзисторов по аналогии с программированием можно считать компиляцией, то обратный процесс (извлечение описания уровня логических элементов) является декомпиляцией описания схемы на транзисторном уровне [7].

Структурный анализ цифровых схем путем их декомпиляции изучается в течение длительного времени. Обзор результатов решения этой задачи можно найти в работах [7–10]. Рассматривается наиболее распространенный стиль логики – КМОП – логических комплементарных МОП-структур (металл – оксид – полупроводник). Исходным объектом при декомпиляции является плоское описание схемы исследуемой СБИС. Для этой схемы может быть известна биб-

лиотека логических элементов, использованная при компиляции топологии СБИС, т. е. набор транзисторных подсхем, представляющих логические элементы. В этом случае декомпиляция состоит в распознавании (и замене) в описании декомпилируемой схемы библиотечных подсхем либо на функциональном уровне путем сравнения реализуемых подсхемами функций, либо на структурном уровне путем анализа изоморфизма соответствующих графов соединений транзисторов [11].

Библиотека логических элементов может быть и неизвестна, тогда в процессе декомпиляции выделяются часто встречающиеся фрагменты декомпилируемой схемы, которые выглядят как логические элементы. Те подсхемы, для которых распознаны реализуемые ими функции, являются логическими элементами, остальные относятся к классу псевдоэлементов [12]. Выделенные фрагменты составляют библиотеку элементов для анализируемой транзисторной схемы. В результате декомпиляции [7, 11, 12] плоское описание транзисторной схемы преобразуется в двухуровневое описание путем формирования уровня подсхем библиотечных элементов и генерации связей между экземплярами этих элементов в декомпилируемой схеме.

В настоящей работе исследуется полученное в результате декомпиляции СБИС двухуровневое описание транзисторной КМОП-схемы, представленной в формате SPICE (Simulation Program with Integrated Circuit Emphasis). Решается задача выделения блоков, представляющих логические сети и, соответственно, состоящих только из логических элементов. В процессе построения логической сети исходная транзисторная схема, в которой связи элементов рассматриваются как двунаправленные, преобразуется в логическую схему с направленными связями. Для схемы находятся входные и выходные порты и вычисляются функции, реализуемые схемой.

В графовой интерпретации задача извлечения логической сети сводится к построению помеченного ориентированного графа сети исходя из помеченного неориентированного графа, задающего двухуровневое описание транзисторной схемы. В общем случае могут быть выделены несколько логических сетей и могут оставаться элементы (ими будут, например, псевдоэлементы), не вошедшие ни в одну из сетей. После экстракции логических блоков строится иерархическое SPICE-описание, в которое эти блоки входят как элементы (наряду с оставшимися после экстракции элементами).

Для полученной логической сети (сетей) генерируется описание на языке SF иерархических структурно-функциональных описаний дискретных устройств [13], который является внутренним языком программных средств проектирования компонентов СБИС [14] в Объединенном институте проблем информатики НАН Беларуси. Возможна трансляция полученного SF-описания и на другие языки высокого уровня (VHDL и Verilog).

Разработанные методы реализованы на языке C++, включены в программу декомпиляции транзисторных КМОП-схем и протестированы в ее составе на практических примерах схем транзисторного уровня. Приведены примеры обратного инжиниринга некоторых практических транзисторных схем.

Постановка задачи декомпиляции транзисторных схем. Исходным объектом при декомпиляции является плоское описание схемы, которая состоит из полевых р-МОП- и n-МОП-транзисторов. МОП-транзистор имеет четыре вывода: сток (drain), затвор (gate), исток (source) и подложку (substrate). Доминирующим стилем логики при разработке современных цифровых СБИС является КМОП-структура [15], которая состоит из МОП-транзисторов двух упомянутых типов.

Элемент на основе стандартной (комплементарной) КМОП-логики включает две подсхемы, которые состоят из одинакового числа соответственно р-МОП- и n-МОП-транзисторов и реализуют взаимно инверсные булевы функции. Блок из р-МОП-транзисторов (Pull-Up network) размещен между шиной питания Vdd и выходом элемента, n-МОП-блок (Pull-Down network) – между шиной земли Gnd и выходом (рис. 1, а). Блоки обеспечивают связь выхода схемы элемента с источником питания Vdd, когда значение сигнала на выходе предполагается равным единице, или с землей Gnd в противном случае. Стандартные КМОП-схемы относятся к классу статических схем, в которых каждый выход в любой момент времени связан либо с источником питания Vdd, либо с шиной земли Gnd через тракт с малым сопротивлением.

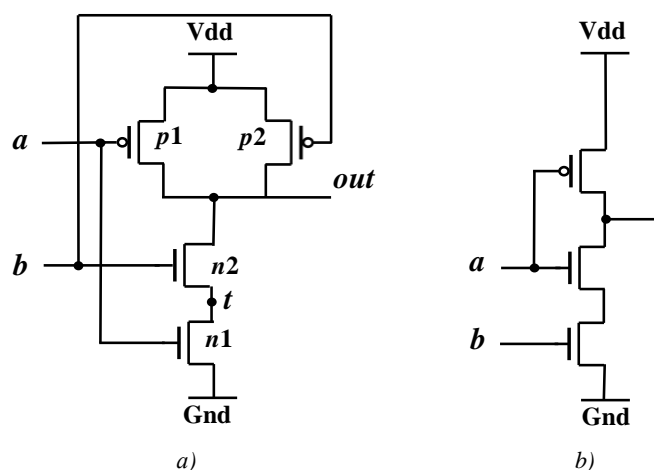


Рис. 1. Элементы КМОП-логики: а) вентиль И-НЕ на два входа; б) псевдоэлемент
 Fig. 1. Elements of CMOS transistor logic: a) CMOS NAND gate; b) pseudo element

Предполагается, что значение сигнала на выходе КМОП-элемента представляется булевой функцией от сигналов, поступающих на входы транзисторов соответствующей подсхемы (если игнорировать переходные эффекты во время переключения). Однако кроме таких (правильных) КМОП-вентилей при декомпиляции могут выделяться и псевдоэлементы, отличающиеся от КМОП-вентилей, например, тем, что р-МОП- и н-МОП-блоки имеют разную мощность или не реализуют взаимно инверсные булевы функции (рис. 1, б).

Наряду со стандартной КМОП-логикой при проектировании сложных регулярных структур интегральных схем широко используется логика на проходных транзисторах (pass transistor logic, PTL) [15], которая позволяет существенно сократить число транзисторов, необходимых для реализации логических функций, а также снизить энергопотребление схемы. В проходной логике сигналы с входных портов схемы допускается использовать для питания не только затворов транзисторов, но и выводов стока и истока н-МОП- и р-МОП-транзисторов. На эти выходы могут подаваться и внутренние сигналы схемы.

Однако транзисторы н-МОП-типа эффективно работают при передаче нулевого сигнала, но значительно понижают уровень сигнала логической единицы, а транзисторы р-МОП-типа, наоборот, эффективно работают при передаче сигнала логической единицы, но значительно понижают уровень сигнала логического нуля. В связи с этим транзисторы н-МОП- и р-МОП-типа используются, как правило, не по отдельности, а соединяются попарно, образуя передаточный вентиль. Такой вентиль состоит из пары транзисторов н-МОП и р-МОП-типа, которые связаны параллельно своими выводами стока и истока (рис. 2, а) [13]. Передаточные вентили могут использоваться в схемах как трехстабильные элементы, а также могут входить в состав схем сложных элементов, таких как мультиплексоры или вентили исключаящее ИЛИ, которые реализуются более просто на основе передаточной логики, чем на основе стандартной статической КМОП-логики (рис. 2, б, в).

Подсхемы, представляющие передаточные вентили, находятся на этапе обработки плоского описания декомпилируемой схемы, которая предваряет экстракцию из него описаний стандартных КМОП-вентилей. После нахождения передаточных вентилей распознаются подсхемы мультиплексоров и вентилей исключаящее ИЛИ [16].

На следующем этапе в транзисторной схеме распознаются подсхемы, реализующие статические КМОП-вентили. Структурный подход к декомпиляции КМОП-схем [12] позволяет разбить транзисторную схему на непересекающиеся подсхемы, представляющие группы транзисторов, связанных по постоянному току. Среди найденных групп транзисторов выделяются правильные подсхемы, представляющие собой статические КМОП-вентили, и определяются реализуемые ими функции. Каждая из подсхем, не распознанных как КМОП-вентиль, объявля-

ется псевдоэлементом, функциональное описание которого структурными методами в общем случае определить не удастся.

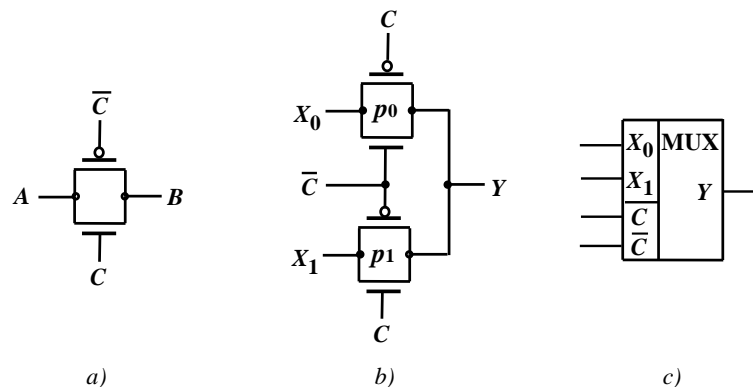


Рис. 2. Элементы проходной транзисторной логики: а) передаточный КМОП-вентиль; б) схемная реализация мультиплексора MUX 2×1 на основе передаточных вентиляей; в) схематическое представление мультиплексора

Fig. 2. Elements of pass transistor logic: а) CMOS transmission gate; б) multiplexer MUX 2×1 circuit implementation based on transmission gates; в) schematic representation of the multiplexer

Далее множества всех распознанных подсхем, представляющих выделенные элементы (КМОП-вентили, передаточные вентили, псевдоэлементы, мультиплексоры, вентили исключающее ИЛИ, трехстабильные инверторы), разбиваются на классы функционально и топологически эквивалентных подсхем [11]. Каноническое описание представителя каждого класса эквивалентных подсхем вентиляей представляет собой найденный библиотечный элемент. Им заменяются все экземпляры подсхемы этого класса в плоском описании декомпилируемой схемы, порождая двухуровневое описание декомпилируемой транзисторной схемы.

Задание транзисторных схем. Полученная при декомпиляции двухуровневая транзисторная схема задается в формате SPICE [1]. При распознавании подсхем, реализующих логические элементы, анализируется основная часть SPICE-описания транзисторной схемы Data Statements, которая описывает компоненты схемы и взаимосвязи между ними. Остальные части, если они присутствуют в исходном файле, игнорируются.

Раздел Data Statements описывает компоненты схемы в виде «моделей» элементов. В SPICE используются два основных типа моделей, которые и рассматриваются в настоящей работе: модели устройств (device models) и модели подсхем (subcircuit models). Модели устройств, по сути, представляют собой типы примитивных компонентов, таких как транзисторы, диоды и т. д. Модель подсхемы определяет схему непримитивного элемента и обычно (хотя и не обязательно) представляет собой схему. Модели других типов игнорируются. Модель подсхемы (.subckt в SPICE-описании) задается в виде блока в SPICE-файле и описывает структуру соединений элементов (примитивных и непримитивных) в этой подсхеме.

Тип модели используемого в SPICE-описании элемента задается первой буквой его имени. Имена примитивных элементов, являющихся МОП-транзисторами, должны начинаться с латинской буквы М. Имена непримитивных элементов, представляемых подсхемами, начинаются с латинской буквы Х. Следует заметить, что описания на языке SPICE нечувствительны к регистру символов, т. е., в частности, допустимо указывать и М, и m (Х и x).

Схема задается списком соединений ее элементов. В этом списке указаны связи между элементами, а точнее, между их выводами, которые осуществляются с помощью электрических цепей (nets). Каждая цепь представляет собой связанный набор выводов элементов, на которые подается один и тот же сигнал, и каждый вывод элемента подключен ровно к одной цепи. Для каждого элемента указываются связи всех его выводов путем задания имен связанных с ними цепей.

Описание транзистора начинается с его имени и содержит перечисление меток цепей, связанных с выводами стока, затвора, истока и подложки в заранее определенной последовательности. Общая форма описания связей униполярного транзистора в формате SPICE имеет вид

$$\langle \text{name} \rangle \langle \text{nd} \rangle \langle \text{ng} \rangle \langle \text{ns} \rangle \langle \text{nb} \rangle \langle \text{model-name} \rangle,$$

где name – имя транзистора в схеме (начиная с буквы M); nd, ng, ns и nb – идентификаторы цепей, связанных с выводами стока (drain), затвора (gate), истока (source) и подложки (substrate) соответственно; model-name – имя модели устройства (для *n*-МОП- и *p*-МОП-транзисторов это могут быть nmos и pmos).

Общая форма описания связей непримитивного элемента, модель которого представляется подсхемой с *n* выводами, в формате SPICE имеет вид

$$X\langle \text{name} \rangle \langle P1 \rangle \langle P2 \rangle \dots \langle Pn \rangle \langle \text{model-name} \rangle,$$

где name – имя элемента в схеме; P1, P2... Pn – идентификаторы цепей, связанных с выводами элемента; model-name – имя модели элемента.

В программе декомпиляции принято соглашение, что имена моделей элементов разного типа различаются первой буквой. Например, имена моделей логических элементов начинаются с буквы G, псевдоэлементов – с P, передаточных вентилях – с CN, мультиплексоров – с GM. Логические элементы являются (*n*, 1)-полюсниками, т. е. имеют один выходной полюс. По соглашению последний параметр в описании моделей всех элементов всегда соответствует выходному полюсу и именуется как Y.

Пример представления транзисторной схемы в виде иерархического описания в формате SPICE, полученного в результате декомпиляции плоского описания схемы fd, приведен в листинге 1. В этом описании представлены четыре модели непримитивных элементов, задающих вентиль 2И-НЕ, инвертор, передаточный элемент и мультиплексор. Их модели обозначены соответственно через G0, G1, CN2 и GMAB_0. Последний уровень иерархии схемы содержит девять экземпляров логических элементов: четыре вентиля 2И-НЕ, три инвертора и два мультиплексора MUX 2×1. Последний параметр в описании моделей соответствует выходному полюсу. Передаточные вентиля являются двунаправленными, поэтому описание их модели CN2 не имеет параметра, обозначаемого как Y. В описании используются четыре экземпляра элемента модели CN2, и все они входят в состав мультиплексоров.

Листинг 1. Иерархическое SPICE-описание транзисторной схемы fd

```
* SPICE deck for cell fd_gen
.GLOBAL vcc gnd
.SUBCKT G0 A B Y
* (A AND B)
M1 1 A gnd gnd nmos
M2 Y B 1 gnd nmos
M3 vcc A Y vcc pmos
M4 Y B vcc vcc pmos
.ENDS

.SUBCKT G1 A Y
* A
M1 Y A gnd gnd nmos
M2 Y A vcc vcc pmos
.ENDS

.SUBCKT CN2 A nC B C NB PB
M1 A nC B PB pmos
M2 A C B NB nmos
.ENDS

.SUBCKT GMAB_0 A B C nC Y
```

```
* (A AND C) OR (B AND nC)
XCN20 Y nC A C gnd vcc CN2
XCN21 B C Y nC gnd vcc CN2
.ENDS

.SUBCKT fd_gen r1 s1 c d q
XM0I1 r1 11 14 G0 Fets=nmos10+nmos9+pmos24+pmos25
XM0I2 s1 12 11 G0 Fets=nmos11+nmos12+pmos26+pmos27
XM0I3 8 r1 9 G0 Fets=nmos0+nmos1+pmos15+pmos16
XM0I4 s1 9 13 G0 Fets=nmos2+nmos3+pmos17+pmos18
XM1I1 11 qn G1 Fets=nmos13+pmos28
XM1I2 12 q G1 Fets=nmos14+pmos29
XM1I3 c 10 G1 Fets=nmos6+pmos21
XGMUAB_0 9 14 c 10 12 GMAB_0
XGMUAB_1 13 d c 10 8 GMAB_0
.ENDS
```

Графовая модель транзисторных схем и ее внутреннее представление. При декомпиляции удобной моделью схем на уровне транзисторов является помеченный неориентированный двудольный граф $G = (V_1, V_2, E)$, $V_1 \cap V_2 = \emptyset$. Этот граф задает структуру списка соединений элементов транзисторной схемы (транзисторов и подсхем из транзисторов). В нем вершины из первой доли V_1 графа соответствуют выводам экземпляров элементов (транзисторов или подсхем из транзисторов, представляющих элементы) и портам схемы (входам и выходам электрической схемы). Вершинам из V_2 ставятся в соответствие цепи, т. е. связи между выводами элементов. Примерами цепей являются цепи питания и земли, которые связаны с большим числом элементов схемы. Каждая из дуг $e \in E$ графа G связывает вершины из разных множеств V_1 и V_2 .

Вершины графа из множества V_2 помечаются именами цепей, а вершины из V_1 имеют метки, идентифицирующие выводы экземпляров элементов. Эти метки состоят из имени экземпляра элемента и имени вывода модели этого элемента. Двудольный граф $G = (V_1, V_2, E)$, являющийся моделью схемы в формате SPICE, разреженный, и степени всех вершин в доле V_1 выводов элементов равны единице.

При решении задачи декомпиляции граф G задается в виде массива N списков смежности вершин из множества V_1 . Такой массив обычно индексируется вершинами: каждой вершине из V_1 соответствует список ее соседей, в данном случае список связанных с ней цепей. Поскольку в графе G , задающем модель схемы в формате SPICE, степени всех вершин из множества V_1 равны единице, то списки соседей всех вершин такого графа являются одноэлементными и для задания графа G используется одномерный массив N .

Для того чтобы привязать задание графа к структуре транзисторной схемы, массив N смежности вершин графа G предлагается разбивать и, следовательно, индексировать на списки смежности вершин графа, соответствующих выводам отдельных элементов схемы (на начальном этапе декомпиляции элементами являются транзисторы). Для каждого элемента выделяется список связей его выводов, приведенных в том порядке, в котором они упоминаются в описании связей модели этого элемента в формате SPICE. Список связей i -го элемента схемы в массиве смежности вершин графа задается номером в N цепи, смежной вершине, которая соответствует первому выводу этого элемента (для транзистора это сток, для непримитивного элемента – вывод P1). Порядок следования списков цепей в массиве смежности определяется порядком следования элементов в SPICE-описании схемы.

Например, в табл. 1 приведен массив N смежности вершин из множества V_1 графа $G = (V_1, V_2, E)$, представляющего схему fd (листинг 1). Элементами этого массива являются вершины из V_2 , соответствующие цепям схемы. Индексами 0, 3, 6, 9, 12, 14, 16, 18 и 23 массив делится на девять списков, задающих связи выводов отдельных элементов. Для удобства восприятия массив показан не в виде вектора, а разбит на списки, каждый из которых соответствует выводам одного элемента и приведен в отдельной строке. Каждая из строчек табл. 1 задает связи выводов одного элемента схемы. Например, первая строка соответствует элементу M0I1

(в листинге `xM0I1`), тип которого определяется моделью `G0`, а связи задаются парами `M0I1.P1 = r1`, `M0I1.P2 = l1` и `M0I1.Y = l4`.

Таблица 1
Массив смежности вершин графа, соответствующих выводам элементов схемы

Table 1
Adjacency array of graph vertices corresponding to the circuit element outputs

Элементы схемы <i>Elements of the diagram</i>	Связи выводов элементов <i>Connections of element pins</i>
M0I1	r1 l1 l4
M0I2	s1 l2 l1
M0I3	8 r1 9
M0I4	s1 9 l3
M1I1	l1 qn
M1I2	l2 q
M1I3	c l0
GMUAB_0	9 l4 c l0 l2
GMUAB_1	l3 d c l0 8

Упомянутое задание графа $G = (V_1, V_2, E)$, представляющего транзисторную схему, строится на этапе обработки исходного двухуровневого описания в формате SPICE. Табл. 1 полностью определяет граф $G = (V_1, V_2, E)$ схемы. Однако для сокращения вычислительных затрат при извлечении связанных блоков, представляющих логические сети, из двухуровневых описаний транзисторных схем одновременно с построением списков смежности вершин из множества V_1 формируются также и списки вершин, смежных вершинам из множества V_2 . Другими словами, связи цепей описания схемы задаются также в явном виде. Для рассматриваемой схемы эти связи показаны во втором столбце табл. 2, где через точку приведены имена элемента и его вывода.

Таблица 2
Массив смежности вершин графа, соответствующих цепям схемы

Table 2
Adjacency array of graph vertices corresponding to the circuit nets

Цепи <i>Chains</i>	Связи <i>Connections</i>
r1	M0I1.A M0I3.B
s1	M0I2.A M0I4.A
c	M1I3.A GMUAB_0.C GMUAB_1.C
d	GMUAB_1.B
q	M1I2.Y
qn	M1I1.Y
l1	M0I1.B M0I2.Y M1I1.A
l4	M0I1.Y GMUAB_0.B
l2	M0I2.B M1I2.A GMUAB_0.Y
8	M0I3.A GMUAB_1.Y
9	M0I3.Y xM0I4.B GMUAB_0.A
l3	M0I4.Y GMUAB_1.A
l0	M1I3.A GMUAB_0.nC GMUAB_1.nC

Извлечение логической сети из двухуровневой транзисторной схемы. Логическая сеть дискретного устройства отражает его внутреннее строение с точностью до функций, реализуемых его элементами. В графовой интерпретации моделью логической сети является помеченный ориентированный граф $H = (W, A)$, где W и A – множества вершин и дуг графа. Множество W разбивается на три подмножества вершин, соответствующих входным и выходным портам сети и элементам. Каждая вершина из первых двух подмножеств помечена входной или выходной переменной сети. Вершины из третьего подмножеств помечены функциями, реализуемыми элементами сети.

Ориентированный граф $H = (W, A)$ логической сети строится исходя из неориентированного графа $G = (V_1, V_2, E)$, соответствующего объектной двухуровневой транзисторной схеме.

В множество W помимо вершин, соответствующих портам сети, входят вершины, соответствующие экземплярам логических элементов. Помимо таких элементов могут существовать (как результат декомпиляции) и элементы, которые не распознаны как логические. Например, они могут соответствовать псевдоэлементам. Наличие таких элементов приводит к тому, что может быть извлечена не логическая сеть, а несколько логических блоков. Эти блоки порождаются непересекающимися связными подграфами $H_i = (W_i, A_i)$ графа $G = (V_1, V_2, E)$.

Дуги $a \in A$ генерируемого графа $H = (W, A)$ логической сети порождаются цепями $x \in V_2$ графа G . Вершины $v \in W$ и $u \in W$ связываются дугой $a = (v, u)$, соответствующей цепи x , если эта цепь связывает выводы элементов, соответствующих вершинам v и u , причем для v цепь x указана в качестве выхода, а для u – в качестве одного из входов.

Выделение компонентов, представляющих логические сети, из двухуровневого описания транзисторной схемы включает решение следующих основных задач:

Задача 1. Поиск компонента связности графа $G = (V_1, V_2, E)$, описывающего логическую сеть $H^* = (W^*, A^*)$.

Задача 2. Определение входных и выходных портов извлеченной логической сети, описываемой графом $H^* = (W^*, A^*)$.

Задача 3. Генерация описания логической сети на языках высокого уровня.

Следует заметить, что, если декомпилируемая транзисторная схема получена путем автоматического синтеза с помощью какой-либо САПР с заданной библиотекой проектирования, результатом решения задачи 1 является одна логическая сеть, функционально эквивалентная исходной транзисторной схеме. Задача 2 представляет собой самостоятельный этап декомпиляции транзисторных схем, о которых ничего неизвестно, кроме описания структуры связей их транзисторов. Решение этой задачи нетривиально для сложных транзисторных схем, содержащих сотни миллионов транзисторов. Решение задачи 3 является завершающим этапом декомпиляции схемы, оно дает возможность использовать известные САПР для моделирования транзисторной схемы и перепроектирования ее на основе другого технологического базиса.

Задача 1. Поиск компонента связности графа $G = (V_1, V_2, E)$, описывающего логическую сеть. Поиск компонента связности графа $G = (V_1, V_2, E)$, описывающего логическую сеть $H^* = (W^*, A^*)$, осуществляется в процессе обхода графа G по входящим и исходящим путям, начиная от вершин из V_1 , которые помечены как выводы некоторого произвольно взятого экземпляра логического элемента, не вошедшего ни в одну из ранее найденных логических сетей. При встрече вершины, не помеченной как логический элемент, просмотр соответствующего пути обрывается. Обход графа с целью построения компоненты связности, представляющей логическую сеть, осуществляется известным методом поиска в ширину (BFS, от англ. breadth-first search). Такой алгоритм имеет линейную временную сложность.

Метод поиска позволяет не только найти компоненту связности графа G и соответствующий граф логической сети $H^* = (W^*, A^*)$, но и получить лексикографическое упорядочение вершин графа H^* , учитывающее достижимость вершин друг из друга, и, следовательно, ранжировать вершины графа по уровням.

Поиск в ширину начинается с некоторой начальной вершины v графа, соответствующей некоторому логическому элементу. Рассмотренные в процессе просмотра вершины отмечаются как просмотренные. Анализируемая вершина v_0 включается в множество W^* и помечается как

вершина уровня k . Затем в множество W^* вносятся все вершины, которые соответствуют логическим элементам и входят в полуокрестности захода Γ^+v_0 и исхода Γ^-v_0 . Вершины из Γ^+v_0 помечаются как вершины уровня $(k-1)$, а вершины из Γ^-v_0 – уровня $(k+1)$. После этого аналогично рассматриваются вершины из окрестностей всех вновь введенных в граф $H^* = (W^*, A^*)$ вершин и т. д. В общем случае все не помеченные ранее вершины из окрестностей уровня k помечаются как вершины уровня $(k-1)$ или $(k+1)$. Просмотр вершин прекращается, когда все вершины будут помечены как просмотренные.

Процедура построения графа $H^* = (W^*, A^*)$ логической сети ведется, исходя из массивов смежности вершин из множеств V_1 и V_2 графа $G = (V_1, V_2, E)$ (см. табл. 1 и 2).

Для примера начнем с вершины v_0 , соответствующей элементу M0I2 (см. табл. 1), который не входит ни в одну ранее построенную логическую сеть и ранее не рассматривался. Выводы этого элемента связаны с цепями s1, 12 и 11, причем полуокрестность захода $\Gamma^-v_0 = \{s1, 12\}$, а полуокрестность исхода $\Gamma^+v_0 = \{11\}$ (см. табл. 1). Вершину v_0 включим в множество W^* графа $H^* = (W^*, A^*)$ и отнесем ее к некоторому уровню k . Среди вершин из V_1 графа $G = (V_1, V_2, E)$, смежных вершине s1 $\in V_2$ (их две: M0I2.A и M0I4.A), ищем вершину, соответствующую выводу Y какого-либо логического элемента. Таких вершин нет, значит, вершина s1 является концевой. Среди вершин из V_1 , смежных вершине 12 (их три: M0I2.B, M1I2.A и GMUAB_0.Y), находится вершина (для правильно построенной схемы она всегда одна), удовлетворяющая этому условию: GMUAB_0.Y. Она соответствует выходному полюсу элемента GMUAB_0 и является драйвером второго входа элемента M0I2. Введем в сеть вершину для GMUAB_0.Y и отнесем ее к уровню $(k-1)$. Аналогично рассматривается единственная вершина 11 из Γ^+v_0 , но в этом случае берутся все смежные ей вершины, соответствующие выводам элементов, отличных от Y (их две: M0I1.B и M1I1.A). Вершины, соответствующие элементам M0I1 и M1I1, вносятся в множество W^* графа сети и относятся к уровню $(k+1)$. В множество A^* вносятся дуги (s1, M0I2), (GMUAB_0, M0I2), (M0I2, M0I1), (M0I2, M1I1).

Аналогичные действия повторяются для каждой из вновь введенных в граф вершин (GMUAB_0, M0I1, M1I1) уровней $(k-1)$ и $(k+1)$ до тех пор, пока они находятся. На рис. 3 показаны состояния графа логической сети после выполнения первой и второй итераций. Для каждой входящей дуги графов на рисунке приведено имя соответствующего вывода элемента (согласно описанию его модели на листинге 1).

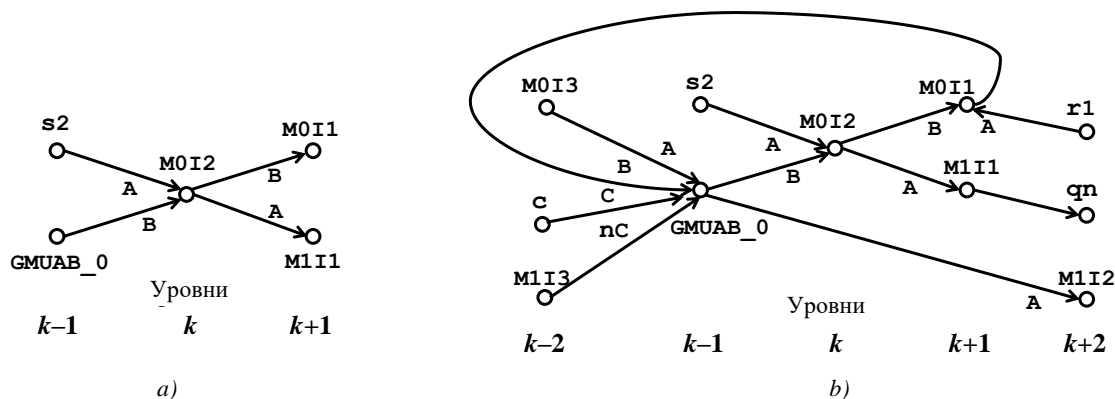


Рис. 3. Граф $H^* = (W^*, A^*)$ после выполнения первой итерации (a); второй итерации (b)

Fig. 3. Graph $H^* = (W^*, A^*)$ after the first iteration (a); the second iteration (b)

Элементы, входящие в иерархическое SPICE-описание fd, порождают единственный связный компонент графа $G = (V_1, V_2, E)$, порождающий граф $H^* = (W^*, A^*)$ (рис. 4). Вершин, не вошедших в этот компонент, не осталось, значит, построена единственная логическая сеть, реализующая анализируемую транзисторную схему.

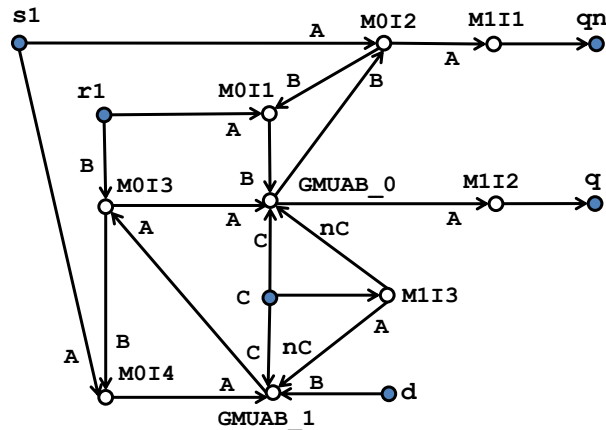


Рис. 4. Граф $H^* = (W^*, A^*)$ логической сети
Fig. 4. Graph $H^* = (W^*, A^*)$ of the logical network

Задача 2. Определение входных и выходных портов логической сети. Следующей задачей, связанной с выделением логической сети, является определение ее входных и выходных портов. Эта задача просто решается путем рассмотрения полуокрестностей исхода Γ^+v и захода Γ^-v для всех вершин v графа $H^* = (W^*, A^*)$.

Если для некоторой вершины $v \in W^*$ все вершины из ее окрестностей Γ^+v и Γ^-v помечены как элементы, то вершина v является внутренней. Вершины, не являющиеся внутренними, связаны с входами или выходами логической сети в зависимости от того, какое из множеств Γ^+v или Γ^-v содержит каждую из этих не внутренних вершин.

Входы полученной логической сети соответствуют вершинам $r1, s1, c$ и d , так как в графе $H^* = (W^*, A^*)$ полуокрестности захода этих вершин являются пустыми ($\Gamma^-v = \emptyset$). Выходы сети порождаются вершинами q и qn , так как полуокрестности исхода этих вершин $\Gamma^+v = \emptyset$. На рис. 4 вершины графа $H^* = (W^*, A^*)$, для которых полуокрестности исхода или захода являются пустыми, помечены затемненным цветом.

В листинге 2 приведен фрагмент SPICE-описания, которое было получено после выделения компонента C0, описывающего логическую сеть. Приведенный фрагмент есть результат преобразования раздела .SUBCKT fd_gen (листинг 1). Остальные разделы листингов 1 и 2 совпадают. Здесь параметры модели C0, имена которых начинаются с символа P (P0, P1, P2, P3), задают входные полюсы, а параметры, имена которых начинаются с символа O (O4, O5), – выходные полюсы сети.

Листинг 2. Фрагмент иерархического SPICE-описания транзисторной схемы fd, полученный после выделения логической сети

```
.SUBCKT C0 P0 P1 P2 P3 O4 O5
XG0M0I1 P2 2 5 G0
XG0M0I2 P3 3 2 G0
XG0M0I3 6 P2 7 G0
XG0M0I4 P3 7 4 G0
XG1M1I1 P0 1 G1
XG1M1I2 2 O5 G1
XG1M1I3 3 O4 G1
XGMAB_0GMUAB_0 7 5 P0 1 3 GMAB_0
XGMAB_0GMUAB_1 4 P1 P0 1 6 GMAB_0
.ENDS
.SUBCKT fd_gen_ier r1 s1 c d q qn
XC0 c d r1 s1 q qn C0
.ENDS
```

Логическая сеть, построенная по SPICE-описанию транзисторной схемы, приведенному в листинге 2, показана на рис. 5.

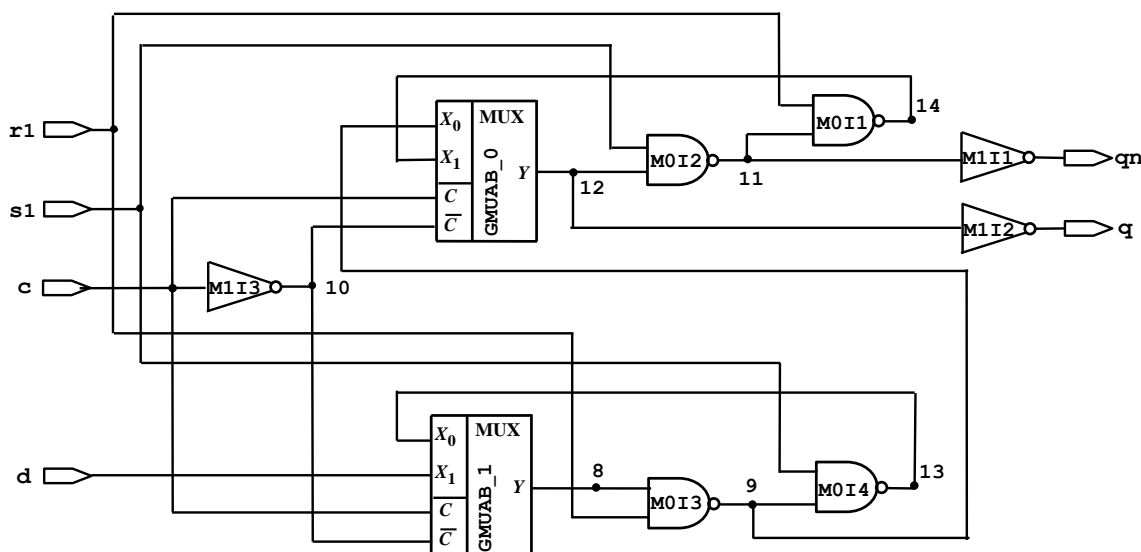


Рис. 5. Логическая сеть

Fig. 5. Logical network

Заметим, что задача определения входных и выходных портов логической сети имеет и самостоятельный интерес. Анализ сложных транзисторных схем, для которых известна только структура связей транзисторов, сильно затруднен без явного перечисления входных и выходных портов устройства. Нахождение внешних портов таких схем представляет собой трудоемкую задачу.

Для приводимого в статье демонстрационного примера список внешних портов (выделены жирным шрифтом) задан (хотя и без разделения на входы и выходы): `.SUBCKT fd_gen r1 s1 c d q` (листинг 1). В процессе декомпиляции после извлечения логической сети порты были разделены на входные и выходные. Однако оказалось, что приведенный в описании схемы список портов неполон: имеется еще один выходной порт – `qn`: `.SUBCKT fd_gen_ier r1 s1 c d q qn` и `XC0 c d r1 s1 q qn` (листинг 2).

Задача 3. Генерация описания логической сети на языках высокого уровня. Предлагаемый метод выделения логической сети ранжирует логические элементы по уровням. От такой сети несложно осуществить переход к логическим уравнениям, задающим функции, которые реализуются на ее внутренних и выходных полюсах. В работе реализован алгоритм построения представления логической сети в формате LOG [13] логических уравнений языка SF описания иерархических структурно-функциональных описаний дискретных устройств. Этот алгоритм исходит из иерархического SPICE-описания транзисторной схемы, полученного после выделения логической сети. Реализованы также программные средства перевода описаний на языке SF на другие языки проектирования, такие как VHDL и Verilog [17]. Для полученной логической сети LOG- и VHDL-описания приведены в листингах 3 и 4.

Листинг 3. Описание логической сети c0 в формате LOG языка SF

```
TITLE C0
FORMAT SF
AUTHOR extractor
DATE 18-04-2024
PROJECT fd_gen_ier
DCL_PIN
EXT
```



```
INP
P0 P1 P2 P3
OUT
O4 O5
INTER
END_PIN
FUNCTION
LOG
4 2 4
O4 = ^ s3 ;
O5 = ^ s2 ;
s2 = ^ (P3 * s3 );
s3 = (s7 * P0 ) + ((^ (P2 * s2 )) * s1 );
s1 = ^ P0 ;
s7 = ((^(^(P3 * s7 )) * P0 ) + (P1 * s1 )) * P2 );
END_LOG
END_FUNCTION
END_C0
```

Листинг 4. Описание логической сети C0 в формате языка VHDL

```
library IEEE;
use ieee.Std_Logic_1164.all;

entity C0 is
  port (
    P0 : in std_logic;
    P1 : in std_logic;
    P2 : in std_logic;
    P3 : in std_logic;
    O4 : out std_logic;
    O5 : out std_logic
  );
end entity C0;

architecture sf of C0 is
  signal s2: std_logic;
  signal s3: std_logic;
  signal s1: std_logic;
  signal s7: std_logic;

  begin
    O4<=NOT s3;
    O5<=NOT s2;
    s2<=NOT (P3 AND s3);
    s3<=((s7 AND P0) OR (NOT (P2 AND s2) AND s1));
    s1<=NOT P0;
    s7<=(NOT (NOT (P3 AND s7) AND P0) OR (P1 AND s1)) AND P2;

  end architecture sf;
```

Обсуждение практических результатов. Предложенный графовый метод выделения компонентов следующего уровня иерархии SPICE-описания, которые представляют собой логические сети, позволяет переходить от описания схемы на уровне транзисторов к описанию на уровне логических элементов. Как правило, результатом такого преобразования является одна логическая сеть, поведение которой на функциональном уровне эквивалентно поведению исходной транзисторной схемы. Это гарантируется, если схема на уровне транзисторов получена с помощью систем автоматизации проектирования на основе некоторой заданной библиотеки логических элементов.

Произвольные транзисторные схемы, которые синтезированы некоторым иным способом, могут содержать выделяемые при декомпиляции подсхемы псевдоэлементы, для которых не удастся определить функциональное описание. Соответственно, помимо выделенной логической сети в описании могут присутствовать и не вошедшие в нее элементы (в том числе туда могут попасть и КМОП-клапаны). Плоские описания таких схем на транзисторном уровне декомпилируются в смешанные иерархические описания на транзисторно-логическом уровне. В них могут быть также несколько компонентов, представленных логическими сетями, и могут присутствовать компоненты, описанные на транзисторном уровне. Сведения о декомпиляции некоторых таких SPICE-описаний приведены в табл. 3, где к логическим элементам отнесены КМОП-клапаны, мультиплексоры и КМОП-инверторы с тремя состояниями. Под остатком понимаются логические элементы, не вошедшие ни в одну логическую сеть. Время выполнения программы поиска логических сетей существенно зависит не только от сложности декомпилируемой схемы (измеряемой числом транзисторов), но от числа псевдоэлементов (которые могут увеличивать число порождаемых логических сетей) и числа передаточных элементов.

Таблица 3
Результаты выделения логических сетей

Table 3
Results of extracting logical networks

	Схемы Circuits					
	1	2	3	4	5	6
Число:						
транзисторов;	243 806	11 935	39 424	7884	11 436	247
логических элементов;	50 524	2777	4737	378	952	41
псевдоэлементов;	1	119	1178	341	284	4
передаточных элементов;	6902	11	766	–	89	–
выделенных логических сетей	1	1	13	4	8	4
Остаток логических элементов	–	34	228	36	3	3
Время выполнения, с	20,3460	0,0560	0,2410	0,0030	0,0140	0,0010

Заключение. Предложенные методы решения задачи извлечения логических сетей из двухуровневых SPICE-описаний реализованы как часть программы декомпиляции КМОП-схем на уровне транзисторов. Программа декомпиляции протестирована на практических схемах транзисторного уровня и имеет достаточное быстродействие, чтобы обрабатывать схемы более чем с 100 000 транзисторов за несколько минут на ПЭВМ. Декомпилированные схемы прошли проверку на соответствие исходной топологии транзисторной схемы с помощью разработанных средств верификации [18], а также с помощью Mentor Graphics Calibre nmLVS. Во всех случаях декомпилированные схемы успешно проходили проверку LVS топологии СБИС.

Вклад авторов. Л. Д. Черемисинова разработала метод решения задачи извлечения логических сетей из двухуровневых SPICE-описаний и подготовила текст статьи. Д. И. Черемисин разработал алгоритмы и программные средства построения логических сетей и генерации их описаний на языках высокого уровня.

Список использованных источников

1. Baker, R. J. CMOS Circuit Design, Layout, and Simulation / R. J. Baker. – Third ed. – Wiley-IEEE Press, 2010. – 1214 p.
2. Abadir, M. S. An improved layout verification algorithm (LAVA) / M. S. Abadir, J. Ferguson // Proc. of the European Design Automation Conf., Glasgow, UK, 12–15 Mar. 1990. – Glasgow, 1990. – P. 391–395.
3. Hunt, V. D. Reengineering: Leveraging the Power of Integrated Product Development / V. D. Hunt. – Wiley, 1993. – 283 p.

4. Rostami, M. A primer on hardware security: Models, methods, and metrics / M. Rostami, F. Koushanfar, R. Karri // *Proceedings of the IEEE*. – 2014. – Vol. 102, no. 8. – P. 1283–1295.
5. Tehranipoor, M. A survey of hardware trojan taxonomy and detection / M. Tehranipoor, F. Koushanfar // *IEEE Design & Test of Computers*. – 2010. – Vol. 27, no. 1. – P. 10–25.
6. Белоус, А. И. Основы кибербезопасности. Стандарты, концепции, методы и средства обеспечения / А. И. Белоус, В. А. Солодуха. – М. : Техносфера, 2021. – 482 с.
7. Черемисинов, Д. И. Извлечение сети логических элементов из КМОП-схемы транзисторного уровня / Д. И. Черемисинов, Л. Д. Черемисинова // *Микроэлектроника*. – 2019. – Т. 48, № 3. – С. 224–234. <https://doi.org/10.1134/S0544126919030037>
8. Yang, L. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits / L. Yang, C.-J. R. Shi // *Integration the VLSI J.* – 2006. – Vol. 39, no 4. – P. 311–339.
9. Zhang, N. The subcircuit extraction problem / N. Zhang, D. C. Wunsch, F. Harary // *Proc. IEEE Intern. Behavioral Modeling and Simulation Workshop*. – 2005. – Vol. 33(3). – P. 22–25.
10. Han, M. Efficient Subgraph Matching: Harmonizing Dynamic Programming, Adaptive Matching Order, and Failing Set Together / M. Han, H. Kim, G. Gu // *Proc. of Intern. Conf. on Management of Data (SIGMOD '19)*, Amsterdam, Netherlands, 30 June – 5 July 2019. – Amsterdam, 2019. – P. 1429–1446.
11. Черемисинов, Д. И. Канонизация графов при декомпиляции транзисторных схем / Д. И. Черемисинов, Л. Д. Черемисинова // *Информатика*. – 2022. – Т. 19, № 3. – С. 25–39. <https://doi.org/10.37661/1816-0301-2022-19-3-25-39>
12. Черемисинов, Д. И. Распознавание логических вентилях в плоской транзисторной схеме / Д. И. Черемисинов, Л. Д. Черемисинова // *Информатика*. – 2021. – Т. 18, № 4. – С. 96–107. <https://doi.org/10.37661/1816-0301-2021-18-4-96-107>.
13. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием производственно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов. – Минск : Беларус. навука, 2011. – 279 с.
14. Система логического проектирования функциональных блоков заказных КМОП СБИС с пониженным энергопотреблением / П. Н. Бибило [и др.] // *Микроэлектроника*. – 2018. – Т. 47, № 1. – С. 72–88.
15. Рабаи, Ж. М. Цифровые интегральные схемы. Методология проектирования : пер с англ. / Ж. М. Рабаи, А. Чандраксан, Б. Николич. – Изд. 2-е. – М. : Вильямс, 2007. – 912 с.
16. Черемисинов, Д. И. Распознавание логических структур на основе проходных транзисторов в КМОП-схеме / Д. И. Черемисинов, Л. Д. Черемисинова // *Информационные технологии и системы 2023 (ИТС 2023) = Information Technologies and Systems 2023 (ITS 2023)* : материалы Междунар. науч. конф., Минск, Беларусь, 22 нояб. 2023 г. – Минск : БГУИР, 2023. – С. 113–114.
17. Черемисинов, Д. И. Анализ и преобразование структурных описаний СБИС / Д. И. Черемисинов. – Минск : Беларуская навука, 2006. – 275 с.
18. Черемисинов, Д. И. Верификация логических схем из КМОП-транзисторов / Д. И. Черемисинов, Л. Д. Черемисинова // *Новые информационные технологии в исследовании сложных структур* : материалы 13-й Междунар. конф., 7–9 сент. 2020 г. – Томск : Изд. дом Томского гос. ун-та, 2020. – С. 150–151.

References

1. Baker R. J. *CMOS Circuit Design, Layout, and Simulation*. Third edition. Wiley-IEEE Press, 2010, 1214 p.
2. Abadir M. S., Ferguson J. An improved layout verification algorithm (LAVA). *Proceedings of the European Design Automation Conference, Glasgow, UK, 12–15 March 1990*. Glasgow, 1990, pp. 391–395.
3. Hunt V. D. *Reengineering: Leveraging the Power of Integrated Product Development*. Wiley, 1993, 283 p.
4. Rostami M., Koushanfar F., Karri R. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 2014, vol. 102, no. 8, pp. 1283–1295.
5. Tehranipoor M., Koushanfar F. A survey of hardware trojan taxonomy and detection. *IEEE Design & Test of Computers*, 2010, vol. 27, no. 1, pp. 10–25.
6. Belous A. I., Solodukha V. A. Osnovy kiberbezopasnosti. Standarty, kontseptsii, metody i sredstva obespecheniya. *Fundamentals of Cybersecurity. Standards, Concepts, Methods and Means of Support*. Moscow, Tekhnosfera, 2021, 482 p. (In Russ.).

7. Cheremisinov D. I., Cheremisinova L. D. *Extracting a logic gate network from a transistor-level CMOS circuit*. *Mikroelektronika [Russian Microelectronics]*, 2019, vol. 48, no. 3, pp. 224–234. <https://doi.org/10.1134/S0544126919030037> (In Russ.).
8. Yang L., Shi C.-J. R. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits. *Integration the VLSI Journal*, 2006, vol. 39, no 4, pp. 311–339.
9. Zhang N., Wunsch D. C., Harary F. The subcircuit extraction problem. *Proceedings IEEE International Behavioral Modeling and Simulation Workshop*, 2005, vol. 33(3), pp. 22–25.
10. Han M., Kim H., Gu G. Efficient Subgraph Matching: Harmonizing Dynamic Programming, Adaptive Matching Order, and Failing Set Together. *Proceedings of International conference on Management of Data (SIGMOD '19), Amsterdam, Netherlands, 30 June – 5 July 2019*. Amsterdam, 2019, pp. 1429–1446.
11. Cheremisinov D. I., Cheremisinova L. D. *Canonization of graphs during transistor circuits decompilation*. *Informatika [Informatics]*, 2022, vol. 19, no. 3, pp. 25–39 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-3-25-39> (In Russ.).
12. Cheremisinov D. I., Cheremisinova L. D. *Logical gates recognition in a flat transistor circuit*. *Informatika [Informatics]*, 2021, vol. 18, no. 4, pp. 96–107. <https://doi.org/10.37661/1816-0301-2021-18-4-96-107> (In Russ.).
13. Bibilo P. N., Romanov V. I. Logicheskoye proyektirovaniye diskretnykh ustroystv s ispol'zovaniyem produktsionno-fremovoy modeli predstavleniya znaniy. *Logical Design of Discrete Devices Using a Production-Frame Model of Knowledge Representation*. Minsk, Belaruskaja navuka, 2011, 279 p. (In Russ.).
14. Bibilo P. N., Avdeyev N. A., Kardash S. N., Kiriyeenko N. A., Lankevich Yu. Yu., ..., Cheremisinova L. D. *A System for Logical Design of Custom CMOS VLSI Functional Blocks with Reduced Power Consumption*. *Mikroelektronika [Russian Microelectronics]*, 2018, vol. 47, no. 1, pp. 72–88 (In Russ.).
15. Rabaev J. M., Chandrakasan A., Nikolic B. *Digital Integrated Circuits*, 2nd edition. Pearson, 2002, 800 p.
16. Cheremisinov D. I., Cheremisinova L. D. *Recognition of logical structures from pass transistors in a CMOS circuit*. Informatsionnye tehnologii i sistemy 2023 (ITS 2023) : materialy Mezhdunarodnoj nauchnoj konferencii, Minsk, Belarus', 22 nojabrja 2023 g. [*Information Technologies and Systems 2023 (ITS 2023) : Proceedings of the International Scientific Conference, Minsk, Belarus, 22 November 2023*]. Minsk, Belorusskij gosudarstvennyj universitet informatiki i radioelektroniki, 2023, pp. 113–114 (In Russ.).
17. Cheremisinov D. I. Analiz i preobrazovaniye strukturnykh opisaniy SBIS. *Analysis and Transformation of VLSI Structural Descriptions*. Minsk, Belaruskaja navuka, 2006, 275 p. (In Russ.).
18. Cheremisinov D. I., Cheremisinova L. D. *Verification of logic circuits from CMOS transistors*. Novyye informatsionnyye tehnologii v issledovanii slozhnykh struktur : materialy 13-j Mezhdunarodnoj konferencii, 7–9 sentyabrja 2020 g. [*New Information Technologies in the Study of Complex Structures : Proceedings of the 13th International Conference, 7–9 September 2020*]. Tomsk, Izdatel'skij dom Tomskogo gosudarstvennogo universiteta, 2020, pp. 150–151 (In Russ.).

Информация об авторах

Черемисинов Дмитрий Иванович, кандидат технических наук, доцент, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

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

Черемисинова Людмила Дмитриевна, доктор технических наук, профессор, главный научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

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

Information about the authors

Dmitry I. Cheremisinov, Ph. D. (Eng.), Assoc. Prof., Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

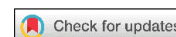
E-mail: cher@newman.bas-net.by

Ljudmila D. Cheremisinova, D. Sc. (Eng.), Prof., Chief Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

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

ЗАЩИТА ИНФОРМАЦИИ И НАДЕЖНОСТЬ СИСТЕМ

INFORMATION PROTECTION AND SYSTEM RELIABILITY



УДК 519.719.2
<https://doi.org/10.37661/1816-0301-2024-21-3-39-47>

Оригинальная статья
Original Article

Разделение секрета в специальной линейной группе

В. И. Янчевский¹, И. О. Говорушко^{1✉}, Г. В. Матвеев²

¹*Институт математики
Национальной академии наук Беларуси,
ул. Сурганова, 11, Минск, 220012, Беларусь
✉E-mail: govorushko88@gmail.com*

²*Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь*

Аннотация

Цели. Решается задача по разработке математических основ модулярного разделения секрета в специальной линейной группе над кольцом целых чисел.

Актуальность задачи определяется тем, что к схемам разделения секрета предъявляется большое число требований. К ним относятся идеальность схемы, возможность проведения верификации, изменения порога без участия дилера, реализации непороговой структуры доступа и некоторые другие.

Каждая разработанная к настоящему времени схема разделения секрета не в полной мере удовлетворяет всем этим требованиям. Она обладает лишь определенной конфигурацией требуемых свойств. Разработка же схемы на новой математической основе призвана расширить список таких конфигураций, что создает для пользователя больше возможностей в выборе оптимального варианта.

Методы. Используется теория групп, модулярная арифметика и теория схем разделения секрета.

Результаты. Строится фундаментальная область относительно действия главной конгруэнц-подгруппы правыми сдвигами в специальной линейной группе матриц второго порядка над кольцом целых чисел. На этой основе предложены способы модулярного разделения секрета и его порогового восстановления.

Заключение. Дано строгое математическое обоснование корректности алгоритмов генерации частичных секретов и восстановления основного секрета в специальной линейной группе над кольцом целых чисел. Эти результаты будут использованы для изучения конфигурации свойств разделения секрета в данной группе.

Ключевые слова: специальная линейная группа, конгруэнц-подгруппа, фундаментальная область, модулярное разделение секрета, пороговая структура доступа

Для цитирования. Янчевский, В. И. Разделение секрета в специальной линейной группе / В. И. Янчевский, И. О. Говорущко, Г. В. Матвеев // Информатика. – 2024. – Т. 21, № 3. – С. 39–47.

<https://doi.org/10.37661/1816-0301-2024-21-3-39-47>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.06.2024

Подписана в печать | Accepted 10.07.2024

Опубликована | Published 30.09.2024

Secret sharing in a special linear group

Vyacheslav I. Yanchevskii¹, Ihar A. Havarushka^{1✉}, Gennadii V. Matveev²

¹*Institute of Mathematics
of the National Academy of Sciences of Belarus,
st. Surganova, 11, Minsk, 220012, Belarus
✉E-mail: govorushko88@gmail.com*

²*Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus*

Abstract

Objectives. The problem of developing the mathematical foundations of modular secret sharing in a special linear group over the ring of integers is being solved.

The relevance of the problem is reduced to the fact that a large number of requirements are imposed on secret sharing schemes. These include the ideality of the scheme, the possibility of verification, changing the threshold without the participation of the dealer, the implementation of a non-threshold access structure and some others. Every secret sharing scheme developed to date does not fully satisfy all these requirements. It only has a certain configuration of these properties. The development of a scheme on a new mathematical basis is intended to expand the list of these configurations, which creates more opportunities for the user in choosing the optimal option.

Methods. Group theory, modular arithmetic and theory of secret sharing schemes are used.

Results. A fundamental domain with respect to the action of the main congruence subgroup by right shifts in the special linear group of second-order matrices over the ring of integers is constructed. On this basis, methods for modular secret sharing and its threshold restoration are proposed.

Conclusion. A rigorous mathematical justification is given for the correctness of the algorithms for generating partial secrets and restoring the main secret in the special linear group over the ring of integers. These results will be used to study the configuration of secret sharing properties in this group.

Keywords: special linear group, congruence subgroup, fundamental domain, modular secret sharing, threshold access structure

For citation. Yanchevskii V. I., Havarushka I. A., Matveev G. V. *Secret sharing in a special linear group*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 39–47 (In Russ.).

<https://doi.org/10.37661/1816-0301-2024-21-3-39-47>

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

Введение. В последнее время все большее значение приобретает организация схем доступа к тем или иным информационным ресурсам. Подобного рода задачи призваны решать схемы разделения секрета, относящиеся к числу важных криптографических протоколов. Они используются в системах электронного голосования [1], шифрования на основе атрибутов [2] и рас-

пределенных конфиденциальных вычислениях [3]. Такие схемы строятся на основе протоколов с нулевым разглашением.

С помощью схемы разделения секрета решается следующая задача. Пусть имеется некоторая важная информация (секрет) s и множество $I = \{1, 2, \dots, k\}$ пользователей. Требуется сообщить каждому пользователю i некоторую информацию s_i (частичный секрет) таким образом, чтобы только заранее определенные группы участников могли, объединяя свои частичные секреты, восстановить секрет s , а для остальных групп эта задача являлась бы трудноразрешимой. Как правило, под этим понимается, что задача восстановления секрета неразрешенной группой участников должна быть эквивалентна полному перебору.

Настоящее исследование посвящено модулярному подходу в теории разделения секрета. Основы этого подхода и теории в целом были заложены в статье А. Шамира [4], а собственно модулярный подход получил развитие в работах К. Асмута и Дж. Блюма [5] и М. Миньотта [6].

В дальнейшем модулярный подход был развит в работах [7–9]. В частности, он был обобщен на кольца многочленов от одной и нескольких переменных над полем Галуа. Было показано, что любая структура доступа допускает модулярную реализацию в кольцах целых чисел и полиномов над полями Галуа. В статье [7] доказано, что модулярный подход в кольце полиномов от одной переменной над полем Галуа позволяет реализовать любую структуру доступа совершенно и идеально. Более того, модулярная пороговая схема в кольце полиномов от одной переменной над полем Галуа легла в основу стандарта Республики Беларусь 12.34.101.60-2014 «Информационные технологии и безопасность. Алгоритмы разделения секрета». В работах [10, 11] были предложены методы верификации модулярных схем.

На сегодняшний день для схем разделения секрета разработано много критериев качества, например таких, как совершенность, идеальность, верифицируемость, пригодность для реализации предпороговых структур доступа и ряд других. Схем разделения секрета, удовлетворяющих всем известным критерием качества, все еще нет. Вот почему построение новых схем, основанных на принципиально иной алгебраической базе, представляет определенный интерес. В настоящей работе в качестве такой базы предлагается специальная линейная группа и модулярное разделение секрета в ней.

1. Модулярное разделение секрета. Дадим необходимые далее определения.

Под структурой доступа Γ понимают монотонное семейство подмножеств, т. е. предполагается, что для его элементов выполняется условие

$$A \in \Gamma, A \subset B \subset I \Rightarrow B \in \Gamma. \quad (1)$$

Такие подмножества называют разрешенными, а остальные – запрещенными. Структура доступа, когда разрешенными считаются подмножества A с условием $|A| \geq t$, называется пороговой, а число t , $1 \leq t \leq k$, называется ее порогом.

Схемой разделения секрета (СРС) называют алгоритмы распределения частичных секретов и восстановления исходного секрета. Они, в частности, должны обеспечивать правильное восстановление секрета разрешенными группами участников. Схему разделения секрета называют совершенной, если запрещенное множество участников не получает никакой информации о секрете, кроме априорной, и схему разделения секрета называют идеальной, если ключи всех участников и ключ s имеют один и тот же размер. Иногда в условие идеальности включают и совершенство схемы.

В самых общих чертах СРС позволяет распределить секрет между t участниками таким образом, чтобы заданные разрешенные множества участников могли однозначно восстановить секрет, а неразрешенные – не получили бы никакой дополнительной к имеющейся априорной информации о возможном значении секрета.

Среди СРС важное место занимают пороговые схемы. Одной из первых пороговых схем была схема Шамира. Она строится следующим образом. Рассмотрим многочлен $f(x)$ степени $k-1$ над конечным полем F_p при достаточно большом p . Объявим секретом некоторый коэф-

фициент этого многочлена, например $c = f(0)$. Распределим среди t участников информацию $f(x_i) = c_i, i = 1, 2, \dots, t$, чтобы каждый участник знал значение c_i этого многочлена в некоторой точке x_i . Тогда по известному свойству многочленов исходный многочлен может быть однозначно восстановлен по любым своим t -парам (x_i, c_i) . Это можно сделать при помощи интерполяционной формулы Лагранжа.

Модулярное разделение секрета в некотором смысле обобщает схему Шамира. Оно основано на следующем простом наблюдении (К. Асмут и Д. Блюм [5], М. Миньотт [6]). Пусть $m_1 < m_2 < \dots < m_l$ – система попарно взаимно простых натуральных модулей. Если секретом является некоторое натуральное число c , а секретом i -го участника, $i \in P = \{1, 2, \dots, l\}$, – наименьший неотрицательный вычет c по модулю m_i , т. е. $c_i \equiv c \pmod{m_i}$, то группа участников $A \subseteq P$ восстанавливает исходный секрет c путем решения системы сравнений $x \equiv c_i \pmod{m_i}, i \in A$. Это можно сделать, например, с помощью китайской теоремы об остатках. Правильно найдет секрет c лишь та группа участников A , для которой выполнено условие $c < \prod_{i \in A} m_i$. Тот же принцип используется при построении схемы разделения секрета над кольцом полиномов от одной и нескольких переменных [7–9].

Замечание 1. Схема Миньотта описана выше. В схеме Асмута – Блюма пользователи находят вспомогательный секрет, как показано выше. Хранимым секретом является вычет вспомогательного по некоторому несекретному модулю.

Замечание 2. Схему Шамира можно отнести к классу модулярных схем, поскольку значение многочлена в заданной точке $f(x_0) = c$ равно остатку от деления многочлена $f(x)$ на линейный многочлен $x - x_0$.

2. Фундаментальная область в специальной линейной группе. Как уже отмечалось, целью исследования является построение модулярной схемы разделения секрета в специальной линейной группе $SL_2(\mathbb{Z})$ (напомним, что это мультипликативная группа целочисленных 2×2 -матриц с определителем, равным 1). Эта группа тесно связана с исследованиями по модулярным функциям. В данной группе следует найти все необходимое для построения модулярных схем подобно тому, как это происходит в кольце целых чисел \mathbb{Z} . Необходимо иметь, во-первых, аналог деления с остатком, а во-вторых, аналог алгоритма CRT (Chinese Remainder Theorem, китайской теоремы об остатках).

Начнем с первой задачи. В качестве аналога кольца \mathbb{Z} возьмем группу $SL_2(\mathbb{Z})$, а в качестве модуля m – главную конгруэнц-подгруппу по модулю m . Она определяется следующим образом:

$$\Gamma(m) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}); \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \pmod{m} \right\}. \quad (2)$$

Здесь и далее сравнимость матриц по модулю понимают как их поэлементную сравнимость. Хорошо известно [12], что подгруппа $\Gamma(m)$ является нормальным делителем группы, а ее индекс находится по формуле

$$[SL_2(\mathbb{Z}) : \Gamma(m)] = m^3 \prod_{p|m} \left(1 - \frac{1}{p^2}\right), \quad (3)$$

где p – простые делители m .

Подгруппу $\Gamma(m)$ называют главной конгруэнц-подгруппой. Также потребуется промежуточная подгруппа $\Gamma_0(m) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}); c \equiv 0 \pmod{m} \right\}$. Известны следующие равенства [12]:

$$[\Gamma_0(m) : \Gamma(m)] = m^2 \prod_{p|m} \left(1 - \frac{1}{p}\right), \quad [SL_2(\mathbb{Z}) : \Gamma_0(m)] = m \prod_{p|m} \left(1 + \frac{1}{p}\right), \quad (4)$$

где p – простые делители m .

Главная конгруэнц-подгруппа $\Gamma(m)$ действует правыми сдвигами на группах $\Gamma_0(m)$ и $SL_2(\mathbb{Z})$. Система представителей орбит называется фундаментальной областью. Две матрицы A и B принадлежат одной орбите, если $A\Gamma(m) = B\Gamma(m)$. Это условие будет удобно трактовать иначе.

Лемма. Условие $A\Gamma(m) = B\Gamma(m)$ эквивалентно условию $A \equiv B \pmod{m}$.

Доказательство. Условие $A\Gamma(m) = B\Gamma(m)$ означает $AA_1 = BB_1$, где $A_1, B_1 \in \Gamma(m)$. Поэтому $A_1 = E + mA_2$, $B_1 = E + mB_2$. В связи с этим $A + mA_2 = B + mB_2$, откуда следует, что $A \equiv B \pmod{m}$.

Докажем обратное:

$$A \equiv B \pmod{m} \Rightarrow B^{-1}A \equiv E \pmod{m} \Rightarrow B^{-1}A \in \Gamma(m) \Rightarrow A \in B\Gamma(m),$$

что и требовалось доказать.

Для реализации пороговой модулярной схемы в группе $SL_2(\mathbb{Z})$ необходимо получить явное описание фундаментальной области относительно подгруппы $\Gamma(m)$, что является аналогом полной системы вычетов $0, 1, 2, \dots, m-1$ по модулю m в кольце \mathbb{Z} . Это можно было бы сделать с помощью техники подъема (лифтинга), поскольку канонический гомоморфизм $\varphi_m : SL_2(\mathbb{Z}) \rightarrow SL_2(\mathbb{Z})$ является сюръективным. Однако данная задача пока решена не в полной мере.

Замечание 3. Задача восстановления секрета в специальной линейной группе матриц второго порядка над кольцом целых чисел в несколько иных терминах комментируется в книге [13, с. 438, 439]: «Мы настоятельно рекомендуем читателю убедиться, что установить возможность подъема решений этого уравнения ($xy - zt = 1$) по некоторой системе модулей до целочисленного решения не так-то легко».

Авторы не претендуют на решение данной задачи в полной общности. В настоящей работе удалось построить значительную часть фундаментальной области, лежащую в группе $\Gamma_0(m)$. Этого оказалось достаточно для поставленных целей.

Дадим явное описание фундаментальной области группы $\Gamma_0(m)$ при действии на ней правыми сдвигами группы $\Gamma(m)$. В соответствии с леммой укажем семейство попарно несравнимых элементов группы $\Gamma_0(m)$ по модулю m в количестве, равном $[\Gamma_0(m) : \Gamma(m)] = m^2 \prod_{p|m} \left(1 - \frac{1}{p}\right) = m\varphi(m)$.

Прежде всего получим

$$\varepsilon\varepsilon' \equiv 1 \pmod{m} \Leftrightarrow \varepsilon\varepsilon' = 1 + km \Leftrightarrow \begin{pmatrix} \varepsilon & k \\ m & \varepsilon' \end{pmatrix} \in \Gamma_0(m). \quad (5)$$

Частичными секретами участников служат поэлементные вычеты этой матрицы по модулям m_1, \dots, m_k . Например, частичным секретом первого участника будет образ матрицы S при каноническом эпиморфизме

$$SL_2(\mathbb{Z}) \rightarrow SL_2(\mathbb{Z}) / \Gamma(m_1) \cong SL_2(\mathbb{Z}_{m_1}), \quad (8)$$

что является аналогом обычного частичного секрета в схеме Миньотта.

3. Восстановление секрета S по частичным секретам подмножества участников A , где $|A| \geq t$. Секрет S восстанавливается по следующему алгоритму:

- m находится автоматически;
 - s'_i находится по китайской теореме об остатках путем решения системы сравнений, причем найденное решение в силу выбора s'_i будет одним и тем же по модулям $\prod_{l \in A} m_l$ и m , так как $s'_i < \prod_{l \in A} m_l$;
 - s_i находится путем решения сравнения $s_i s'_i \equiv 1 \pmod{m}$. Напомним, что все модули m_1, \dots, m_k известны участникам;
 - k_i находится по формуле $k_i = \frac{s_i s'_i - 1}{m}$;
 - параметр j находится при условии, что известно, с чем сравнимо $k_i + j s'_i$ по модулю m .
- Так как $(s'_i, m) = 1$, то значение j единственно по модулю m .

Таким образом, матрица $S = \begin{pmatrix} s_i + jm & k_i + j s'_i \\ m & s'_i \end{pmatrix}$ корректно восстановлена.

Пример (2,3)-пороговой СРС в группе $SL_2(\mathbb{Z})$. В качестве числовых модулей возьмем $m_1 = 5, m_2 = 6, m_3 = 7$. Тогда имеем схему Миньотта с параметрами $M_1 = 7 < 5 \cdot 6 = 30 = M_2$, т. е. хранимый секрет следует брать из промежутка $7 < s'_i < 30$. Возьмем, например, $s'_i = 11$.

В соответствии с п. 1 предложенной схемы в группе $SL_2(\mathbb{Z})$ открытыми ключами участников будут конгруэнц-подгруппы $\Gamma(5), \Gamma(6), \Gamma(7)$. В соответствии с п. 2 секретом в группе $SL_2(\mathbb{Z})$ будет матрица из фундаментальной области в группе $\Gamma_0(m)$, где $m = 5 \cdot 6 \cdot 7 = 210$. Поскольку $11 \cdot 191 = 1 + 210 \cdot 10$, откуда $s_i = 191$ и $k_i = 10$, в качестве такой матрицы берем

$$S = \begin{pmatrix} 401 & 21 \\ 210 & 11 \end{pmatrix}. \text{ Частичными секретами участников будут матрицы } \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 5 & 3 \\ 0 & 5 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}.$$

С помощью алгоритма CRT любые два участника правильно восстановят секрет S .

Заключение. В работе построена фундаментальная область группы $\Gamma_0(m)$ при действии на ней группы $\Gamma(m)$. Предложен алгоритм модулярного разделения секрета, где в качестве открытых ключей участников взяты главные конгруэнц-подгруппы.

Вклад авторов. В. И. Янчевский сформулировал и обосновал алгебраическую часть, И. О. Говорушко составил алгоритм и написал текст статьи, Г. В. Матвеев обосновал криптографическую часть.

Список использованных источников

1. Cramer, R. Multiparty Computation from Threshold Homomorphic Encryption / R. Cramer, I. Damgard, J. Nielsen // LNCS. – 2001. – Vol. 2045. – P. 280–300.
2. Bethencourt, J. Ciphertext-policy attribute-based encryption / J. Bethencourt, A. Sahai, B. Waters // Proc. of IEEE Symp. on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007. – Berkeley, 2007. – P. 321–334.
3. Benaloh, J. Secret sharing homomorphisms: keeping shares of a secret sharing / J. Benaloh // LNCS. – 1987. – Vol. 263. – P. 251–260.
4. Shamir, A. How to share a secret / A. Shamir // Communications of the ACM. – 1979. – Vol. 22. – P. 612–613. <https://doi.org/10.1145/359168.359176>
5. Asmuth, C. A modular approach to key safeguarding / C. Asmuth, J. Bloom // IEEE Transactions on Information Theory. – 1983. – Vol. 29. – P. 156–169. <https://doi.org/10.1109/TIT.1983.1056651>
6. Mignotte, M. How to share a secret / M. Mignotte // LNCS. – 1983. – Vol. 149. – P. 371–375.
7. Galibus, T. Some structural and security properties of the modular secret sharing / T. Galibus, G. Matveev, N. Shenets // 2008 10th Intern. Symp. on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 26–29 Sept. 2008. – Timisoara, 2008. – P. 197–200. <https://doi.org/10.1109/SYNASC.2008.14>
8. Galibus, T. Generalized Mignotte's Sequences Over Polynomial Rings / T. Galibus, G. Matveev // Electronic Notes in Theoretical Computer Science. – 2007. – Vol. 186. – P. 43–48. <https://doi.org/10.1016/j.entcs.2006.12.044>
9. Galibus, T. Finite Fields. Gröbner Bases and Modular Secret Sharing / T. Galibus, G. Matveev // J. of Discrete Mathematical Sciences and Cryptography. – 2012. – Vol. 15. – P. 339–348. <https://doi.org/10.1080/09720529.2012.10698386>
10. Васьковский, М. М. Верификация модулярного разделения секрета / М. М. Васьковский, Г. В. Матвеев // Журнал Белорусского государственного университета. Математика. Информатика. – 2017. – № 2. – С. 17–22.
11. Матвеев, Г. В. Совершенная верификация модулярной схемы / Г. В. Матвеев, В. В. Матулис // Журнал Белорусского государственного университета. Математика. Информатика. – 2018. – № 2. – С. 4–9.
12. Di Matteo, G. The action of $SL_2(\mathbb{Z})$ on the upper-half complex plane / G. Di Matteo. – Mode of access: <https://www.dimatteo.is/Mathematics/Courses/Modular-forms/02-SL2Z.pdf>. – Date of access: 10.04.2024.
13. Платонов, В. П. Алгебраические группы и теория чисел / В. П. Платонов, А. С. Рапичук. – М. : Наука, 1991. – 656 с.

References

1. Cramer R., Damgard I., Nielsen J. Multiparty Computation from Threshold Homomorphic Encryption. *LNCS*, 2001, vol. 2045, pp. 280–300.
2. Bethencourt J., Sahai A., Waters B. Ciphertext-policy attribute-based encryption. *Proceedings of IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007*. Berkeley, 2007, pp. 321–334.
3. Benaloh J. Secret sharing homomorphisms: keeping shares of a secret sharing. *LNCS*, 1987, vol. 263, pp. 251–260.
4. Shamir A. How to share a secret. *Communications of the ACM*, 1979, vol. 22, pp. 612–613. <https://doi.org/10.1145/359168.359176>
5. Asmuth C., Bloom J. A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, 1983, vol. 29, pp. 156–169. <https://doi.org/10.1109/TIT.1983.1056651>
6. Mignotte M. How to share a secret. *LNCS*, 1983, vol. 149, pp. 371–375.
7. Galibus T., Matveev G., Shenets N. Some structural and security properties of the modular secret sharing. *2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 26–29 September 2008*. Timisoara, 2008, pp. 197–200. <https://doi.org/10.1109/SYNASC.2008.14>
8. Galibus T., Matveev G. Generalized Mignotte's Sequences Over Polynomial Rings. *Electronic Notes in Theoretical Computer Science*, 2007, vol. 186, pp. 43–48. <https://doi.org/10.1016/j.entcs.2006.12.044>
9. Galibus T., Matveev G. Finite Fields. Gröbner Bases and Modular Secret Sharing. *Journal of Discrete Mathematical Sciences and Cryptography*, 2012, vol. 15, pp. 339–348. <https://doi.org/10.1080/09720529.2012.10698386>

10. Vaskouski M. M., Matveev G. V. *Verification of modular secret sharing*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [Journal of the Belarusian State University. Mathematics and Informatics], 2017, no. 2, pp. 17–22 (In Russ.)

11. Matveev G. V., Matulis V. V. *Perfect verification of modular scheme*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [Journal of the Belarusian State University. Mathematics and Informatics], 2018, no. 2, pp. 4–9 (In Russ.)

12. Di Matteo G. The action of $SL_2(\mathbb{Z})$ on the upper-half complex plane. Available at: <https://www.dimatteo.is/Mathematics/Courses/Modular-forms/02-SL2Z.pdf> (accessed 10.04.2024).

13. Platonov V. P., Rapinchuk A. S. Algebraicheskie gruppy i teoriya chisel. *Algebraic Groups and Number Theory*. Moscow, Nauka, 1991, 656 p. (In Russ.).

Информация об авторах

Янчевский Вячеслав Иванович, доктор физико-математических наук, академик Национальной академии наук Беларуси, заведующий отделом алгебры, Институт математики Национальной академии наук Беларуси.

E-mail: yanch@im.bas-net.by

Говорущко Игорь Олегович, кандидат физико-математических наук, научный сотрудник, Институт математики Национальной академии наук Беларуси.

E-mail: govorushko88@gmail.com

<https://orcid.org/0009-0004-9914-1635>

Матвеев Геннадий Васильевич, кандидат физико-математических наук, доцент, доцент кафедры высшей математики факультета прикладной математики и информатики, Белорусский государственный университет.

E-mail: matveev@bsu.by

Information about the authors

Vyacheslav I. Yanchevskii, D. Sc. (Phys.-Math.), Acad. of the National Academy of Sciences of Belarus, Head of the Algebra Department, Institute of Mathematics of the National Academy of Sciences of Belarus.

E-mail: yanch@im.bas-net.by

Ihar A. Havarushka, Ph. D. (Phys.-Math.), Researcher, Institute of Mathematics of the National Academy of Sciences of Belarus.

E-mail: govorushko88@gmail.com

<https://orcid.org/0009-0004-9914-1635>

Gennadii V. Matveev, Ph. D. (Phys.-Math.), Assoc. Prof., Assoc. Prof. of the Department of Higher Mathematics of the Faculty of Applied Mathematics and Computer Sciences, Belarusian State University.

E-mail: matveev@bsu.by

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

INTELLIGENT SYSTEMS



УДК 004.032.26, 004.942, 519.876.5
<https://doi.org/10.37661/1816-0301-2024-21-3-48-62>

Оригинальная статья
Original Article

Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта

Т. Ю. Ким[✉], Г. А. Прокопович

Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
[✉]E-mail: tatyana_kim92@mail.ru

Аннотация

Цели. Поставлена цель разработать новый метод обучения системы управления мобильным роботом поиску выхода из лабиринта на основе обучения с подкреплением и алгоритма правой руки.

Методы. В работе применен метод компьютерного моделирования в среде MATLAB/Simulink.

Результаты. Предложен новый метод обучения системы управления мобильным роботом, способный реализовывать алгоритм правой руки для поиска выхода из лабиринта. Данный метод основан на работе двух агентов, взаимодействующих между собой: первый непосредственно реализует поисковый алгоритм и ищет выход из лабиринта, а второй, следуя за ним, с помощью метода подражательного обучения пытается научиться находить выход из лабиринта. Агент-эксперт, реализуя дискретный алгоритм движения по лабиринту, совершает точные дискретные шаги и движется почти независимо от второго агента. Единственным ограничением является скорость его движения, которая прямо пропорционально зависит от расстояния между агентами. Второй агент, агент-ученик, методом проб и ошибок старается сократить расстояние до первого. Для реализации процесса обучения использовался метод обучения с подкреплением в режиме подражания, для которого была разработана соответствующая функция вознаграждения, позволяющая удерживать центр масс робота в центре коридора и при необходимости поворачивать, следуя за агентом-экспертом. Агенты передвигаются по виртуальному полигону, состоящему из разветвленных коридоров, достаточно широких для реализации различных маневров движений.

Заключение. Было доказано, что благодаря предложенному методу подражательного обучения агент-ученик способен не только перенимать от агента-эксперта требуемые паттерны поведения (искать в ранее неизвестном лабиринте выход по алгоритму правой руки), но и самостоятельно приобретать новые (изменять скорость на повороте, обходить небольшие коридоры-тупики), которые положительным образом влияют на выполнение поставленной задачи.

Ключевые слова: мобильный робот, агент, обучение с подкреплением, алгоритм правой руки, лабиринт, подражательное обучение

Благодарности. Работа была выполнена при поддержке гранта БРФФИ Ф22КИТГ-002 и задания Т31 ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства» (2021–2025).

Для цитирования. Ким, Т. Ю. Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта / Т. Ю. Ким, Г. А. Прокопович // Информатика. – 2024. – Т. 21, № 3. – С. 48–62.
<https://doi.org/10.37661/1816-0301-2024-21-3-48-62>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.07.2024

Подписана в печать | Accepted 15.08.2024

Опубликована | Published 30.09.2024

Development of an imitation learning method for a neural network system of mobile robot's movement on example of the maze solving

Tatyana Yu. Kim[✉], Ryhor A. Prakapovich

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*

[✉]E-mail: tatyana_kim92@mail.ru

Abstract

Objectives. To develop a new method for training a mobile robot control system to use a maze solver algorithm based on reinforcement learning and the right-hand algorithm.

Methods. The work uses the method of computer modeling in the MATLAB/Simulink environment.

Results. A new method for training a mobile robot control system capable of implementing the right-hand algorithm for finding an exit from a maze is proposed. The proposed method is based on the work of two agents interacting with each other: the first directly implements the search algorithm and searches for an exit from the maze, and the second, following it, tries to learn using the imitation learning method. The expert agent, implementing a discrete algorithm for moving through the maze, makes precise discrete steps and moves almost independently of the second agent. The only limitation is its speed, which is directly proportional to the distance between the agents. The second agent, the student agent, tries to reduce the distance to the first agent by trial and error. The learning process was implemented using the reinforcement learning method, which was used in the imitation mode and for which a corresponding reward function was developed, allowing the robot's center of mass to be kept in the center of the corridor and, if necessary, to turn, following the expert agent. The agents move along a virtual polygon consisting of branched corridors wide enough to implement various movement maneuvers.

Conclusion. It was proven that, thanks to the proposed method of imitative learning, the student agent is able not only to adopt the required behavior patterns from the expert agent – to search for an exit in a previously unknown labyrinth using the right-hand algorithm, but also to independently acquire new ones (changing speed on a turn, bypassing small dead-end corridors), which positively influence the performance of the assigned task.

Keywords: mobile robot, agent, reinforcement learning, right-hand algorithm, maze, imitative learning

Acknowledgments. The work was supported by the BRFFR grant F22KITG-002 and the task T31 of the State Program for Scientific Research "Digital and Space Technologies, Security of Man, Society and the State" (2021–2025).

For citation. Kim T. Yu., Prakapovich R. A. *Development of an imitation learning method for a neural network system of mobile robot's movement on example of the maze solving*. *Informatika [Informatics]*, 2024, vol. 21, no. 3, pp. 48–62 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-48-62>

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

Введение. Роботы все больше интегрируются в различные сферы жизни общества, включая здравоохранение, военную сферу, реагирование на стихийные бедствия, мониторинг окружающей среды и бытовые задачи. Способность к автономной навигации увеличивает их полезность при выполнении сложных задач в непредсказуемых условиях. Автономная навигация осуществляется, когда робот перемещается в среде без какого-либо вмешательства со стороны внешнего контроллера (например, человека). Она является одной из ключевых тем исследований в области мобильной робототехники [1]. Благодаря развитию искусственного интеллекта и компьютерного зрения были достигнуты огромные успехи в автономной навигации мобильных роботов [2]. Однако по-прежнему остается сложной задачей обеспечения автономной навигацией мобильных роботов в реальном мире.

За последние несколько лет популярность глубокого обучения с подкреплением резко возросла. Она началась с двух историй успеха, когда впечатляющие результаты были достигнуты при сочетании обучения с подкреплением с глубокими нейронными сетями.

Во-первых, сообществом DeepMind разработан агент обучения с подкреплением, который способен играть одновременно в несколько видеоигр Atari 2600 на человеческом уровне [3]. В основе агента лежит метод, известный как глубокая Q-сеть, которая использует многослойную нейронную сеть в качестве аппроксиматора функции Q-обучения и решает проблему нестабильности. Во-вторых, сообществом DeepMind разработана программа AlphaGo [4], победившая чемпиона мира Ли Седоля в настольной игре Go. AlphaGo демонстрирует эффективное сочетание контролируемого обучения и обучения с подкреплением для освоения стратегической игры Go. Глубокое обучение с подкреплением в робототехнике по-прежнему остается сложной задачей, однако в последние годы оно применяется в таких областях, как роботизированная манипуляция [5], локомоция [6] и автономное управление автомобилем [7–9]. Трудности создания беспилотных систем управления заключаются не столько в проблемах распознавания различных объектов на дороге [10], сколько в описании сложных зрительных сцен и выборе соответствующей последовательности действий. Примером может служить шагающий робот, перемещающийся в сложной среде и адаптирующийся к изменениям условий с большой автономностью и эффективностью [11]. Как правило, при этом процесс обучения заключается в описании проблемы в виде оптимизационной задачи, для которой требуется найти минимум или максимум функции вознаграждения. Для большинства практических задач можно формализовать и реализовать процесс оптимизации на основе одной целевой функции, но есть задачи, где функция вознаграждения не является гладкой функцией, которую можно представить последовательностью условных (логических) операторов. Таким образом, в процессе усложнения задачи ставится цель усложнить вознаграждение для формирования лучшей политики. Также в процессе обучения агент привыкает решать однотипные задачи. В данных случаях традиционное обучение с подкреплением не может осуществляться с помощью обычных алгоритмов¹.

Из-за большого числа возможных вариантов точный выбор действий может стать сложной задачей. На высоких размерностях и длинных последовательностях действий такой подход уже не справляется. В результате при выполнении сложных задач в структурированных средах конечное решение принимает оператор или водитель с помощью сложных программных систем.

Чтобы правильно передвигаться, агент должен понимать окружающую среду и действовать согласно данным, соответствующим реальному миру. Для решения подобных задач существуют методы с применением подражательного обучения (Imitation Learning) [12]. Реализация данных методов заключается в том, что происходит взаимодействие только эксперта и награды, а ученик взаимодействует лишь с экспертом [13, с. 208]. В этих методах есть свои недостатки – это трудности в реализации наблюдения за движением эксперта, который, в свою очередь, контролирует двигательный аппарат ученика [14]. При этом ученик не видит всей картины и конечной цели.

¹Part 2: Kinds of RL Algorithms [Electronic resource]. – Mode of access: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html. – Date of access: 12.06.2024.

На основе подражательного обучения авторами предложен новый метод реализации системы управления мобильным роботом. Традиционное обучение с подкреплением управляет учеником, а алгоритм правой руки – экспертом. Обучение является сложным, так как задача может быть каждый раз разной и иметь различную навигацию, в том числе включать различные образцы. Необходимо сделать так, чтобы ученик не просто копировал действие эксперта в процессе обучения, а, скорее, пытался достичь цели эксперта, выполняя новые действия, подобные таким, которые выполняет эксперт.

В настоящей работе описывается попытка создания программного комплекса для решения поставленных задач. Для упрощения и первого приближения задача автономного управления может быть формализована как задача поиска выхода из лабиринта. В качестве системы правил движения в условной местности – лабиринте – было предложено воспользоваться алгоритмом выхода из лабиринта по правилу правой руки.

1. Исследование методов классического обучения с подкреплением. Метод обучения с подкреплением основан на реализации процесса максимизации некоторого сигнала вознаграждения при переборе различных вариантов поведения исследуемых систем-агентов. Агент учится выполнять те действия, которые могут принести ему наибольшую награду. В наиболее интересных и важных случаях действия агента могут влиять не только на локальные вознаграждения, получаемые немедленно, но и на возникшую ситуацию в целом. Формирование долгосрочной награды – достаточно сложный процесс, так как правильно сформированная награда принесет лучший результат, сократит время обучения и процесс обучения пройдет качественнее [15].

Принцип обучения с подкреплением показан на рис. 1, где s_t – состояние агента в момент времени t ; a_t – действие, совершаемое агентом в момент времени t в среде. Следующее состояние s_{t+1} среды достигается действия a_t , при этом среда генерирует новую обратную связь r_{t+1} в следующем состоянии s_{t+1} . Действие a_{t+1} агент выполняет с помощью s_{t+1} и r_{t+1} , повторяя этот процесс до тех пор, пока не достигнет конца итерации [16].

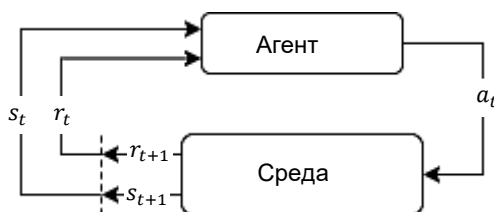


Рис. 1. Принцип обучения с подкреплением
Fig. 1. Reinforcement learning principle

Одним из классических алгоритмов обучения с подкреплением является алгоритм Q-learning, который обладает высокой надежностью и способностью адаптироваться к неопределенной среде [17], но в то же время имеет такие недостатки, как длительное время обучения, низкая эффективность исследования и медленная скорость сходимости [18].

1.1. Применение классического алгоритма Q-learning. Q-learning – это безмодельный алгоритм обучения с подкреплением, позволяющий узнавать ценность действий в определенном состоянии методом проб и ошибок. Оптимальное действие аппроксимируется с помощью алгоритма Q-learning, который постоянно обновляет функцию значения состояния-действия $Q(s_t, a_t)$ во время итерации [19]. Q-значение алгоритма Q-learning обновляется по формуле

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [R_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (1)$$

где $\alpha \in [0, 1]$ – параметр скорости обучения, R_t – вознаграждение состояния s в момент времени t , $\gamma \in [0, 1]$ – коэффициент дисконтирования.

В результате создается новая таблица, называемая Q-таблицей, в которой хранится вся информация о состоянии и действии агента.

В работах [20, 21] описано обучение с подкреплением, где использовался алгоритм Q-learning, идея которого состоит в том, чтобы применять табличный метод обучения, представленный по координатам.

Для того чтобы агент выполнял поставленную задачу, пространство его действий задается следующим образом:

$$\text{Action} = [\text{up}, \text{down}, \text{left}, \text{right}].$$

При этом угол направленности агента имеет вид

$$\text{Rot}_A = [90^\circ, -90^\circ, 0^\circ, 180^\circ].$$

В эксперименте, поставленном авторами, сетка (лабиринт) состоит из 27×27 ячеек. Координаты положения агента рассчитываются по формуле (1).

Классический алгоритм использует моделирование на основе сеток (таблиц), поэтому размер его пространства состояний может определяться размером таблицы. Один из недостатков Q-learning заключается в том, что матрица вознаграждений и политики имеет ту же размерность, что и лабиринт, и оценивается согласно карте, на которой обучался агент (рис. 2, a).

На рис. 2, b показан результат обученного агента на экспериментальной карте 27×27 . Видно, что агент достиг конечной точки, так как сформированная политика и матрица вознаграждения соответствуют первоначальной карте, на которой обучался робот. Если изменить размер или цель в лабиринте, то агенту необходимо переучиваться с нуля на новом лабиринте.

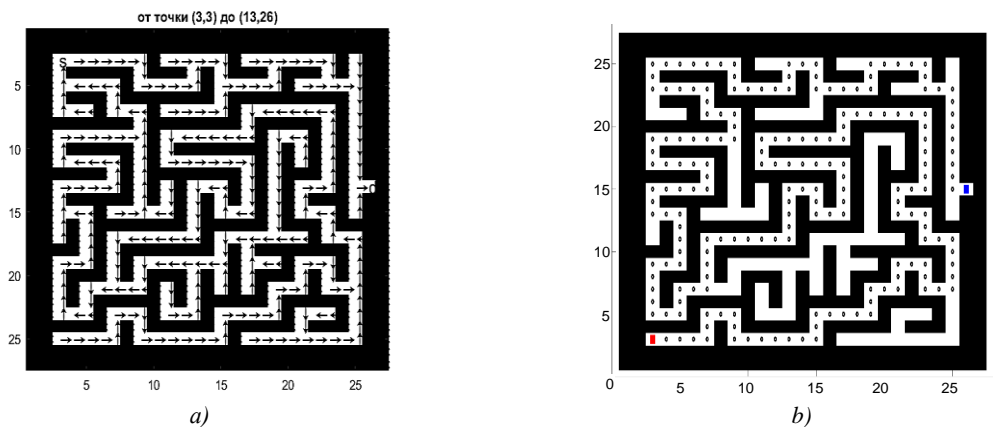


Рис. 2. Реализация алгоритма Q-learning: a) иллюстрация функции политики с помощью диаграммы политики; b) результат моделирования алгоритма

Fig. 2. Implementation of the Q-learning algorithm: a) illustrating a policy function using a policy diagram; b) algorithm simulation result

Черным цветом обозначены стены, красным – стартовая позиция, синим – выход из лабиринта
The walls are indicated in black, the starting position in red, and the exit from the maze in blue

Для того чтобы преодолеть возникшую проблему, представим состояния более общим способом в виде пикселей на карте и данных с лидара. Состояния, выраженные в виде функции, используют аппроксимацию функции, которая позволит обобщить множество различных лабиринтов, применяемых в обучении. Авторам неизвестны работы, в которых доказана возможность прохождения обученным роботом по неизвестной карте.

1.2. Алгоритм Актор-Критик обучения с подкреплением. Более сложным и варибельным методом реализации обучения с подкреплением является подход «глубокое обучение с подкреплением», использующий современный алгоритм Актор-Критик, который позволяет работать в непрерывных средах [22, 23]. Алгоритм содержит две нейронные сети (рис. 3): Актор и Критик. Актор решает, какое действие следует предпринять, а Критик сообщает Актору, насколько хорошим было действие и как его следует откорректировать.

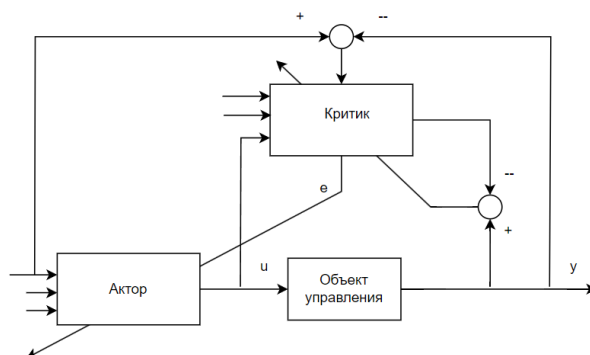


Рис. 3. Архитектура алгоритма Актор-Критик
Fig. 3. Architecture of the Actor-Critic Algorithm

Обучение объекта управления основано на стратегическом градиентном подходе. Критик оценивает действие, произведенное Актором, вычисляя функцию вознаграждения. Поэтому разработка функции вознаграждения для решения задач – возможно, самое важное в глубоком обучении с подкреплением. Награда должна помогать агенту совершать действия, которые максимизируют долгосрочную награду, а чем больше рекомендаций дать, тем быстрее и правильнее агент научится. Дополнительным критерием является получение награды агентом после выполнения определенного действия. Цель агента – получить награду, которая будет соответствовать поставленной задаче.

Для того чтобы агент передвигался по лабиринту, сначала необходимо создать подходящие условия, а затем обучить агента. Для этого следует выполнить следующие шаги:

Наблюдение. Данные для наблюдения представлены 27-элементным вектором, в котором 25 элементов содержат информацию с лидара и два элемента – сигналы обратной связи по линейной и угловой скоростям.

Критерии останова. Моделирование начинается заново, если минимальное расстояние, полученное по показаниям лидара (lid_{min}), меньше радиуса ученика (2) или сумма каждого четвертого элемента вектора (3) равна нулю:

$$isdone = \begin{cases} lid_{min} \leq Rob_{rad}, & (2) \\ \sum_{i=1}^5 v_{4i}(t) = 0, & (3) \end{cases}$$

где $v_{4i}(t)$ – каждый четвертый элемент 20-значного вектора (v) в момент времени (t).

Вознаграждение. На данном этапе формируется положительное вознаграждение, называемое наградой. Лидар определяет ближайшее препятствие впереди на пути передвижения, а получение награды поощряет прямолинейное движение по следующей формуле:

$$lid_{min} = \frac{1}{5} \cdot \sum_{k=-2}^2 lid_{i+k} - Rob_{rad}, \quad (4)$$

где lid_{i+k} – значение пяти центральных лучей лидара с (k), изменяющимся от -2 до 2 .

Учитывая минимальный показатель лидара (lid_{min}), агент не допускает столкновения с препятствием. Награда вычисляется по формуле

$$reward = 0,5 \cdot lid_{min} + 0,8 \cdot v_{rew} + k \cdot finish, \quad (5)$$

где v_{rew} – средняя линейная скорость последних пяти шагов моделирования; $k = 10$ – коэффициент, который подбирался экспериментально для того, чтобы агент получил наивысшую

награду за выход из лабиринта и наказание за столкновение с внешней стеной; *finish* – переменная, которая принимает значение 1, если ученик достиг выхода из лабиринта, и 0 – в противном случае.

Отрицательное вознаграждение, называемое штрафом (наказанием) агента за каждое действие, побуждает его делать как можно меньше лишних шагов и сокращать время обучения. Для этого было решено использовать угловую скорость (w), чтобы уменьшить ее влияние на линейную (v). Функция штрафа вычисляется по формуле

$$\text{penalty} = 0,2 \cdot w + v_{\text{penalty}} + 9 \cdot \text{isdone}, \quad (6)$$

где w – угловая скорость, *isdone* – условия критериев остановки (2), (3), а v_{penalty} определяется по условию

$$v_{\text{penalty}} \begin{cases} -1, & \text{если } v = 0, \\ 0, & \text{если } v > 0. \end{cases} \quad (7)$$

Совместив награду (5) и штраф (6), получим функцию вознаграждения для агента:

$$\text{reward}_{\text{Agent}} = \text{reward} - \text{penalty}. \quad (8)$$

Обучение. Шаг моделирования составляет 0,1 с. Моделирование завершается, если закончилось время, отведенное для этого, или агент достиг своей цели. На рис. 4 видно, что обученный агент обходит препятствия и поворачивает, но его траектория еще не соответствует предполагаемой (выделена зеленым).



Рис. 4. Синий путь – траектория движения агента, который передвигается по коридору, зеленый путь – предполагаемая для него траектория

Fig. 4. The blue path is the trajectory of the agent moving along the corridor, the green path is the expected trajectory

Таким образом, при применении алгоритма Q-learning идет поиск оптимального пути, что хорошо для решения определенных задач. Однако сформированная таблица актуальна только для карты, которая была на этапе обучения. Обученный агент находил выход из лабиринта, указав на карте любую стартовую точку. Тот же агент может не справиться, если разместить его на новой карте, так как она не соответствует его политике обучения. Кроме того, агент действует дискретно, т. е. передвигается по клеткам, и действия его ограничены. При применении алгоритма Актор-Критик агент действует непрерывно. Данные, полученные от наблюдения, недостаточны для поставленной цели, так как функция вознаграждения сложная, нелинейная и разрывная и требует последовательных логических решений после получения определенных данных от сенсора. Поэтому авторы предлагают свой метод реализации поставленной задачи.

2. Комбинированный метод системы управления с применением обучения с подкреплением и алгоритма правой руки. Так как широко используемые методы обучения с подкреплением не позволили реализовать обучение нейросетевой системы управления автономным роботом в задаче поиска выхода из лабиринта, авторами был предложен кардинально новый подход. Ключевой идеей предлагаемого метода является введение в систему управления дополнительного агента, выполняющего роль эксперта, который действует по правилам так, как бы хотелось авторам, чтобы действовал искомый агент-ученик. В этом случае агент представляет ученика. Тогда функция вознаграждения будет определяться физическими величинами, которыми можно описать систему взаимодействия эксперта и ученика.

2.1. План эксперимента. Предположим, что есть два агента, условно называемые экспертом и учеником, которых поместили в неизвестную среду (лабиринт). Предлагается использовать навигационную схему эксперта, который руководствуется дискретным алгоритмом правой руки. Цель ученика – найти выход из лабиринта, следуя за экспертом и используя алгоритм обучения с подкреплением. Суть эксперимента заключается в том, чтобы обучить ученика следовать за экспертом, причем таким образом, чтобы положение ученика в процессе движения по лабиринту было максимально близким к центру коридора. При этом робот может двигаться только вперед, совершая при необходимости повороты и даже развороты. В результате это помогает ученику держаться центра коридора и гарантированно найти выход.

На рис. 5 показана схема предлагаемого метода совместного использования алгоритма обучения с подкреплением и алгоритма правой руки. Дискретный агент функционирует на основе алгоритма правой руки и выступает в качестве ориентира для обучения второго, аналогового, агента, нейросетевая система управления которого способна экстраполировать входные сенсорные данные на управляющие сигналы своего привода. Оба алгоритма взаимодействуют со средой, от которой они получают сенсорные данные о местоположении и углах направления их движений. Однако аналоговый агент получает больше сенсорных данных, так как у него есть собственный лидар, также он получает сенсорные данные о расстоянии до первого агента. В свою очередь, дискретный агент получает от второго агента управляющий сигнал о расстоянии между ними, который может замедлять его собственную скорость движения.

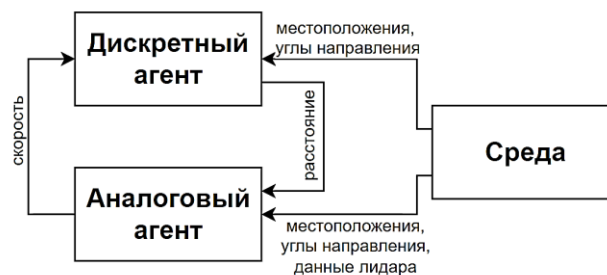


Рис. 5. Схема предложенного метода применения обучения с подкреплением совместно с алгоритмом правой руки

Fig. 5. Scheme of the proposed method of applying reinforcement learning together with the right-hand algorithm

Рассмотрим функциональную схему на рис. 6, где *осн* – оценочное состояние награды. Красной пунктирной линией обозначено состояние, от которого зависит скорость передвижения эксперта по лабиринту. Предложенная комбинация глубокого обучения с подкреплением и алгоритма правой руки работает следующим образом. Блок Среда включает ученика и эксперта. Блок Ученик, который является объектом управления, принимает множество действий (v, w) из блоков Актора и Критика. Сеть Критика корректирует (направляет) обучение Актора путем оценивания значения награды (*осн*) за действие-состояние. Актор выдает окончательную стратегию обучения в каждой итерации.

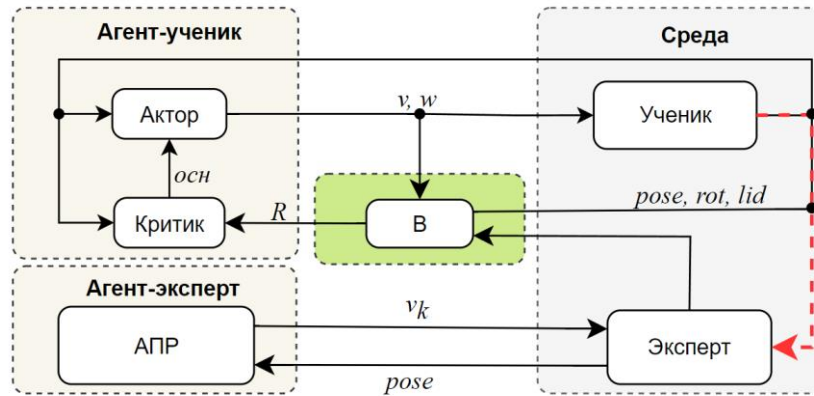


Рис. 6. Схема применения системы Актор-Критик глубокого обучения с подкреплением в блоке Агент-ученик совместно с алгоритмом правой руки (АПР) в блоке Агент-эксперт

Fig. 6. The application diagram shows the Actor-Critic system of deep reinforcement learning in the Agent-student block together with the right-hand algorithm (RHA) in the Agent-expert block

2.2. Реализация предложенного метода. Основопологающим этапом является разработка функции вознаграждения для ученика. Полученная награда должна помогать агенту-ученику обучаться. Чем больше ограничений (рекомендаций) будет иметь функция вознаграждения, тем быстрее и правильнее он научится. Дополнительным критерием является то, что агент получает награду, когда выполнит действие.

Для реализации предложенного метода необходимо выполнить следующие действия:

Наблюдение. В процессе обучения было определено, что влияние сигналов обратной связи по линейной и угловой скоростям недостаточно велико в сравнении с влиянием сигналов лидара. Для повышения влияния сигналов скорости на результат обучения в наблюдение были добавлены следующие величины: $\sin(v)$, $\cos(v)$, $\sin(w)$, $\cos(w)$, v'' , w'' . Всего шесть элементов, где некоторая нелинейная функция от линейной и угловой скоростей дополняет массив входных данных лидара и позволяет избежать повышения входных данных лидара. В качестве нелинейной функции выбраны функции синуса и косинуса. Наблюдаемые данные представлены в виде 33-элементного вектора, где 25 элементов содержат информацию с лидара, два элемента – сигналы обратной связи по линейной и угловой скоростям и шесть элементов – нелинейные функции синуса и косинуса, укоренения (v'') и углового ускорения (w'').

Критерии остановки. В ходе многочисленных экспериментов были определены критерии остановки. Остановка происходит в следующих случаях:

- сумма каждого четвертого элемента вектора линейной скорости 20-элементного вектора равна нулю (9);
- минимальное расстояние до препятствия по показаниям лидара меньше, чем радиус объекта управления ученика (физическая составляющая) (10);
- ученик нашел выход из лабиринта (11);
- евклидово расстояние между учеником и экспертом (C) больше, чем C_{\max} , где $C_{\max} = 0,3$ м (12):

$$\text{isdone} = \begin{cases} \sum_{i=1}^5 v_{4i}(t) = 0, & (9) \\ \text{lid}_{\min} \leq \text{Rob}_{\text{rad}}, & (10) \\ \begin{cases} x_{\text{real}} = x \\ y_{\text{real}} = y, \end{cases} & (11) \\ C > C_{\max}. & (12) \end{cases}$$

Вознаграждение. Для гарантии, что агент-ученик пройдет весь путь, введем переменную $dist$ – расстояние между начальной и текущей позициями ученика. Чем дальше ученик прошел за экспертом, тем больше он получит награду. Далее, для того чтобы ученик притормаживал при повороте либо в центре перекрестка и следовал алгоритму правой руки, в формуле (4) заменим переменную Rob_{rad} на переменную $width$ (ширина коридора). В результате получим равенство

$$lid_{rot} = \frac{1}{5} \cdot \sum_{k=-2}^2 lid_{i+k} - \frac{width}{2}. \quad (13)$$

Как будет показано далее на рис. 7, *a* и *b*, ученик передвигается по коридору, но не способен преодолеть тупик, поэтому было решено ввести переменную $impasse$, которая зависит от линейной и угловой скоростей:

$$impasse = \begin{cases} 0,5, & \text{если } \sum_{i=t-1}^t v_i = 0 \text{ и } \sum_{j=t-5}^t w_j \neq 0, \\ -0,5, & \text{если } \sum_{i=t-1}^t v_i = 0 \text{ и } \sum_{j=t-5}^t w_j = 0, \end{cases} \quad (14)$$

где v_i – значение линейной скорости, w_j – значение угловой скорости, t – текущий момент времени.

В результате получим награду по формуле

$$reward = dist + lid_{rot} + v + 0,0015 \cdot w^2 + 0,3 \cdot impasse + 10 \cdot finish. \quad (15)$$

Далее разработаем штраф для обучения. Чтобы побудить ученика смотреть в том же направлении, что и эксперт (рис. 7, *c*), согласно алгоритму правой руки, вычислим разницу между углами поворотов агентов и получим rot_{α} . Для сокращения евклидова расстояния между агентами воспользуемся переменной C (12). В итоге получим следующую формулу штрафа:

$$penalty = 0,8 \cdot rot_{\alpha} + C + 1,5 \cdot isdone. \quad (16)$$

Совместив награду (15) и штраф (16), запишем функцию вознаграждения для агента-ученика:

$$reward_{agent} = reward - penalty. \quad (17)$$

3. Результаты. На рис. 7 показаны промежуточные результаты, благодаря которым усовершенствована функция вознаграждения.

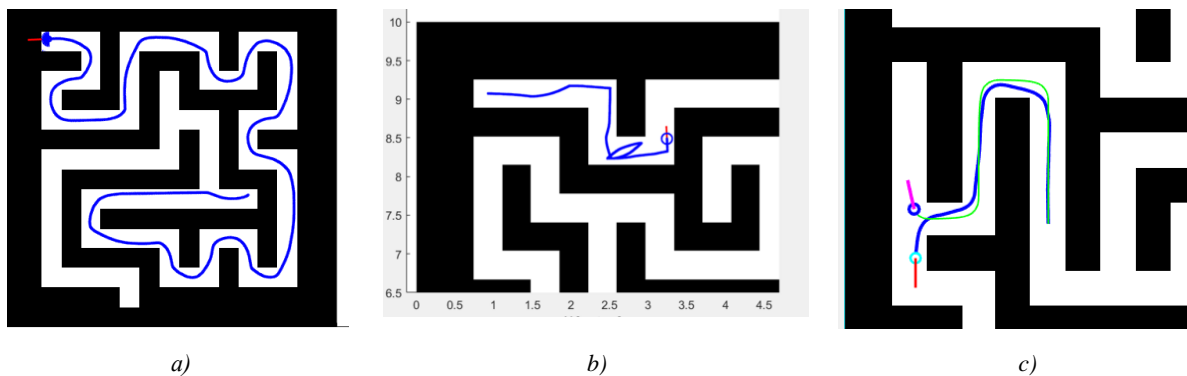


Рис. 7. Промежуточный результат обучения: *a*) результат обучения без блока обхода тупика (14); *b*) влияние угловой скорости ученика; *c*) результат обучения до добавления штрафа на угловую скорость (16) и решение одностороннего направления между агентами

Fig. 7. Intermediate learning result: a) Result of training without a deadlock bypass block (14); b) the influence of the angular velocity of the student; c) the result of training before adding the penalty to the angular velocity (16) and the decision of one-way direction between agents

На рис. 8 изображен один из моментов обучения, когда ученик следует за экспертом. На визуализации хорошо прослеживается, как разница между направлениями агентов и расстоянием между ними влияет на общую награду.

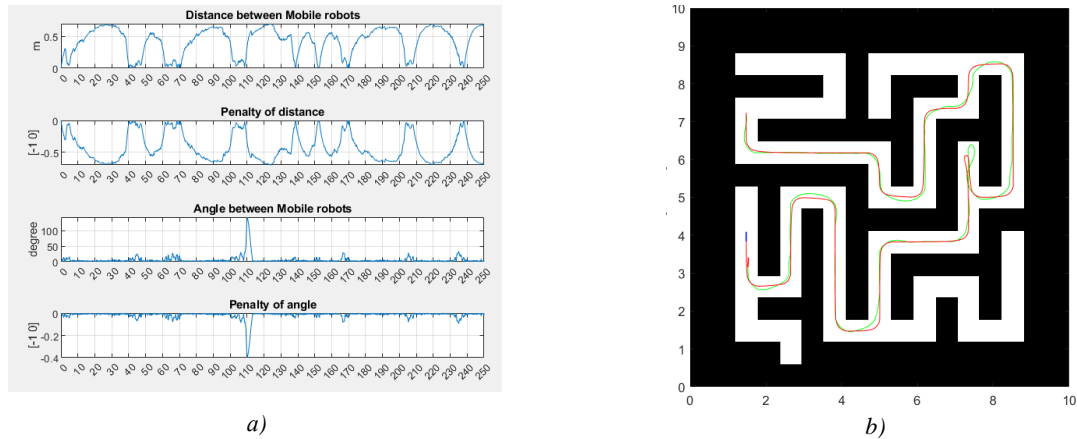


Рис. 8. Итоговый результат передвижения двух агентов после обучения: *a)* статистика передвижения агентов, где штрафная функция зависит от дистанции между роботами (тот же принцип действует на угол направленности между роботами); *b)* визуализация агентов, где красный и зеленый пути означают передвижения агента-эксперта и агента-ученика соответственно

Fig. 8. The final result of the two agents movement after training: a) statistics of agent movement, where the penalty function depends on the distance between robots, the same principle applies to the angle of orientation between robots; b) visualization of agents, where the red path and the green path represent the movement of the expert agent and the student agent respectively

Результаты экспериментов показывают, что спустя 2500 эпизодов ученик, следуя за экспертом, смог найти выход из лабиринта.

По завершении обучения агент-ученик принимает на себя управление, устраняя действия контроля агента-эксперта. Таким образом, остается только обученный ученик. Особенность обученного ученика заключается в том, что на участке, где эксперт следует правилу правой руки и потом направляется в тупик, ученик, предугадывая это, сокращает путь и направляется сразу к финишу. Это говорит о решении неоднозначной задачи для обучаемого агента-ученика. На рис. 9 места, где агент-ученик обходит данные участки, выделены красным.

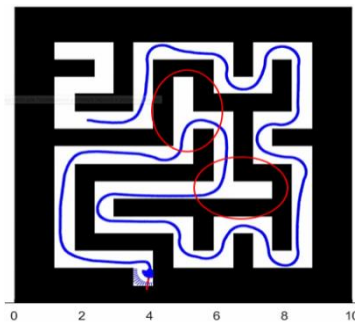


Рис. 9. Результат действия ученика после обучения без контроля эксперта

Fig. 9. The result of the student's action after training without the control of an expert

С помощью предложенной комбинации алгоритма обучения с подкреплением и алгоритма правой руки авторы вычислили, что ученик прошел лабиринт и нашел выход за 256 с.

4. Верификация обученного агента в различных средах. Обученный агент-ученик передвигается в незнакомой среде, где больше вариантов выбора повернуть направо или налево, количество стен и тупиковых ситуаций, вариаций ширины коридоров и стен.

Последовательно усложняя функцию вознаграждения, удалось решить задачу поиска пути в лабиринте. Для верификации обученного агента-ученика было решено проверить его поведение в незнакомой среде (рис. 10).

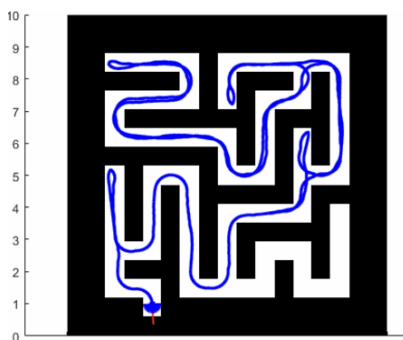


Рис. 10. Результаты моделирования поведения ученика в незнакомой среде

Fig. 10. Results of modeling student behavior in an unfamiliar environment

Результаты обучения показали, что совместное использование алгоритма обучения с подкреплением и алгоритма правой руки, а также разработанной функции вознаграждения побудило агента-ученика держаться центра коридора, совершать повороты и развороты по лабиринту, а также избегать столкновений.

Заключение. Предложенный авторами метод подражания предполагает совместное использование дискретного управления для эксперта и аналогового – для ученика. Несмотря на то что алгоритмы управления эксперта могут быть достаточно сложными и даже неизвестными ученику, путем анализа действий эксперта предложенный метод может обучить ученика сложному поведению. В результате подражательного обучения аналоговый агент приобретает новые паттерны поведения [24], что позволяет ему ориентироваться в неизвестной среде по правилу правой руки. Разработанный метод имеет большой потенциал для использования обучения с подкреплением в тех областях, где его раньше было сложно реализовать. В результате обученный ученик обобщает заданные действия и выводит собственные правила благодаря разработанной универсальной функции вознаграждения. Также преимущество предложенного метода состоит в том, что ученик передвигается с динамической скоростью, снижая ее при поворотах и достигая максимальной скорости на длинных дистанциях [25].

Вклад авторов. *Т. Ю. Ким* – разработка нового метода на базе глубокого обучения с подкреплением и алгоритма правой руки, программная реализация функции вознаграждения, верификация полученных результатов, проведение экспериментальных исследований с помощью классических алгоритмов обучения с подкреплением Q-learning и Актор-Критик. *Г. А. Прокопович* – постановка проблемы, разработка концепции статьи, обоснование актуальности работы, развитие ключевых целей и задач с помощью обучения с подкреплением, критический анализ работы.

Список использованных источников

1. Towards continuous control for mobile robot navigation: A reinforcement learning and slam based approach / К. А. А. Mustafa [et al.] // Intern. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – 2019. – Vol. 42. – P. 857–863. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-857-2019>

2. Truong, X. T. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model / X. T. Truong, T. D. Ngo // *IEEE Transactions on Automation Science and Engineering*. – 2017. – Vol. 14, no. 4. – P. 1743–1760. <https://doi.org/10.1109/TASE.2017.2731371>
3. Playing Atari with Deep Reinforcement Learning [Electronic resource] / V. Mhin [et al.]. – 2013. – Mode of access: <https://doi.org/10.48550/arXiv.1312.5602>. – Date of access: 20.06.2024.
4. Mastering the game of Go with deep neural networks and tree search / D. Silver [et al.] // *Nature*. – 2016. – Vol. 529, no. 7587. – P. 484–489.
5. Learning dexterous in-hand manipulation / M. Andrychowicz [et al.] // *The Intern. J. of Robotics Research*. – 2020. – Vol. 39, no. 1. – P. 3–20. <https://doi.org/10.1177/0278364919887447>
6. Emergence of Locomotion Behaviours in Rich Environments [Electronic resource] / N. Heess [et al.]. – 2017. – Mode of access: <https://doi.org/10.48550/arXiv.1707.02286>. – Date of access: 20.06.2024.
7. Autonomous vehicle perception: The technology of today and tomorrow / J. V. Brummelen [et al.] // *Transportation Research Part C: Emerging Technologies*. – 2018. – No. 86. – P. 384–406. <https://doi.org/10.1016/j.trc.2018.02.012>
8. Huang, W. Learning to drive via Apprenticeship Learning and Deep Reinforcement Learning [Electronic resource] / W. Huang, F. Braghin, Z. Wang. – 2020. – P. 1–7. – Mode of access: <https://doi.org/10.48550/arXiv.2001.03864>. – Date of access: 20.06.2024.
9. Robust AI driving strategy for autonomous vehicles / S. Nagesh Rao [et al.] // *AI-enabled Technologies for Autonomous and Connected Vehicles*. – Springer, 2022. – P. 161–212.
10. Sensor and sensor fusion technology in autonomous vehicles: A review / D. J. Yeong [et al.] // *Sensors*. – 2021. – Vol. 21, iss. 6. – P. 2140. <https://doi.org/10.3390/s21062140>
11. Kweon, J. Deep reinforcement learning for guidewire navigation in coronary artery phantom / J. Kweon, K. Kim, Ch. Lee // *IEEE Access*. – 2021. – Vol. 9. – P. 166409–166422. <https://doi.org/10.1109/ACCESS.2021.3135277>
12. An Algorithmic Perspective on Imitation Learning / T. Osa [et al.]. – Boston : Now publishers Inc., 2018. – 188 p.
13. Лонца, А. Алгоритмы обучения с подкреплением на Python / А. Лонца ; пер. с англ. А. А. Слинкина. – М. : ДМК Пресс, 2020. – 285 с.
14. Chella, A. Imitation learning and anchoring through conceptual spaces / A. Chella // *Applied Artificial Intelligence*. – 2007. – No. 21. – P. 343–359.
15. Kim, T. Automatic tuning of the motion control system of a mobile robot along a trajectory based on the reinforcement learning method / T. Kim, R. Prakapovich // *Communications in Computer and Information Science*. – Springer, Cham, 2022. – Vol. 1562. – P. 234–244. https://doi.org/10.1007/978-3-030-98883-8_17
16. Sutton, R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – 2nd ed. – London, England : The MIT Press, 2014. – 352 p.
17. Watkins, C. Q-learning / C. Watkins, P. Dayan // *Machine Learning*. – 1992. – Vol. 8, iss. 3–4. – P. 279–292.
18. Duan, J. M. Prior knowledge based Q-learning path planning algorithm / J. M. Duan, Q. L. Chen // *Electronics Optics & Control*. – 2019. – Vol. 26, iss. 9. – P. 29–33.
19. Sutton, R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – 2nd ed. – London, England : The MIT Press, 2014. – 338 p.
20. Rossi, F. Horizontal and vertical scaling of container-based applications using reinforcement learning / F. Rossi, M. Nardelli, V. Cardellini // 2019 IEEE 12th Intern. Conf. on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019. – Milan, 2019. – P. 329–338. <https://doi.org/10.1109/CLOUD.2019.00061>
21. PAC model-free reinforcement learning / A. L. Strehl [et al.] // *ICML'06: Proc. of the 23th Intern. Conf. on Machine Learning, Pittsburgh, Pennsylvania, USA, 25–29 June 2006*. – Pittsburgh, 2006. – P. 881–888. <https://doi.org/10.1145/1143844.114395>
22. Ravichandiran, S. Deep Reinforcement Learning with Python / S. Ravichandiran. – 2nd ed. – Packt Publishing, 2020. – 760 p.
23. Yu, Ch. Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units / Ch. Yu, G. Ren // *BMC Medical Informatics and Decision Making*. – 2020. – No. 20 (S3). – P. 1–8. <https://doi.org/10.1186/s12911-020-1120-5>
24. Imitation learning: progress, taxonomies and challenges [Electronic resource] / B. Zheng [et al.] // *IEEE Transactions on Neural Networks and Learning Systems*. – 2022. – P. 1–22. – Mode of access: <https://arxiv.org/abs/2106.12177>. – Date of access: 20.06.2024.
25. Ким, Т. Ю. Форсированное управление движением мобильного робота / Т. Ю. Ким, Г. А. Прокопович, А. А. Лобатый // *Информатика*. – 2022. – Т. 19, № 3. – С. 86–100. <https://doi.org/10.37661/1816-0301-2022-19-3-86-100>

References

1. Mustafa K. A. A., Botteghi N., Sirmacek B., Poel M., Stramigioli S. Towards continuous control for mobile robot navigation: A reinforcement learning and slam based approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019, vol. 42, pp. 857–863. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-857-2019>
2. Truong, X. T., Ngo T. D. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model. *IEEE Transactions on Automation Science and Engineering*, 2017, vol. 14, no. 4, pp. 1743–1760. <https://doi.org/10.1109/TASE.2017.2731371>
3. Mhin V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., ..., Riedmiller M. *Playing Atari with Deep Reinforcement Learning*, 2013. Available at: <https://doi.org/10.48550/arXiv.1312.5602> (accessed 20.06.2024).
4. Silver D., Huang A., Maddison C. J., Guez A., Sifre L., ..., Hassabis D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, vol. 529, no. 7587, pp. 484–489.
5. Andrychowicz M., Baker B., Chociej M., Józefowicz R., McGrew B., ..., Zaremba W. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2020, vol. 39, no. 1, pp. 3–20. <https://doi.org/10.1177/0278364919887447>
6. Heess N., Dhruva T. B., Sriram S., Lemmon J., Merel J., ..., Silver D. *Emergence of Locomotion Behaviours in Rich Environments*, 2017. Available at: <https://doi.org/10.48550/arXiv.1707.02286> (accessed 20.06.2024).
7. Brummelen J. V., O'Brien M., Gruyer D., Najjaran H. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 2018, no. 86, pp. 384–406. <https://doi.org/10.1016/j.trc.2018.02.012>
8. Huang W., Braghin F., Wang Z. *Learning to drive via Apprenticeship Learning and Deep Reinforcement Learning*, 2020, pp. 1–7. Available at: <https://doi.org/10.48550/arXiv.2001.03864> (accessed 20.06.2024).
9. Nagesh Rao S., Rahman Y., Ivanovic V., Jankovic M., Tseng E., ..., Filev D. Robust AI driving strategy for autonomous vehicles. *AI-enabled Technologies for Autonomous and Connected Vehicles*. Springer, 2022, pp. 161–212.
10. Yeong D. J., Velasco-Hernandez G., Barry J., Walsh J. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 2021, vol. 21, iss. 6, p. 2140. <https://doi.org/10.3390/s21062140>
11. Kweon J., Kim K., Lee Ch. Deep reinforcement learning for guidewire navigation in coronary artery phantom. *IEEE Access*, 2021, vol. 9, pp. 166409–166422. <https://doi.org/10.1109/ACCESS.2021.3135277>
12. Osa T., Pajarinen J., Neumann G., Bagnell J. A., Abbeel P., Peters J. *An Algorithmic Perspective on Imitation Learning*. Boston, Now publishers Inc., 2018, 188 p.
13. Lonza, A. *Reinforcement Learning Algorithms with Python*. Packt Publishing, 2019, 366 p.
14. Chella, A imitation learning and anchoring through conceptual spaces. *Applied Artificial Intelligence*, 2007, no. 21, pp. 343–359.
15. Kim T., Prakupovich R. Automatic tuning of the motion control system of a mobile robot along a trajectory based on the reinforcement learning method. *Communications in Computer and Information Science*. Springer, Cham, 2022, vol. 1562, pp. 234–244. https://doi.org/10.1007/978-3-030-98883-8_17
16. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*, 2nd edition. London, England, The MIT Press, 2014, 352 p.
17. Watkins C., Dayan P. Q-learning. *Machine Learning*, 1992, vol. 8, iss. 3–4, pp. 279–292.
18. Duan J. M., Chen Q. L. Prior knowledge based Q-learning path planning algorithm. *Electronics Optics & Control*, 2019, vol. 26, iss. 9, pp. 29–33.
19. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*, 2nd edition. London, England, The MIT Press, 2014, 338 p.
20. Rossi F., Nardelli M., Cardellini V. Horizontal and vertical scaling of container-based applications using reinforcement learning. *2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019*. Milan, 2019, pp. 329–338. <https://doi.org/10.1109/CLOUD.2019.00061>
21. Strehl A. L., Li L., Wiewiora E., Langford J., Littman M. L. PAC model-free reinforcement learning. *ICML'06: Proceeding of the 23th International Conference on Machine Learning. Pittsburgh, Pennsylvania, USA, 25–29 June 2006*. Pittsburgh, 2006, pp. 881–888. <https://doi.org/10.1145/1143844.114395>
22. Ravichandiran S. *Deep Reinforcement Learning with Python*, 2nd edition. Packt Publishing, 2020, 760 p.
23. Yu Ch., Ren G. Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC Medical Informatics and Decision Making*, 2020, no. 20 (S3), pp. 1–8. <https://doi.org/10.1186/s12911-020-1120-5>

24. Zheng B., Verma S., Zhou J., Tsang I., Chen F. Imitation learning: progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2022, pp. 1–22. Available at: <https://arxiv.org/abs/2106.12177> (accessed 20.06.2024).

25. Kim T. Yu., Prakapovich R. A., Lobatiy A. A. *Forced motion control of a mobile robot*. *Informatika [Informatics]*, 2022, vol. 19, no. 3, pp. 86–100 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-3-86-100>

Информация об авторах

Ким Татьяна Юрьевна, младший научный сотрудник, лаборатория робототехнических систем № 116, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: tatyana_kim92@mail.ru

<http://orcid.org/0000-0002-4126-6572>

Прокопович Григорий Александрович, кандидат технических наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: rprakapovich@robotics.by

<http://orcid.org/0000-0002-3412-9174>

Information about the authors

Tatyana Yu. Kim, Junior Researcher, Laboratory of Robotic Systems No. 116, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.

E-mail: tatyana_kim92@mail.ru

<http://orcid.org/0000-0002-4126-6572>

Ryhor A. Prakapovich, Ph. D. (Eng.), Assoc. Prof., The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.

E-mail: rprakapovich@robotics.by

<http://orcid.org/0000-0002-3412-9174>



УДК 004.932.2
<https://doi.org/10.37661/1816-0301-2024-21-3-63-79>

Оригинальная статья
Original Article

Гибридный контроль дефектов печатных плат

В. В. Венгеренко[✉], А. В. Инютин

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

Аннотация

Цели. Предлагается гибридный подход к задаче поиска и классификации дефектов печатных плат. Рассмотрены ключевые факторы и тенденции в проектировании и производстве печатных плат. Актуальность исследования определяется использованием новых материалов и технологий производства.

Методы. Для решения поставленной задачи применяется гибридный подход, основанный на алгоритме сравнения с эталоном и использовании семейства нейросетевых моделей обнаружения объектов YOLO.

Результаты. Проведено обучение моделей на публичных наборах изображений печатных плат с шестью классами дефектов, выполнена оценка точности общепринятыми метриками.

Заключение. Эксперименты показали, что нейросетевая архитектура YOLOv8 имеет высокую точность детекции дефектов, низкую чувствительность к качеству изображений, наличию надписей и графических объектов на печатной плате, но низкое качество обучающих выборок накладывает ограничения на использование только нейронных сетей для поиска дефектов. Предлагается гибридный подход для повышения качества контроля дефектов за счет применения разных методов в зависимости от оценки качества анализируемых изображений.

Ключевые слова: математическая морфология, контроль дефектов, печатные платы, детекторы объектов, ограничивающие прямоугольники, нейронные сети, сравнение с эталоном

Для цитирования. Венгеренко, В. В. Гибридный контроль дефектов печатных плат / В. В. Венгеренко, А. В. Инютин // Информатика. – 2024. – Т. 21, № 3. – С. 63–79.
<https://doi.org/10.37661/1816-0301-2024-21-3-63-79>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 22.07.2024

Подписана в печать | Accepted 22.08.2024

Опубликована | Published 30.09.2024

Hybrid inspection of printed board defects

Vadim V. Vengerenko[✉], Alexander V. Inyutin

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*

[✉]E-mail: vengerenko@lsi.bas-net.by

Abstract

Objectives. A hybrid approach to the problem of searching and classifying defects in printed circuit boards (PCB) is proposed. Key factors and trends in the design and production of PCBs are considered. The relevance of the study is determined by the use of new materials and production technologies.

Methods. A hybrid approach based on the algorithm of comparison with reference and the use of the YOLO family of neural network models for detecting objects is used to solve the problem.

Results. Models were trained on public sets of PCB images with six classes of defects, and their accuracy was assessed using generally accepted metrics.

Conclusion. Experiments have shown that the YOLOv8 neural network architecture has high accuracy of defect detection, low sensitivity to image quality, presence of text and graphic objects on the PCB, but the low quality of training datasets imposes restrictions on the use of only neural networks for defect detection. It is proposed to use a hybrid approach to improve the quality of defect inspection by applying different methods depending on the quality assessment of the analyzed images.

Keywords: mathematical morphology, inspection of defects, printed circuit boards, object detection, bounding boxes, neural networks, comparison with reference

For citation. Vengerenko V. V., Inyutin A. V. *Hybrid inspection of printed board defects*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 63–79 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-63-79>

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

Введение. Существует несколько ключевых факторов, которые стимулируют мировой рынок печатных плат в настоящее время и будут стимулировать его в течение следующих пяти лет^{1, 2}:

- развертывание беспроводных сетей 5G, что потребует создания большого количества сетевой инфраструктуры и беспроводного оборудования;
- разработка приборов, машин и устройств интернета вещей (IoT);
- разработка модулей управления двигателями и аккумуляторной батареей электромобилей (EV), а также электронных блоков и дисплеев всех типов автомобилей;
- внедрение усовершенствованных систем помощи водителю (Advanced Driver Assistance Systems, ADAS) для предотвращения столкновений и автоматической парковки. Такие системы используют набор камер, радаров, лидаров и ультразвуковых датчиков, а также сложные печатные платы высокой плотности;
- развертывание источников возобновляемой энергии, таких как солнечные фотоэлектрические системы и ветряные турбины;
- производство современной техники и систем связи для подразделений обороны и аэрокосмической промышленности, требующих специализированных, высокопроизводительных и высоконадежных печатных плат;
- разработка современных медицинских устройств для диагностики, терапии и дистанционного мониторинга пациентов;
- широкое развертывание дата-центров с высокопроизводительными серверами и коммутаторами для облачных вычислений, которые требуют самых современных печатных плат.

¹What is the PCB Market Situation Now? (Update 2023) [Electronic resource]. – Mode of access: <https://www.raypcb.com/pcb-market/>. – Date of access: 07.05.2024.

²The Global Printed Circuit Board Market: Key Insights, Forecasts, & Growth Outlook [Electronic resource]. – Mode of access: <https://www.mktpcb.com/pcb-industry-statistics-trends-infographic/>. – Date of access: 07.05.2024.

Системы контроля топологии печатных плат могут использовать как контактные, так и бесконтактные методы [1–7]. Системы, основанные на контактных методах, касаются щупами контактных площадок и проверяют соответствие электрических связей между ними. В результате обнаруживаются обрывы (open) и короткие замыкания (short). С помощью контактного метода принципиально не могут быть найдены такие дефекты, как островки меди (spurious copper), выступы (spur) и вырывы (mousebite) на дорожках, проколы (pin-hole), отклонения минимальной ширины проводника (conductor too thin) и расстояния между проводниками (conductor too close). Увеличение сложности печатной платы приводит к значительному усложнению контроля. Бесконтактные методы контроля печатных плат делятся на оптические, рентгеновские, ультразвуковые, термические и др. Зачастую при производстве печатных плат применяется сразу несколько методов контроля, например контактный, оптический и рентгеновский.

В настоящее время за рубежом выпускаются автоматизированные системы контроля качества печатных плат. Их ведущими производителями сегодня являются компании Cognex Corporation, Gardien Services Inc., Manncorp Inc., Nordson YESTECH Inc., Omron Electronics LLC, Vision Engineering Inc, ViTrox Corp Bhd и Iбiden.

Актуальность задачи поиска дефектов печатных плат обусловлена следующими тенденциями в их проектировании и производстве [2]:

- использованием соединений высокой плотности (High-Density Interconnect, HDI), что выражается в более плотной разводке печатных плат и уменьшении размера контактных площадок. Отверстия для соединений между слоями могут иметь размер менее 150 мк. Однако существует мнение, что уменьшение шага выводов компонентов фактически достигло своих пределов: 0,65 и 0,3 мм для компонентов, выполненных по технологии PBGA (Plastic Ball Grid Array) и QFP (Quad Flat Package) соответственно. При этом минимальный шаг выводов компонентов FBGA (Fine-pitch BGA) равен 0,3 мм и может уменьшиться до 0,15 мм;
- сочетанием гибких схем и жестких секций на одной печатной плате, которое позволяет осуществлять трехмерную компоновку внутри устройств;
- производством растягивающихся схем, основанных на новых проводящих материалах и методах изготовления, которые позволяют разрабатывать гибкие носимые устройства;
- использованием низкотемпературной керамики совместного обжига (Low Temperature Co-fired Ceramic, LTCC) для высокотемперостойких печатных плат, которые могут работать в условиях агрессивной среды в оборонных, аэрокосмических и автомобильных системах;
- выпуском печатных плат с изолированной металлической подложкой (Insulated Metal Substrate, IMS) для эффективного отвода тепла от мощных компонентов;
- внедрением встроенных пассивных элементов (например, конденсаторов) внутрь печатной платы, что дает возможность более эффективно интегрировать компоненты по сравнению с поверхностным монтажом;
- применением аддитивной струйной или аэрозольной печати проводников, позволяющей упростить быстрое прототипирование и настройку печатных плат в ходе разработки;
- расположением выводов под корпусом компонентов;
- интеграцией искусственного интеллекта и машинного обучения для обнаружения дефектов и внедрением принципов Индустрии 4.0 для анализа данных, спрос на высокоскоростной и точный контроль.

В статье предлагается гибридный подход к оптическому контролю дефектов печатных плат, использующему сравнение с эталонным изображением [10] и нейросетевые архитектуры YOLOv8.

Анализ подходов к детекции объектов. Модели детекции (обнаружения) объектов на основе глубокого обучения с использованием сверточных нейронных сетей (СНС, или Convolutional Neural Network, CNN) и трансформеров сейчас играют ключевую роль в развитии этой области. Такие модели могут предоставить существенную информацию для семантического понимания изображений и видео. Среди многочисленных примеров их применения стоит отметить поддержку беспилотных автомобилей для безопасной навигации в пробках, распознавание лиц, анализ поведения человека, медицинскую визуализацию, робототехнику, общие методы обработки изображений, приложения дистанционного зондирования.

Основой детекторов объектов служат СНС. Основная цель такой сети – извлекать особенности из изображений перед отправкой их на дальнейшие этапы, например локализацию. Существует несколько стандартных СНС, используемых детекторами объектов, включая VGGNets, ResNets и EfficientNets, которые предварительно обучены для задач классификации.

Поля привязки (anchor boxes) представляют собой заранее определенную коллекцию ограничивающих прямоугольников (bounding boxes) с выбранной шириной и высотой, которые отражают ширину и высоту объектов в наборе обучающих данных. Разумеется, они также включают в себя различные соотношения сторон и масштабы, найденные в наборе данных. При обнаружении заранее определенные поля привязки располагаются на изображении в виде мозаики. Более того, на каждом изображении постоянно предлагаются одни и те же области привязки. Вместо прогнозирования прямоугольников сеть прогнозирует вероятность и другие атрибуты, такие как фон, пересечение над объединением (Intersection over Union, IoU) и смещения для каждого мозаичного поля привязки. Она возвращает уникальную коллекцию прогнозов для каждого установленного поля привязки. Использование полей привязки позволяет сети обнаруживать несколько объектов, объекты разных масштабов и перекрывающиеся объекты [9]. При обнаружении объектов детекторы на основе привязки определяют поля привязки в каждой позиции на карте объектов. Сеть прогнозирует вероятность появления объектов в каждом поле привязки, а затем подгоняет размер полей привязки в соответствии с объектом. Детекторы с использованием привязки обычно делятся на двухэтапные на основе предложений и одностадийные без предложений. Привязки служат эталонами для регрессии и кандидатами на классификацию для прогнозирования предложений для двухэтапных детекторов и окончательных ограничивающих прямоугольников для одностадийных детекторов.

Алгоритмы обнаружения объектов на основе областей были одними из наиболее широко используемых методов обнаружения объектов на изображениях. Двухэтапные методы являются производными от методов R-CNN (Region-based CNN), которые извлекают области интереса (Regions of Interest, RoI) с использованием метода выборочного поиска, а затем классифицируют эти области и выполняют их регрессионный анализ. Faster R-CNN является наиболее известным двухэтапным детектором на основе привязки. Впоследствии было представлено множество моделей для улучшения его характеристик, например Mask R-CNN.

Одностадийные детекторы на основе привязки характеризуются прежде всего эффективностью вычислений и временем выполнения. Такие детекторы напрямую классифицируют предопределенные поля привязки и выполняют их регрессионный анализ вместо использования областей интереса. SSD (Single Shot MultiBox Detector) стал первым известным одностадийным детектором объектов, основанным на привязке. Главной проблемой, с которой сталкиваются одностадийные детекторы на основе привязки, является дисбаланс между положительными и отрицательными образцами. Для решения возникшей проблемы было реализовано несколько подходов и механизмов, таких как уточнение и сопоставление привязок, обучение с нуля, многоуровневое объединение контекстной информации, обогащение и выравнивание признаков, а также разработаны новые функции потерь и архитектуры.

YOLOv8 – это современная модель классификации изображений, сегментации экземпляров и обнаружения объектов без привязки, разработанная Ultralytics [9]. Модели YOLOv8 можно обучать на больших наборах данных и запускать на различных аппаратных платформах. Одной из ключевых особенностей YOLOv8 является расширяемость. YOLOv8 поддерживает все предыдущие версии YOLO, что позволяет легко переключаться между различными версиями и сравнивать их производительность. YOLO захватывает полное изображение в одном экземпляре, а затем прогнозирует координаты ограничивающих прямоугольников и вероятности классов одной сетью за одну оценку.

Развитием идеи обнаружения объектов является введение угла для ограничивающих прямоугольников, чтобы учитывать ориентацию искомого объекта на изображении. Этот подход называется ориентированным обнаружением объектов. Он позволяет точнее находить объекты

на изображении³. Ограничивающие прямоугольники в таком случае также называются ориентированными (Oriented Bounding Boxes, OBB). Для моделей YOLOv8 OBB в файлах разметки один ориентированный ограничивающий прямоугольник описывается меткой класса и нормированными координатами всех вершин⁴.

Наборы изображений для экспериментов. Для проведения экспериментов были выбраны такие общедоступные наборы изображений, как DeepPCB⁵ и синтетический набор изображений дефектов печатных плат PCB Defect Dataset, опубликованный Открытой лабораторией по взаимодействию человека и робота Пекинского университета (The Open Lab on Human Robot Interaction of Peking University)⁶.

Набор изображений DeepPCB содержит 1500 пар бинарных изображений, каждая из которых состоит из эталонного изображения без дефектов и выровненного тестируемого изображения с аннотациями, включающими позиции шести наиболее распространенных типов дефектов печатных плат: обрыв, короткое замыкание, выступ, вырыв, прокол и островок. Изображения получены с помощью ПЗС-матрицы с линейным сканированием и разрешением около 48 пикселей на 1 мм. Изображения, размер которых составляет 640×640 пикселей, получены путем разделения исходных эталонов и тестируемых изображений размером около 16 000×16 000 пикселей и выровнены с помощью техники совмещения изображений.

На одном изображении присутствуют несколько дефектов разных классов. Пример тестируемого изображения и соответствующего ему эталона из набора данных DeepPCB приведен на рис. 1.

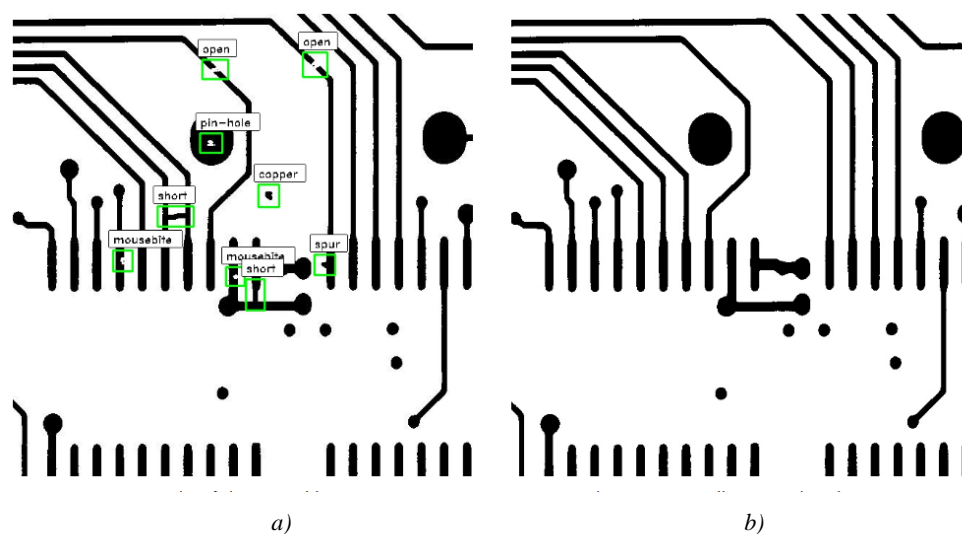


Рис. 1. Пример тестируемого изображения (a) и соответствующего ему эталона (b)

Fig. 1. Example of a tested image (a) and its corresponding reference (b)

Для проведения экспериментов с моделью YOLOv8n использован указанный набор данных уже с примененными к нему техниками аугментации⁷, которые включают следующие операции:

³OBB – Ultralytics YOLO Docs [Electronic resource]. – Mode of access: <https://docs.ultralytics.com/tasks/obb/>. – Date of access: 11.07.2024.

⁴Oriented Bounding Box (OBB) Datasets Overview – Ultralytics YOLO Docs [Electronic resource]. – Mode of access: <https://docs.ultralytics.com/datasets/obb/>. – Date of access: 11.07.2024.

⁵A PCB defect dataset [Electronic resource]. – Mode of access: <https://github.com/tangsanli5201/DeepPCB>. – Date of access: 11.07.2024.

⁶PCB Defects [Electronic resource]. – Mode of access: <https://www.kaggle.com/datasets/akhatova/pcb-defects/data>. – Date of access: 11.07.2024.

⁷DeepPCB Image Dataset [Electronic resource]. – Mode of access: <https://universe.roboflow.com/tack-hwa-wong-zak5u/deeppcb-4dhir/dataset/5>. – Date of access: 11.07.2024.

- вертикальный и горизонтальный поворот;
- поворот на 90° по часовой стрелке, против часовой стрелки и вверх ногами;
- обрезку изображения с минимальным масштабированием 0 % и максимальным масштабированием 20 %;
- поворот между -15° и $+15^\circ$;
- сдвиг $\pm 15^\circ$ по горизонтали, $\pm 15^\circ$ по вертикали.

После аугментации набор разделен на обучающую, валидационную и тестовую выборки, которые включают в себя 3150, 150 и 300 изображений соответственно. Для каждого изображения есть файл разметки в формате .txt, который содержит информацию о дефектах на изображении (номер класса и нормированные координаты ограничивающего прямоугольника).

Второй набор данных PCB Defect Dataset содержит 1386 изображений и координаты созданных в графическом редакторе дефектов шести типов, таких как прокол, вырыв, обрыв, короткое замыкание, выступ и островок. Уникальных изображений в данном наборе 693, а остальные получены путем аугментации, заключающейся в повороте этих уникальных изображений на различные углы относительно центра изображения (рис. 2).

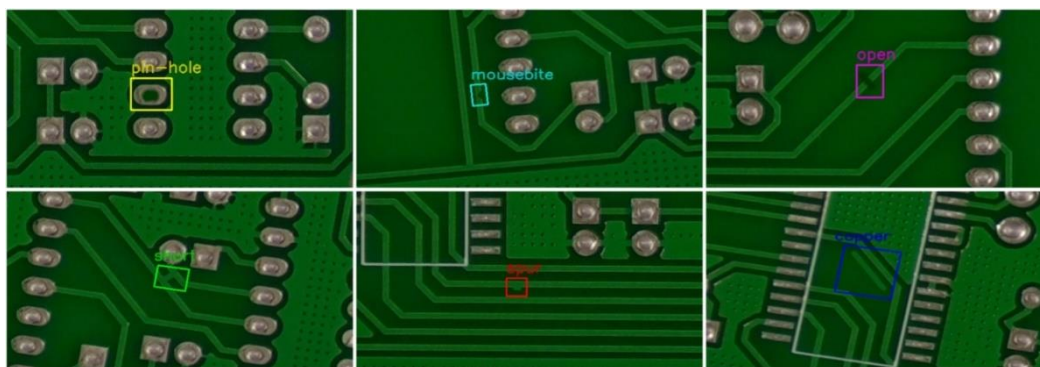


Рис. 2. Примеры изображений дефектов из набора данных PCB Defect Dataset

Fig. 2. Defect image samples from the PCB Defect Dataset

Для проведения экспериментов с моделью YOLOv8n OBB набор данных PCB Defect Dataset разбивается на обучающую, валидационную и тестовую выборки в соотношении 8:1:1 соответственно. Перед разбиением он перемешивается, поскольку уникальные и аугментированные изображения находятся в отдельных папках и при этом упорядочены по типу дефекта. Для каждого изображения существует файл разметки в формате .txt, который содержит информацию о дефектах на изображении в формате OBB, описанном выше. Следует отметить, что на одном изображении может присутствовать несколько дефектов одного класса.

Распределение дефектов в наборах данных изображено на рис. 3.

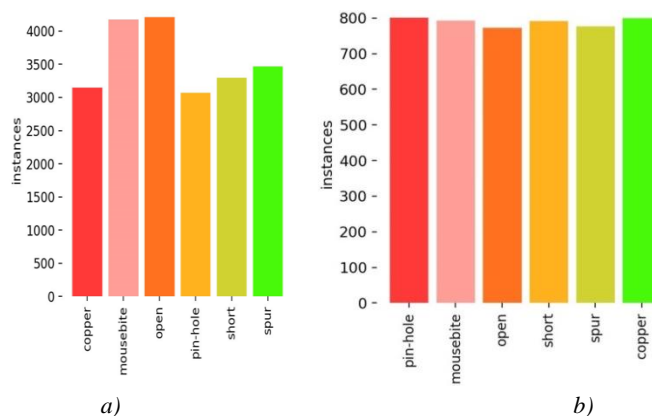


Рис. 3. Распределение дефектов в наборах данных: а) DeepPCB; б) PCB Defect Dataset

Fig. 3. Defect distribution in datasets: a) DeepPCB; b) PCB Defect Dataset

Метрики оценки. Для оценки эффективности модели использованы следующие широко применимые к различным моделям обнаружения объектов показатели⁸:

– средняя точность (Average Precision, AP), которая вычисляет площадь под кривой точности (Precision) и полноты (Recall), предоставляя одно значение, которое отражает производительность модели по точности и полноте;

– среднее значение средней точности (mean AP, mAP), расширяющее концепцию AP путем вычисления средних значений AP для нескольких классов объектов, что обеспечивает всестороннюю оценку производительности модели в сценариях обнаружения объектов с несколькими классами;

– точность, которая определяет долю истинно положительных результатов среди всех положительных прогнозов, оценивая способность модели избегать ложноположительных результатов;

– полнота, вычисляющая долю истинно положительных результатов среди всех фактически положительных результатов, измеряя способность модели обнаруживать все экземпляры класса.

Результаты также содержат разбитые по классам показатели эффективности. Такая детальная информация позволяет понять, насколько хорошо модель работает для каждого конкретного класса, особенно в наборах данных с разнообразными категориями объектов. Для каждого класса в наборе данных предоставляются следующие показатели:

– количество изображений в валидационном наборе, содержащих класс объектов;

– количество появлений класса во всех изображениях в валидационном наборе;

– точность обнаруженных объектов, указывающая, сколько было правильных обнаружений;

– полнота, означающая способность модели идентифицировать все экземпляры объектов на изображениях;

– mAP50 – среднее значение средней точности, рассчитанное при пороге IoU со значением 0,50 и учитывающее только «легкие» обнаружения;

– mAP50-95 – среднее значение средней точности, рассчитанное при различных пороговых значениях IoU в диапазоне от 0,50 до 0,95, что дает полное представление о производительности модели при различных уровнях сложности обнаружения.

Помимо числовых показателей, существуют также визуальные результаты, которые могут обеспечить более интуитивное понимание производительности модели⁹:

– кривая «точность-полнота» – интегральная визуализация любой задачи классификации, которая демонстрирует компромисс между точностью и полнотой при различных пороговых значениях, что особенно важно при работе с несбалансированными классами;

– кривая точности, являющаяся графическим представлением значений точности при различных пороговых значениях, которое помогает понять, как меняется точность при изменении порога;

– кривая полноты, иллюстрирующая, как значения полноты изменяются при различных пороговых значениях;

– матрица ошибок, обеспечивающая подробное представление результатов, демонстрируя количество истинно положительных, истинно отрицательных, ложноположительных и ложноотрицательных результатов для каждого класса;

– нормализованная матрица ошибок, являющаяся нормализованной версией матрицы ошибок и представляющая данные в пропорциях, а не в необработанных значениях.

Постановка экспериментов. Для первой части экспериментов использовался набор данных DeepPCB. Модель YOLOv8 размером nano, предобученная на наборе данных COCO, была дообучена и протестирована на описанном наборе изображений. Обучение выполнено в среде Google Colab с использованием GPU Tesla T4. Параметры обучения модели YOLOv8n: количество эпох – 50, размер пакета – 16, размер изображения – 640×640, метод оптимизации – AdamW, вес компонента потерь прямоугольника в функции потерь – 7,5.

⁸YOLO Performance Metrics [Electronic resource]. – Mode of access: <https://docs.ultralytics.com/guides/yolo-performance-metrics/>. – Date of access: 11.07.2024.

⁹URL: <https://docs.ultralytics.com/guides/yolo-performance-metrics/>

В первой части экспериментов алгоритм сравнения с эталоном использует бинаризацию методом Оцу и морфологические операции отмыкания и замыкания с круглым структурирующим элементом размерами 3×3 и 5×5 . Алгоритм сравнения с эталоном использует семантический фильтр, отсеивающий дефекты площадью менее 20 пикселей.

Графики обучения показаны на рис. 4; кривая «точность-полнота», полученная в процессе обучения, – на рис. 5.

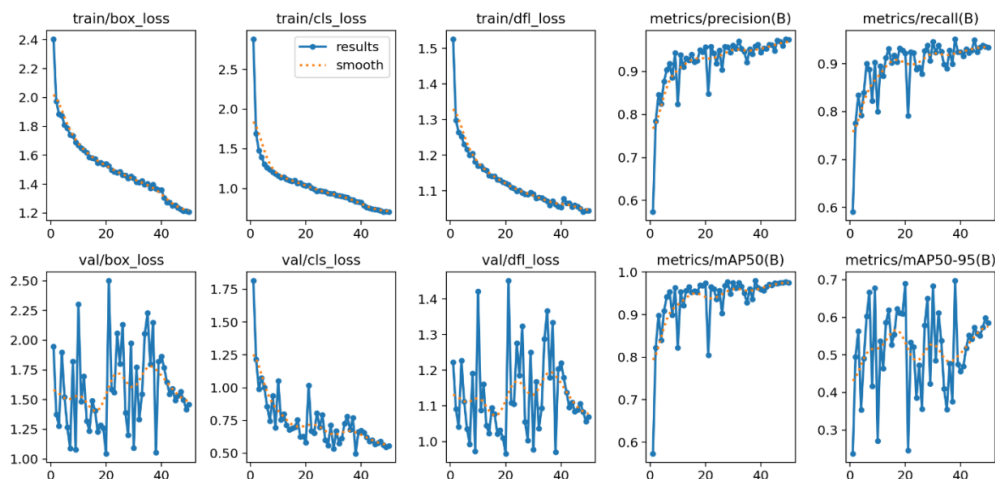


Рис. 4. Графики обучения нейронной сети YOLOv8n

Fig. 4. Training graphs for the YOLOv8n neural network

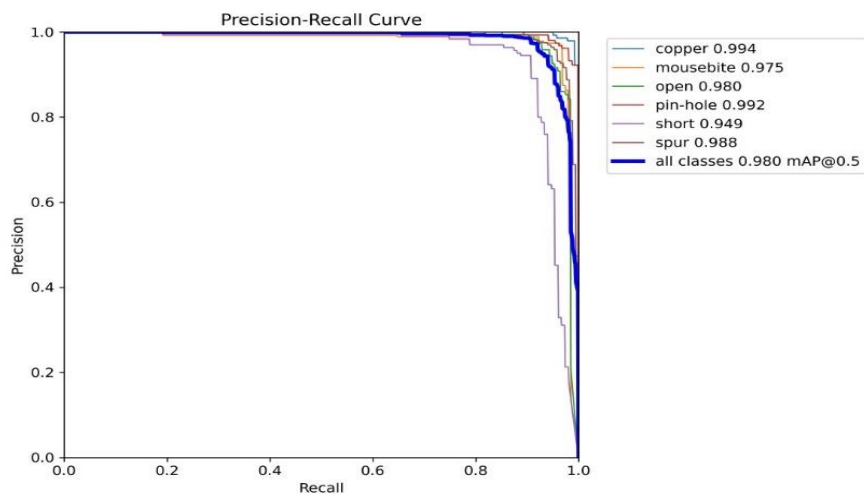


Рис. 5. Кривая «точность-полнота» на этапе обучения

Fig. 5. Precision-recall curve at the training stage

На рис. 6 показаны дефекты (класс и степень уверенности), найденные на изображениях из валидационной выборки.

Результаты работы алгоритма сравнения с эталоном в отдельности и в сочетании с моделью YOLOv8n получены с помощью исходного неаугментированного набора данных. При сочетании алгоритма сравнения с эталоном и модели YOLOv8n к предсказаниям данной модели добавляются предсказания, не выполненные ею, но выполненные алгоритмом сравнения с эталоном.

На рис. 7 представлены дефекты, найденные с помощью гибридного алгоритма, основанного на сравнении с эталоном и использовании нейросетевой модели YOLOv8n. Дефекты обозначены различными цветами: обрыв – розовым, короткое замыкание – зеленым, выступ – красным, вырыв – голубым, прокол – желтым и островок – синим.

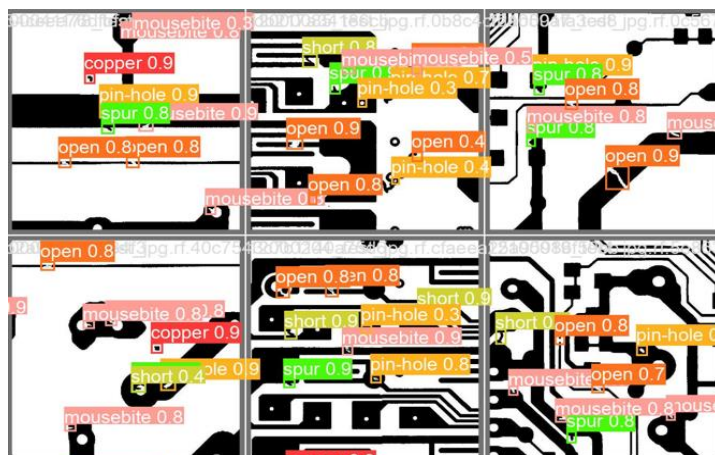


Рис. 6. Дефекты, найденные на изображениях из валидационной выборки
Fig. 6. Defects found in images from the validation sample

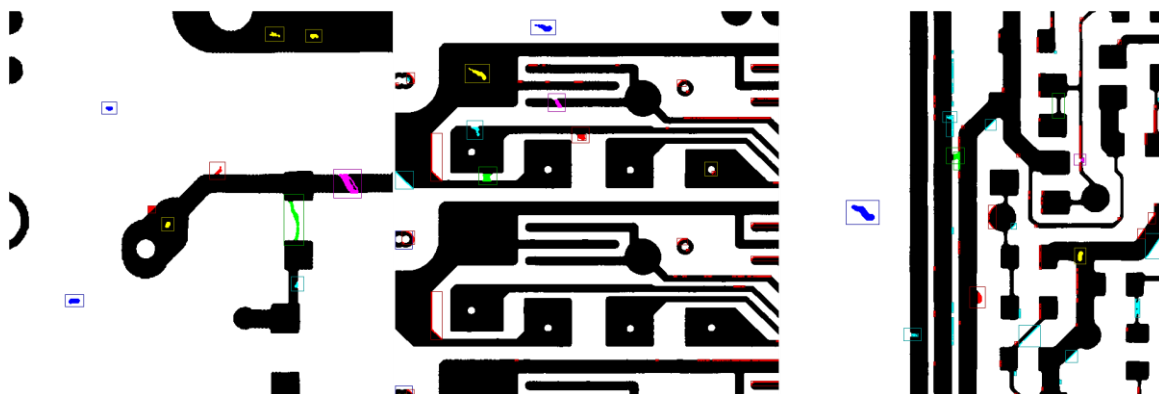


Рис. 7. Дефекты, найденные с помощью гибридного алгоритма
Fig. 7. Defects found using a hybrid algorithm

Результаты детекции дефектов (число дефектов, метрики оценки производительности модели), полученные на тестовом наборе данных из 300 изображений с помощью обученной модели YOLOv8n, представлены в табл. 1.

Таблица 1
Результаты детекции дефектов, полученные на тестовом наборе с помощью обученной модели YOLOv8n

Table 1
Results of defect detection obtained on the test set using the trained YOLOv8n model

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	mAP50	mAP50-95
Copper	282	0,983	0,986	0,992	0,858
Mousebite	371	0,933	0,945	0,977	0,698
Open	338	0,906	0,956	0,972	0,654
Pin-hole	309	1	0,964	0,995	0,831
Short	249	0,953	0,928	0,957	0,628
Spur	295	0,975	0,963	0,989	0,705
Все дефекты	1844	0,958	0,957	0,98	0,729

Отчет о классификации дефектов и матрица ошибок, полученные на полном наборе данных DeepPCB с помощью обученной модели YOLOv8n, представлены в табл. 2 и на рис. 8.

Таблица 2

Отчет о классификации, полученный на исходном наборе данных с помощью модели YOLOv8n

Table 2

Classification report obtained on the original dataset using the YOLOv8n model

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	F1-score	Accuracy
Pin-hole	1554	0,880 73	0,926 64	0,903 10	—
Mousebite	2081	0,764 01	0,936 57	0,841 54	
Open	2015	0,916 05	0,980 15	0,947 02	
Short	1588	0,933 71	0,948 99	0,941 29	
Spur	1681	0,883 48	0,960 74	0,920 49	
Copper	1506	0,837 85	0,984 73	0,905 37	
Все дефекты	11 768	Macro avg			0,847 13
		0,745 12	0,819 69	0,779 83	
		Weighted avg			
		0,767 68	0,847 13	0,804 60	

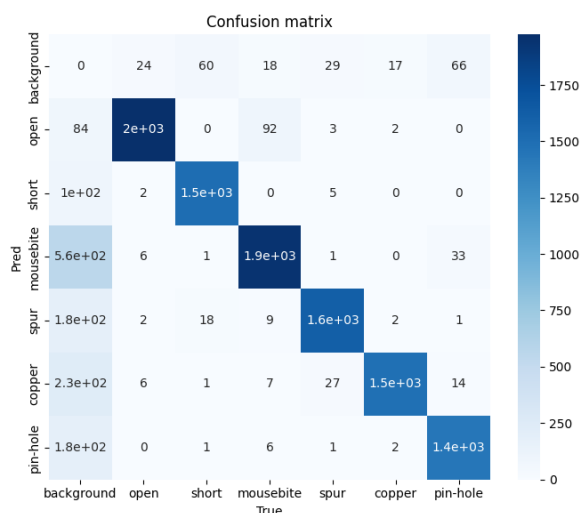


Рис. 8. Матрица ошибок для модели YOLOv8n, полученная на наборе данных DeepPCB

Fig. 8. Confusion matrix for the YOLOv8n model obtained on the DeepPCB dataset

Результаты работы алгоритма сравнения с эталоном и гибридного алгоритма на наборе данных DeepPCB представлены в табл. 3 и 4. Матрицы ошибок изображены на рис. 9.

Таблица 3

Отчет о классификации алгоритма сравнения с эталоном на наборе данных DeepPCB

Table 3

Classification report of the reference-based PCB inspection algorithm on the DeepPCB dataset

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	F1-score	Accuracy
Pin-hole	1502	0,926 20	0,927 43	0,926 81	—
Mousebite	1971	0,628 37	0,757 48	0,686 91	
Open	1956	0,934 98	0,852 76	0,891 98	
Short	1518	0,982 81	0,866 27	0,920 87	
Spur	1637	0,618 12	0,925 47	0,741 19	
Copper	1480	0,981 21	0,987 84	0,984 51	
Все дефекты	11 712	Macro avg			0,755 29
		0,724 52	0,759 61	0,736 04	
		Weighted avg			
		0,718 44	0,755 29	0,730 79	

Таблица 4
Отчет о классификации гибридного алгоритма на наборе данных DeepPCB

Table 4
Classification report of a hybrid algorithm on the DeepPCB dataset

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	F1-score	Accuracy
Pin-hole	1662	0,862 88	0,969 31	0,913 01	—
Mousebite	2344	0,620 17	0,931 31	0,744 54	
Open	2189	0,895 35	0,977 16	0,934 47	
Short	1685	0,926 56	0,958 46	0,942 24	
Spur	1964	0,647 79	0,978 62	0,779 56	
Copper	1558	0,834 14	0,994 22	0,907 17	
Все дефекты	14 359	Macro avg			0,767 39
		0,683 84	0,829 87	0,745 86	
		Weighted avg			
		0,625 45	0,767 39	0,685 30	

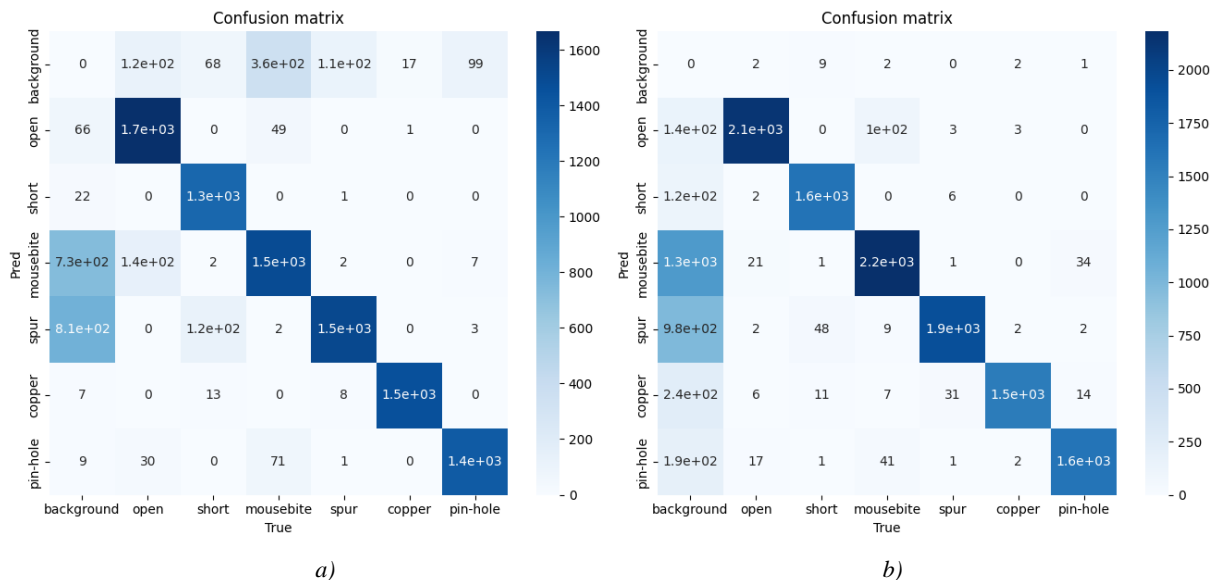


Рис. 9. Матрицы ошибок, полученные для набора DeepPCB: а) для алгоритма сравнения с эталоном; б) гибридного алгоритма

Fig. 9. Confusion matrices obtained for the DeepPCB set: a) for the reference-based PCB inspection algorithm; b) the hybrid algorithm

Результаты экспериментов показывают, что минимальное количество пропущенных дефектов, а именно 16, достигается при использовании гибридного алгоритма. Нейросетевая модель демонстрирует высокую точность детекции дефектов печатных плат (метрика mAP50 равна 0,98 на тестовом наборе данных). При этом на тестовом наборе для этой модели доля ложноположительных ошибок составляет 11,9 % относительно общего количества дефектов, а доля ложноотрицательных ошибок – 2,8 %. Относительно всех дефектов наихудшее значение доли ложноположительных ошибок соответствует классу «обрыв» и составляет 21,9 %, а наихудшее значение доли ложноотрицательных ошибок соответствует классу «короткое замыкание» и составляет 5,2 %.

Для второй части экспериментов использовался набор изображений дефектов печатных плат PCB Defect Dataset.

Модель YOLOv8 размером папо, предобученная на наборе данных DOTAv1¹⁰, была дообучена и протестирована на описанном наборе изображений. Обучение выполнено с использованием GPU NVIDIA GeForce RTX 3060. Некоторые параметры обучения модели YOLOv8n OBB: количество эпох – 200, размер пакета – 16, метод оптимизации – Adam, вес компонента потерь прямоугольника в функции потерь – 7,5.

В данной части экспериментов алгоритм сравнения с эталоном использует глобальную бинаризацию – диапазону значения RGB от (0, 20, 0) до (45, 85, 45) соответствует белый цвет, а остальному диапазону – черный. Для совмещения тестируемого изображения и эталона используются масштабно-инвариантная трансформация признаков (Scale Invariant Feature Transform, SIFT)¹¹, сопоставление дескрипторов методом Brute-Force¹² и интерполяция методом ближайшего соседа¹³. Морфологические операции отмыкания-замыкания с круглым структурирующим элементом размером 5×5 применяются для сглаживания результатов бинаризации и совмещения.

Графики обучения показаны на рис. 10. Кривая «точность-полнота», полученная в процессе обучения, изображена на рис. 11.

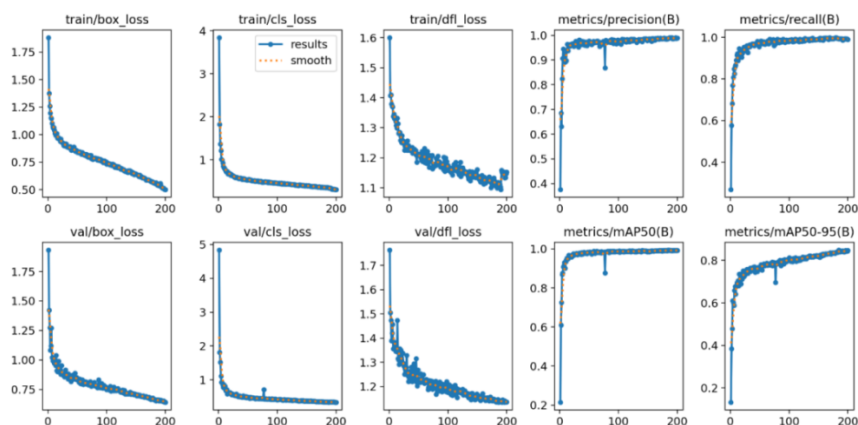


Рис. 10. Графики обучения нейронной сети YOLOv8n OBB

Fig. 10. Training graphs for the YOLOv8n OBB neural network

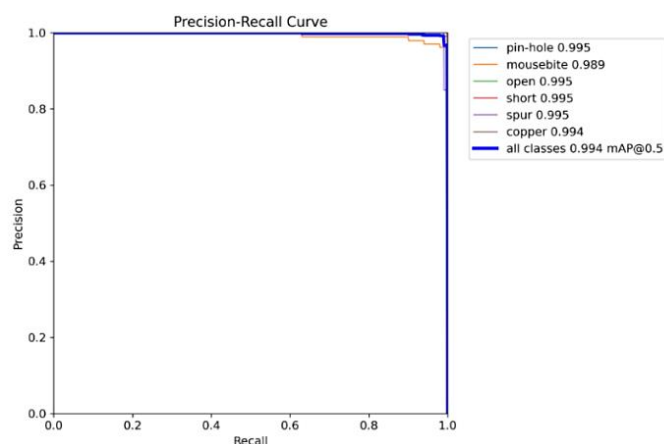


Рис. 11. Кривая «точность-полнота» на этапе обучения

Fig. 11. Precision-recall curve at the training stage

¹⁰DOTA. A Large-Scale Benchmark and Challenges for Object Detection in Aerial Images [Electronic resource]. – Mode of access: <https://captain-whu.github.io/DOTA/index.html>. – Date of access: 11.07.2024.

¹¹OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform) [Electronic resource]. – Mode of access: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html. – Date of access: 11.07.2024.

¹²OpenCV: Feature Matching [Electronic resource]. – Mode of access: https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html. – Date of access: 11.07.2024.

¹³OpenCV: Geometric Image Transformations [Electronic resource]. – Mode of access: https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html. – Date of access: 11.07.2024.

На рис. 12 показаны дефекты (класс и степень уверенности), найденные на изображениях из валидационной выборки.

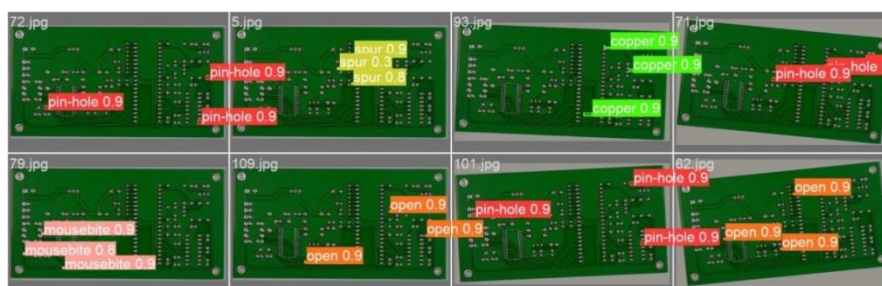


Рис. 12. Дефекты, найденные на изображениях из валидационной выборки

Fig. 12. Defects found in images from the validation sample

Результаты детекции дефектов, полученные на тестовом наборе данных из 139 изображений с помощью обученной модели YOLOv8n, представлены в табл. 5.

Таблица 5

Результаты детекции дефектов, полученные на тестовой выборке набора PCB Defect Dataset с помощью модели YOLOv8n

Table 5

Results of defect detection obtained on a test sample of the PCB Defect Dataset using the YOLOv8n model

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	mAP50	mAP50-95
Pin-hole	97	1	1	0,995	0,891
Mousebite	93	0,989	1	0,994	0,84
Open	94	1	0,989	0,994	0,848
Short	101	1	0,99	0,995	0,857
Spur	99	0,979	0,949	0,969	0,803
Copper	102	1	1	0,995	0,828
Все дефекты	586	0,995	0,988	0,99	0,844

Отчет о классификации дефектов и матрица ошибок, полученные на тестовой выборке из 139 изображений набора PCB Defect Dataset с помощью обученной модели YOLOv8n, представлены в табл. 6 и на рис. 13, а.

Таблица 6

Отчет о классификации, полученный на тестовой выборке исходного набора данных с помощью модели YOLOv8n

Table 6

Classification report obtained on a test sample of the original dataset using the YOLOv8n model

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	F1-score	Accuracy
Copper	102	1	1	1	-
Mousebite	93	0,989 36	1	0,994 65	
Open	94	1	0,989 36	0,994 65	
Pin-hole	97	1	1	1	
Short	101	1	0,9010	0,995 03	
Spur	99	0,979 17	0,949 49	0,964 10	
Все дефекты	586	Macro avg			0,988 05
		0,852 65	0,846 99	0,849 78	
		Weighted avg			
		0,994 79	0,988 05	0,991 37	

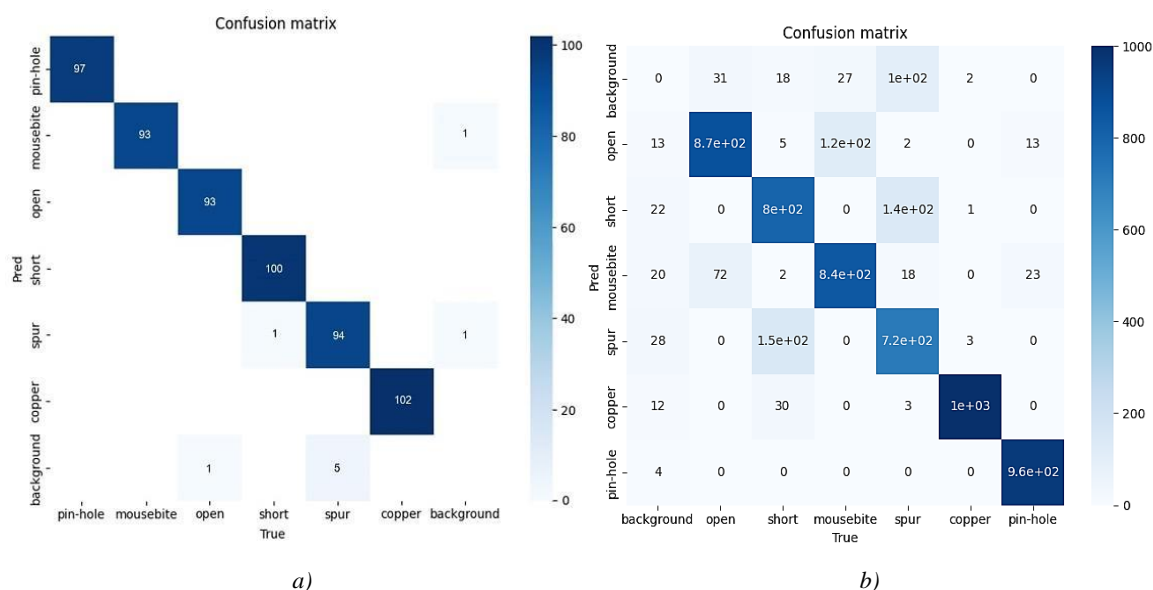


Рис. 13. Матрица ошибок, полученная на тестовой выборке набора данных PCB Defect Dataset для обученной модели YOLOv8n OBB (a); матрица ошибок, полученная на полном наборе данных PCB Defect Dataset для алгоритма сравнения с эталоном (b)

Fig. 13. Confusion matrix obtained on a test sample of the PCB Defect Dataset for trained YOLOv8n OBB model (a); confusion matrix obtained on the full PCB Defect Dataset for the reference-based PCB inspection algorithm (b)

Результаты работы алгоритма сравнения с эталоном на полном наборе данных PCB Defect Dataset представлены в табл. 7, а сопутствующая матрица ошибок – на рис. 13, b.

Таблица 7

Результаты работы алгоритма сравнения с эталоном на исходном наборе

Table 7

Results of the reference-based PCB inspection algorithm on the original set

Класс Class	Число дефектов Number of defects	Метрика оценки Evaluation metric			
		Precision	Recall	F1-score	Accuracy
Pin-hole	994	0,995 84	0,963 78	0,979 55	–
Mousebite	993	0,862 24	0,850 96	0,856 56	
Open	977	0,850 19	0,894 58	0,871 82	
Short	1004	0,829 70	0,795 82	0,812 40	
Spur	982	0,798 67	0,731 16	0,763 42	
Copper	1006	0,956 94	0,994 04	0,975 13	
Все дефекты	6055	Macro avg			0,857 80
		0,756 23	0,747 19	0,751 27	
		Weighted avg			
		0,868 16	0,857 80	0,862 48	

По результатам экспериментов нейросетевая модель показывает высокую точность детекции дефектов печатных плат (метрика mAP50 равна 0,99 на тестовом наборе данных). При этом на тестовом наборе для этой модели доля ложноположительных ошибок составляет 0,5 % относительно общего количества дефектов, а доля ложноотрицательных ошибок – 1,2 %. Наихудшие значения данных показателей среди отдельно взятых классов соответствуют классу «выступ». Доля ложноположительных ошибок составляет 2 % относительно всех дефектов данного класса, а доля ложноотрицательных ошибок – 5,1 %.

Как показали проведенные эксперименты, использование нейронных сетей для поиска и классификации дефектов имеет такие преимущества, как низкая чувствительность к качеству изображений, наличие бликов и защитного лака на поверхности печатной платы (кроме случаев непрозрачного или черного лака), присутствие текстовой и графической информации на поверхности печатной платы.

Между тем следует отметить зависимость результатов от полноты и качества обучающей выборки. Все использованные в экспериментах общедоступные наборы данных для обучения нейронных сетей имеют ошибки в разметке данных. Пары изображений (эталонное и анализируемое) в наборе данных Deep PCB не могут быть приемлемо совмещены с помощью аффинных и проекционных преобразований вследствие, предположительно, ошибочных параметров бинаризации и не должны использоваться для контроля, а также принципиально не имеют информации о связях между дорожками за пределами одного кадра. Все печатные платы, изображения которых входят в набор PCB Defects Dataset, покрыты только зеленым защитным лаком. Некоторые изображения имеют белые графические объекты, указывающие на посадочные места электронных компонентов, что значительно усложняет, а в некоторых случаях делает невозможным оптический контроль печатной платы. Дефекты обоих наборов данных выполнены формально с использованием графического редактора. Набор данных Multiple Datasets on PCB Defects¹⁴ содержит только изображения реальных дефектов печатных плат с разными условиями съемки без информации об их классах.

Перечисленные недостатки определяют необходимость гибридного поиска дефектов печатных плат, основанного на использовании алгоритма сравнения с эталоном (вычитании изображений), операций математической морфологии и нейронных сетей.

Гибридный подход заключается в выборе метода поиска дефектов в зависимости от оценки качества анализируемых изображений по следующим критериям:

- отсутствие нелинейных геометрических искажений, обусловленных непостоянством скорости сканирования печатной платы или неоднородностью свойств оптической системы формирования изображения по всему полю кадра;

- наличие бликов и теней из-за использования неудачной системы освещения;

- определение достаточной резкости изображения для обработки;

- определение общего уровня яркости изображения, количества и разнообразия оттенков. Если уровень яркости слишком высокий или, наоборот, низкий, то это приводит к потерям информации на пересвеченных или слишком темных участках изображения при выравнивании яркости.

Для поиска дефектов на изображениях высокого качества при наличии эталона рекомендуется использовать алгоритм сравнения с эталоном, который позволяет классифицировать все найденные дефекты на шесть непересекающихся классов. Для поиска отклонения от минимальной ширины дорожки и минимального расстояния между дорожками будет использоваться операция математической морфологии – дистанционное преобразование, применяемое как к изображению дорожек, так и к фону.

Возможно одновременное применение алгоритма сравнения и нейросетевого подхода с объединением результатов для снижения вероятности пропуска критически важного дефекта и количества ложных дефектов при плохом совмещении эталонного и анализируемого изображений. При отсутствии эталона или низком качестве изображения следует применять предварительно обученные нейронные сети.

При любом выборе метода поиска необходимо использовать его результаты с целью расширения набора данных для обучения и переобучения нейронных сетей, что позволит улучшить качество детекции и классификации дефектов. Особое внимание необходимо уделить вопросу организации грамотного освещения плат и регистрации их изображений. Эти факторы непосредственно влияют на качество поиска дефектов.

¹⁴Multiple Datasets on PCB Defects [Electronic resource]. – Mode of access://www.kaggle.com/datasets/jiafuwen77/multiple-datasets-on-pcb-defects. – Date of access: 11.07.2024.

Заключение. Авторами рассмотрены ключевые факторы, стимулирующие мировой рынок печатных плат, и тенденции в их проектировании и производстве, определяющие актуальность разработки системы поиска дефектов. В работе решена прикладная задача обнаружения и классификации дефектов на изображениях печатных плат на основе алгоритма сравнения с эталоном и нейросетевой архитектуры YOLOv8, проведено обучение моделей на общедоступных наборах изображений дефектов, выполнена оценка качества поиска. Использование архитектуры YOLOv8 показало высокую точность поиска, низкую чувствительность к качеству изображений и присутствию текстовой и графической информации на поверхности печатной платы. Однако недостатки обучающих выборок не позволяют однозначно рекомендовать использование только нейронных сетей для поиска дефектов, поэтому предлагается гибридный подход, который дает возможность улучшать результаты за счет применения разных методов в зависимости от оценки качества анализируемых изображений.

Вклад авторов. В. В. Венгеренко осуществил подготовку наборов данных, обучение моделей, оценку точности моделей и алгоритмов. А. В. Инютин выполнил постановку задачи, выбор наборов данных, планирование эксперимента, разработал методику поиска дефектов.

Список использованных источников

1. Карпов, С. Прецизионный контроль печатных плат. Что это? / С. Карпов // Технологии в электронной промышленности. – 2008. – № 7. – С. 37–40.
2. A GMR–ECT based embedded solution for applications on PCB inspections / M. Cacciola [et al.] // Sensors and Actuators A: Physical. – 2011. – Vol. 167, iss. 1. – P. 25–33.
3. A survey on industrial vision systems, applications and tools / E. N. Malamas [et al.] // Image and Vision Computing. – 2003. – Vol. 21, iss. 2. – P. 171–188.
4. Interpretation of texture changes during self-annealing of electroplated copper / W. Q. Zhang [et al.] // Microelectronic Engineering. – 2010. – Vol. 87, iss. 12. – P. 2488–2494.
5. Lehmann, D. K. X-ray systems for optimizing PCB inspection: x-ray systems are not all equal, and their differences affect the type of defects that can be detected / D. K. Lehmann // Circuits Assembly. – 2002. – Vol. 13, iss. 2. – P. 35–40.
6. Левданский, А. Оптический и рентгеновский контроль печатных плат при помощи одной системы / А. Левданский // Технологии в электронной промышленности. – 2005. – № 6. – С. 52–54.
7. Ebayyeh, A. A. R. Abu. A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry / A. A. R. Abu Ebayyeh, A. Mousavi // IEEE Access. – 2020. – Vol. 8. – P. 183192–183271.
8. Дудкин, А. А. Техника поиска дефектов и контроля проектных норм на изображении слоев печатных плат / А. А. Дудкин, А. В. Инютин // Искусственный интеллект. – 2006. – № 3. – С. 584–591.
9. Amjoud, A. B. Object detection using deep learning, CNNs and vision transformers: A review / A. B. Amjoud, M. Amrouch // IEEE Access. – 2023. – Vol. 11. – P. 35479–35516.

References

1. Karpov S. *Precision testing of printed circuit boards. What is it?* Tehnologii v jelektronnoj promyshlennosti [Technologies in the Electronics Industry], 2008, no. 7, pp. 37–40 (In Russ.).
2. Cacciola M., Megali G., Pellicano D., Morabito F. C. A GMR–ECT based embedded solution for applications on PCB inspections. *Sensors and Actuators A: Physical*, 2011, vol. 167, iss. 1, pp. 25–33.
3. Malamas E. N., Petrakis E., Zervakis M., Petit L., Legat J. A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 2003, vol. 21, iss. 2, pp. 171–188.
4. Zhang W. Q., Li A. D., Ma G. B., Yin K. B., Xia Y. D., ..., Yee K. W. Interpretation of texture changes during self-annealing of electroplated copper. *Microelectronic Engineering*, 2010, vol. 87, iss. 12, pp. 2488–2494.
5. Lehmann D. K. X-ray systems for optimizing PCB inspection: x-ray systems are not all equal, and their differences affect the type of defects that can be detected. *Circuits Assembly*, 2002, vol. 13, iss. 2, pp. 35–40.
6. Levdanskiy A. *Optical and X-ray inspection of printed circuit boards using one system.* Tehnologii v jelektronnoj promyshlennosti [Technologies in the Electronics Industry], 2005, no. 6, pp. 52–54 (In Russ.).

7. Ebayyeh A. A. R. Abu, Mousavi A. A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry. *IEEE Access*, 2020, vol. 8, pp. 183192–183271.

8. Dudkin A. A., Inyutin A. V. *Techniques for detecting defects and checking design standards on the image of printed circuit board layers*. *Iskusstvennyj intellekt [Artificial Intelligence]*, 2006, no. 3, pp. 584–591 (In Russ.).

9. Amjoud A. B., Amrouch M. Object detection using deep learning, CNNs and vision transformers: A review. *IEEE Access*, 2023, vol. 11, pp. 35479–35516.

Информация об авторах

Венгеренко Вадим Владимирович, магистр, младший научный сотрудник лаборатории идентификации систем, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: vengerenko@lsi.bas-net.by

Инютин Александр Владимирович, заведующий лабораторией идентификации систем, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: avin@newman.bas-net.by

Information about the authors

Vadim V. Vengerenko, Master, Junior Researcher of the Laboratory of System Identification, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: vengerenko@lsi.bas-net.by

Alexander V. Inyutin, Head of the Laboratory of System Identification, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: avin@newman.bas-net.by

БИОИНФОРМАТИКА

BIOINFORMATICS



УДК 51-76:577.322:539.19:004.94
<https://doi.org/10.37661/1816-0301-2024-21-3-80-93>

Оригинальная статья
Original Article

Адаптация архитектуры нейронной сети REINVENT для генерации потенциальных ингибиторов проникновения ВИЧ-1

Д. А. Воробьев^{1✉}, А. Д. Карпенко¹, А. В. Тузиков¹, А. М. Андрианов²

¹Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉E-mail: daniel.vorobiov.2002@yandex.ru

²Институт биоорганической химии
Национальной академии наук Беларуси,
ул. академика В. Ф. Купревича, 5, корп. 2, Минск, 220141, Беларусь

Аннотация.

Цели. Основной целью настоящей работы является адаптация архитектуры нейронной сети REINVENT для генерации потенциальных ингибиторов белка gp120 оболочки ВИЧ-1 с использованием в процессе обучения с подкреплением молекулярного докинга на графических процессорах.

Методы. Для модификации исходной модели сети использован внедренный в процессе обучения с подкреплением молекулярный докинг на графических процессорах и разработан алгоритм, позволяющий преобразовывать генерируемые сетью SMILES представления соединений в необходимый для выполнения докинга формат PDBQT. С целью ускорения обучения нейронной сети в модифицированной версии модели REINVENT использована программа докинга AutoDock-Vina-GPU-2.1, а для уточнения результатов ее работы – процедура переоценки сродства соединений к мишени с помощью оценочной функции RFScore-4.

Результаты. С помощью модифицированной версии модели REINVENT получено более 60 000 соединений, из которых около 52 000 молекул имеют величину энергии связывания с белком gp120 ВИЧ-1, сопоставимую со значением, рассчитанным для ингибитора ВИЧ-1 NBD-14204, использованного в расчетах в качестве позитивного контроля. Из отобранных 52 000 соединений около 34 000 молекул удовлетворяют ограничениям, налагаемым на потенциальное лекарство для обеспечения его биодоступности при пероральном приеме.

Заключение. Полученные результаты позволяют продемонстрировать эффективность адаптированной нейронной сети на примере конструирования новых потенциальных ингибиторов белка gp120 ВИЧ-1, способных блокировать CD4-связывающий сайт белка gp120 оболочки вируса и предотвращать его проникновение в клетки хозяина.

Ключевые слова: генеративные модели ИИ, обучение с подкреплением, компьютерный дизайн лекарств, молекулярный докинг, ВИЧ-1, белок gp120, анти-ВИЧ-препараты

Благодарности. Работа выполнена при поддержке Государственной программы научных исследований «Конвергенция 2025» (подпрограмма «Междисциплинарные исследования и новые технологии», задание 3.04.1).

Для цитирования. Адаптация архитектуры нейронной сети REINVENT для генерации потенциальных ингибиторов проникновения ВИЧ-1 / Д. А. Воробьев [и др.] // Информатика. – 2024. – Т. 21, № 3. – С. 80–93. <https://doi.org/10.37661/1816-0301-2024-21-3-80-93>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 01.07.2024

Подписана в печать | Accepted 28.08.2024

Опубликована | Published 30.09.2024

Adaptation of the REINVENT neural network architecture to generate potential HIV-1 entry inhibitors

D. A. Varabyeu^{1✉}, A. D. Karpenko¹, A. V. Tuzikov¹, A. M. Andrianov²

¹*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*

✉E-mail: daniel.vorobiov.2002@yandex.ru

²*Institute of Bioorganic Chemistry
of the National Academy of Sciences of Belarus,
st. Academician V. F. Kuprevich, 5, bldg. 2, Minsk, 220141, Belarus*

Abstract

Objectives. The main purpose of this work is to adapt the architecture of the REINVENT neural network to generate potential inhibitors of the HIV-1 envelope protein gp120 using in the learning process with reinforcement of molecular docking on GPUs.

Methods. To modify the initial network model, molecular docking on GPUs implemented in the learning process with reinforcement was used, and an algorithm was developed that allows converting the representations of connections generated by the SMILES network into the PDBQT format necessary for docking. To accelerate the learning of the neural network in the modified version of the REINVENT model, the AutoDock-Vina-GPU-2.1 docking program was used, and to clarify the results of its work, the procedure for revaluing the affinity of compounds to the target using the RFScore-4 evaluation function was used.

Results. Using a modified version of the REINVENT model, more than 60,000 compounds were obtained, of which about 52,000 molecules have a binding energy value to the HIV-1 gp120 protein comparable to the value calculated for the HIV-1 inhibitor NBD-14204, used in calculations as a positive control. Of the 52,000 compounds selected, about 34,000 molecules satisfy the restrictions imposed on a potential drug to ensure its bioavailability when taken orally.

Conclusion. The results obtained allow us to demonstrate the effectiveness of an adapted neural network by the example of designing new potential inhibitors of the gp120 HIV-1 protein capable of blocking the CD4-binding site of the gp120 virus envelope protein and preventing its penetration into host cells.

Keywords: generative AI, reinforcement learning, computer-aided drug design, molecular docking, HIV-1, gp120 protein, anti-HIV drugs

Acknowledgments. The work was carried out with the support of the State Scientific Research Program "Convergence 2025" (subprogram "Interdisciplinary Research and New Technologies", task 3.04.1).

For citation. Varabyeu D. A., Karpenko A. D., Tuzikov A. V., Andrianov A. M. *Adaptation of the REINVENT neural network architecture to generate potential HIV-1 entry inhibitors*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 80–93 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-80-93>

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

Введение. Разработка эффективных и безопасных лекарственных средств является чрезвычайно важной и востребованной задачей мирового здравоохранения. Многие заболевания, которые ранее считались неизлечимыми, а причины и условия их возникновения оставались неясными, сегодня успешно исследуются с помощью компьютерных технологий, позволяющих значительно сократить временные и финансовые затраты, необходимые для создания новых лекарственных препаратов. В последние годы эти технологии все чаще применяются на всех этапах разработки лекарственных средств, включая виртуальный скрининг потенциальных лекарств [1–6], перепрофилирование лекарств [7, 8], предсказание аффинности связывания малых молекул с белковой мишенью и профиля межмолекулярных взаимодействий [9–12].

Использование алгоритмов глубокого обучения может существенно повысить эффективность проводимых исследований и, кроме того, сконструировать соединения, еще не зарегистрированные в существующих химических базах данных [5, 6], открывая таким образом многообещающие перспективы для создания новых низкомолекулярных соединений, которые могли бы служить более мощными ингибиторами терапевтических мишеней, чем доступные в настоящее время лекарственные препараты. Однако при создании новой модели нейронной сети для дизайна потенциальных лекарств исследователи неизменно сталкиваются с одной и той же проблемой, связанной с необходимостью оперировать большими массивами обучающих данных, что возможно в полной мере лишь для очень ограниченного числа белковых мишеней. В связи с этим перспективным методом разработки лекарств становится обучение с подкреплением [13–15].

Концепция обучения с подкреплением использует два основных понятия – агента и среды. Агенту обучения, имитирующему некоторую сущность, приходится сталкиваться с определенной средой, имитирующей ряд реальных процессов, и по мере взаимодействия с ней достигать большего вознаграждения, развивая те действия, которые дают положительное подкрепление, и избегая действий, влекущих за собой штрафы. При этом в качестве «действия» агента может выступать упорядоченный набор символов, в том числе и представления молекул. Основным преимуществом обучения с подкреплением служит то, что такая модель часто не требует специализированного набора обучающих данных (этот факт является серьезной проблемой применения технологий глубокого обучения), а нуждается только в специфической среде, позволяющей выделить среди совершенных агентом действий те, которые наиболее подходят для реальных условий.

На сегодняшний день одной из наиболее перспективных моделей нейронной сети для дизайна потенциальных лекарств с помощью обучения с подкреплением является серия платформ REINVENT [14–16]. В данной модели реализовано множество способов задания обучающей среды, в том числе с помощью оценки сродства низкомолекулярного соединения к белковой мишени, общепризнанным критерием которого является минимизация значения свободной энергии связывания (ΔG). Большинство потенциальных лекарственных средств, разрабатываемых с помощью методов компьютерного моделирования, проходят процедуру прогнозирования величины ΔG , которая обычно выполняется с использованием алгоритмов молекулярного докинга [17].

В модели нейронной сети Reinvent для проведения молекулярного докинга применяется платформа DockStream [18], объединяющая несколько методов молекулярного докинга в один вычислительный инструмент для обучения с подкреплением. Однако все используемые в DockStream методы молекулярного докинга реализованы на центральных процессорах, что с учетом трудоемкости применяемых ими алгоритмов вынуждает пользователей либо прибегать к использованию больших вычислительных мощностей (что не всегда возможно), либо понижать точность предсказания энергии связывания для сокращения времени вычислений,

а это может значительно повлиять на результат обучения агента. На сегодняшний день разработаны алгоритмы молекулярного докинга с использованием графических процессоров, позволяющие значительно ускорить время расчетов, не снижая точности прогнозирования значения ΔG .

Цель настоящей работы заключается в адаптации архитектуры нейронной сети REINVENT для генерации потенциальных ингибиторов белка gp120 оболочки ВИЧ-1 с использованием в процессе обучения с подкреплением молекулярного докинга на графических процессорах. Для достижения этой цели в работе проведены исследования, которые включали:

- разработку модифицированной версии REINVENT, использующей для ускорения обучения с подкреплением программу AutoDock-Vina-GPU-2.1 [19];
- создание программного конвейера для конвертации генерируемых нейронной сетью SMILES-строк молекул в формат PDBQT;
- обучение модели с привлечением в качестве терапевтической мишени структуры белка gp120 ВИЧ-1 – гликопротеина оболочки вируса, критически важного для его проникновения в клетки хозяина;
- тестирование нейронной сети и анализ результатов ее работы.

1. Материалы и методы

1.1. Архитектура нейронной сети. В качестве базовой модели использовали нейронную сеть SMILES-RL [20], входящую в состав архитектуры REINVENT [15] и представляющую различные реализации обучения с подкреплением. Модель SMILES-RL является рекуррентной нейронной сетью со слоями LSTM, в которой реализованы 29 комбинаций алгоритмов обучения с подкреплением и буферов воспроизведения. Как и в большинстве предложенных в литературе моделей обучения с подкреплением, процесс обучения сети SMILES-RL происходил циклично (здесь и далее один цикл обучения будем называть шагом). Для каждого шага генерировали 128 строк SMILES (Simplified Molecular Input Line Entry System) [21] – формата представления низкомолекулярных соединений в виде строк, состоящих из букв, обозначающих химические элементы, и специальных символов.

Для достижения цели исследования авторами был выбран алгоритм обучения с подкреплением Advantage Actor-Critic (A2C) [22] стохастического метода Policy Gradient в сочетании с буфером воспроизведения AC, который использует для обучения агента все сгенерированные на текущем шаге последовательности и учитывает только опыт текущего шага. Выбор алгоритма A2C и буфера воспроизведения AC обусловлен тем, что их комбинация обеспечивала генерацию одного из наибольших наборов потенциально активных молекул, полученных с помощью базовой модели SMILES-RL [20]. Авторы работы [20] для оценки сгенерированных нейронной сетью SMILES-RL соединений на каждом шаге обучения применяли алгоритм случайного леса (Random Forest, RF), с помощью которого каждому полученному соединению ставили в соответствие число-вознаграждение, варьирующее от 0 до 1. В модели нейронной сети, реализованной в настоящей работе, этап оценки был полностью изменен. Так, в предлагаемой версии модели REINVENT вместо алгоритма Random Forest был реализован другой алгоритм оценки сгенерированных соединений (см. подразд. 1.2). При этом все остальные этапы обучения нейронной сети проводили по схеме, реализованной в базовой модели SMILES-RL [20].

1.2. Обучение нейронной сети. Процесс обучения длился 10 полных эр и одну нулевую эру, каждая из полных эр включала 50 шагов. Нулевая эра состояла всего из одного шага, и ее вводили для проверки корректности работы алгоритма генерации и оценки соединений. Это было необходимо, так как обучение полной эры, т. е. проведение 50 шагов обучения, занимало в среднем около 6 ч. В конце каждой эры веса для агента сохраняли и в начале следующей эры загружали в качестве исходных данных для агента. В начале нулевой эры использовали случайные веса для неспецифической модели, предварительно обученной генерировать правильные представления SMILES.

На каждом шаге обучения сгенерированные последовательности SMILES конвертировали в формат PDBQT (Protein Data Bank, Partial Charge (Q) & Atom Type (T)), необходимый для проведения молекулярного докинга (см. подразд. 1.3). Далее с помощью программы AutoDock

Vina-GPU-2.1, работающей на графических процессорах, для всех корректно конвертированных соединений проводили молекулярный докинг с белком gp120 ВИЧ-1 (см. подразд. 1.5). Затем результаты докинга для каждого соединения подвергали переоценке с помощью оценочной функции (ОФ) машинного обучения RFScore-4 для того, чтобы уточнить прогнозируемое значение ΔG (см. подразд. 1.6). Исходя из значения ΔG , каждому соединению ставили в соответствие число-вознаграждение от 0 до 1 (см. подразд. 1.7). Если сгенерированная последовательность SMILES-строк не могла быть конвертирована в формат PDBQT, то этой последовательности ставили в соответствие вознаграждение 0. Все соединения проверяли также на соответствие заданным ограничениям на физико-химические параметры, традиционно используемые в виртуальном скрининге наиболее вероятных кандидатов на лекарственные средства для предсказания молекул с большим потенциалом биодоступности при пероральном введении (см. подразд. 1.8). Если соединение не удовлетворяло ограничениям какого-либо физико-химического параметра, то вознаграждение для этого соединения умножали на соответствующий штраф – число от 0 до 1.

1.3. Программный конвейер для конвертации строк SMILES в формат PDBQT. Форматом выходных данных для выбранной модели нейронной сети являются строки SMILES [21], однако программный пакет AutoDock-Vina-GPU-2.1 оперирует с данными в формате PDBQT. Для создания представлений сгенерированных соединений в формате PDBQT авторами был разработан программный конвейер на языке программирования python (рис. 1).

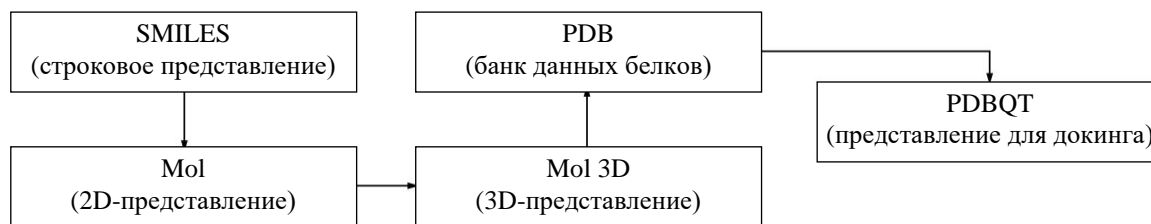


Рис. 1. Общая схема программного конвейера для подготовки соединений к графически ускоренному молекулярному докингу

Fig. 1. The software pipeline general scheme to prepare compounds for graphically accelerated molecular docking

На первом этапе с помощью программного пакета RDKit (URL: <http://www.rdkit.org/>) генерировали двухмерные (2D) структуры молекул с сохранением их топологии, состава и типов химических связей в файлах с расширением .mol, которые преобразовывали в трехмерные (3D) структуры в файлах .mol3d (рис. 1). Затем с привлечением программных средств MGLtools полученные файлы .mol3d конвертировали в формат PDB, используемый для хранения соединений в банке данных белков (Protein Data Bank [23]), и переводили их в формат PDBQT.

1.4. Молекулярный докинг. Кристаллическую структуру белка gp120 ВИЧ-1 в комплексе с контрольным ингибитором NBD-14204 заимствовали из Банка данных белков (PDB ID: 8F9Z) [24]. Подготовку белка-мишени к молекулярному докингу осуществляли с помощью пакета UCSF Chimera [25] и программных средств веб-сервера открытого доступа SWISS-MODEL [26]. Атомы водорода добавляли к структуре белка gp120 с использованием программы UCSF Chimera [25], а исходный файл преобразовывали из формата .pdb в формат .pdbqt с помощью программного пакета AutoDockTools-1.5.7 (<https://ccsb.scripps.edu/mgltools/1-5-7/>). Молекулярный докинг проводили с использованием программы AutoDock-Vina-GPU-2.1 [19] в приближении жесткого рецептора и гибких лигандов. Ячейка для докинга включала CD4-связывающий сайт белка gp120 ВИЧ-1 [24] и имела следующие параметры: $\Delta X = 21,5 \text{ \AA}$, $\Delta Y = 21 \text{ \AA}$, $\Delta Z = 25 \text{ \AA}$ с центром в точках $X = 2,4 \text{ \AA}$, $Y = 16,57 \text{ \AA}$, $Z = 11,95 \text{ \AA}$. Параметр, характеризующий полноту поиска (охват конформационного пространства), был задан равным 30. В качестве позитивного контроля в расчетах использовали ингибитор проникновения ВИЧ-1 NBD-14204, блокирующий связывание вируса с первичным рецептором CD4 клеточной мембраны и его проникновение в клетки-мишени [24].

1.5. Сравнение работы программ AutoDock-Vina и AutoDock-Vina-GPU-2.1. Для реализации молекулярного докинга было крайне важно сократить время его выполнения, поскольку даже при графическом ускорении расчетов обучение на 50 шагах занимало около 6 ч. Более длительные вычисления или меньший охват конформационного пространства сделали бы использование этой процедуры на этапе обучения нейронной сети нецелесообразным. Для решения данной проблемы может быть использован молекулярный докинг на графических процессорах. Однако в этом случае необходимо показать, что его применение как минимум не уменьшает точность прогнозирования энергии связывания по сравнению с традиционно используемой программой молекулярного докинга AutoDock-Vina 1.2. Поэтому, для того чтобы оценить эффективность докинга на графических процессорах, на наборе данных с известными экспериментальными значениями констант диссоциации (K_d) комплексов лиганд-белок был проведен сравнительный анализ точности предсказания аффинности связывания малых молекул с белками-мишенями с использованием программ AutoDock-Vina 1.2.5 и AutoDock-Vina-GPU-2.1 [19]. AutoDock-Vina-GPU-2.1 базируется на алгоритмах программы AutoDock-Vina 1.0, адаптированных для проведения расчетов на графических процессорах [19]. Для сравнения точности прогнозирования величины K_d (моль/л) с помощью данных программ использовалась база данных Binding MOAD [27], из которой были отобраны кристаллические структуры комплексов лиганд-белок с разрешением $< 2,5 \text{ \AA}$, рекомендованным для корректного выполнения молекулярного докинга. Затем для отобранных из этой базы данных 1967 комплексов с помощью ОФ AutoDock-Vina [17] и AutoDockVina-GPU-2.1 [19] были предсказаны значения $pK_d = -\log(K_d)$ и выполнен повторный молекулярный докинг с использованием программы AutoDock-Vina-GPU-2.1 для оценки разброса данных, полученных одним и тем же методом при изменении случайного начального значения. Результаты проведенных расчетов представлены на рис. 2.

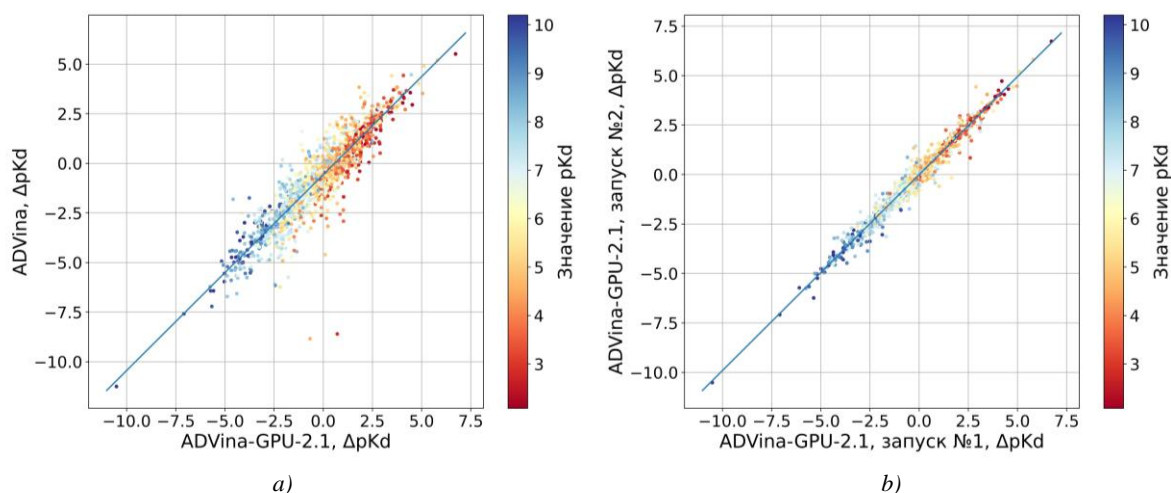


Рис. 2. Отклонения значений pK_d (ΔpK_d), рассчитанных программами AutoDock-Vina и AutoDock-Vina-GPU-2.1, от известных экспериментальных значений ($pK_d(\text{рассчит.}) - pK_d(\text{эксперим.})$) (a) и отклонения значений pK_d для того же набора данных, предсказанные AutoDock-Vina-GPU-2.1 в результате двух расчетов с разными случайными начальными значениями (b)

Fig. 2. pK_d values deviations (ΔpK_d) calculated by AutoDock-Vina and AutoDock-Vina-GPU-2.1 from known experimental values ($pK_d(\text{calc}) - pK_d(\text{exp})$) (a) and pK_d values deviations for the same data set predicted by AutoDock-Vina-GPU-2.1 as a result of two calculations with different random initial values (b)

1.6. Сравнение оценочных функций AutoDock-Vina, AutoDock-Vina-GPU-2.1 и RFScore-4. После проведения молекулярного докинга для уточнения полученных результатов и переоценки K_d была использована ОФ машинного обучения RFScore-4 [28], основанная на алгоритме случайного леса и предназначенная для виртуального скрининга химических баз данных. Отклонения значений pK_d , предсказанных с помощью ОФ RFScore-4 и AutoDock-Vina, от экспериментальных величин (ΔpK_d) представлены на рис. 3.

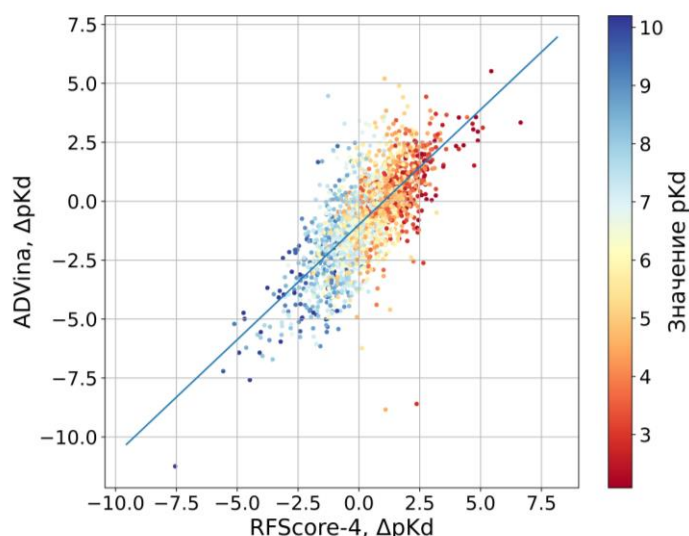


Рис. 3. Отклонения значений pK_d , рассчитанных с помощью ОФ AutoDock-Vina и RFScore-4, от экспериментальных величин (ΔpK_d)

Fig. 3. pK_d values deviations calculated by the SF AutoDock-Vina and RFScore-4 from the experimental data (ΔpK_d)

Средние значения отклонений pK_d между предсказанными и экспериментальными величинами и соответствующие среднеквадратические отклонения свидетельствуют о том (табл. 1), что графически ускоренный молекулярный докинг AutoDock-Vina-GPU-2.1 показал на использованном наборе экспериментальных данных результаты, сопоставимые с результатами, полученными с помощью программы AutoDock-Vina. Несмотря на то что для ОФ RFScore-4 наблюдается большее среднее отклонение предсказанных значений, чем для ОФ AutoDock-Vina-GPU-2.1, ОФ RFScore-4 характеризуется меньшим среднеквадратическим отклонением между экспериментальными и предсказанными величинами аффинности связывания, а также большим значением коэффициента корреляции Пирсона, что чрезвычайно важно для успешного решения задачи обучения нейронной сети (табл. 1).

Таблица 1
Сравнение оценочных функций

Table 1
Comparison of scoring functions

Оценочная функция <i>Evaluation function</i>	Среднее отклонение, ΔpK_d <i>Average deviation, ΔpK_d</i>	Среднеквадратическое отклонение, ΔpK_d <i>Standard deviation, ΔpK_d</i>	Коэффициент корреляции Пирсона <i>Pearson's correlation coefficient</i>	p -уровень значимости для коэффициента корреляции Пирсона <i>p-level of significance for Pearson correlation coefficient</i>
AutoDock-Vina	-0,736	1,897	0,469	$1,69 \cdot 10^{-108}$
AutoDock-Vina-GPU-2.1	-0,169	1,769	0,476	$6,13 \cdot 10^{-112}$
RFScore-4	0,266	1,411	0,661	$7,37 \cdot 10^{-248}$

1.7. Функция вознаграждения. Для применения молекулярного докинга в процессе обучения агента необходимо было преобразовать его выходные данные, а именно значения ΔG , в величины, с которыми оперирует нейронная сеть. Так как базовая модель [20] использует в качестве вознаграждения за каждое сгенерированное соединение реальное значение в интервале от

0 до 1, было решено конвертировать величины ΔG в ту же форму. Для этой цели было удобно использовать асимптотические функции, которые достигают значения 0 или 1 только при бесконечно высоких или низких величинах аргумента соответственно, поскольку в таком случае отсутствует вероятность ошибки при получении неожиданно высокого или низкого значения ΔG .

Для решения поставленной задачи была выбрана модифицированная кумулятивная функция нормального распределения (1) и (2) в предположении, что значения ΔG для сгенерированных соединений имеют нормальное распределение. Функция (1) отличается от кумулятивной функции нормального распределения тем, что для нее большие значения $F(x)$ соответствуют меньшему значению x . Это изменение связано с тем, что большему значению аффинности связывания, на которое в данной работе было ориентировано обучение модели, соответствует меньшее значение ΔG :

$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{-x + \mu}{\sigma\sqrt{2}} \right) \right], \quad (1)$$

где x – значение ΔG для определенного соединения; μ – среднее значение ΔG (значение, при котором вознаграждение $F(x) = 0,5$), выбранное таким образом, чтобы величина вознаграждения для контрольного ингибитора находилась в диапазоне от 0,5 до 0,6; σ – стандартное отклонение, подобранное эмпирически так, чтобы «энергетическая ценность» вознаграждения $F(x) = 0,95$ находилась в диапазоне от 10,5 до 11,0 ккал/моль.

Функция $\operatorname{erf}(z)$ определяется следующим образом:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt. \quad (2)$$

Следует отметить, что чем ниже значение ΔG , тем сильнее средство, поскольку свободная энергия связывания представляет собой разницу между внутренней энергией комплекса лиганда с белком и суммой их энергий в свободном состоянии. Используемая для оценки ΔG функция вознаграждения представлена на рис. 4.

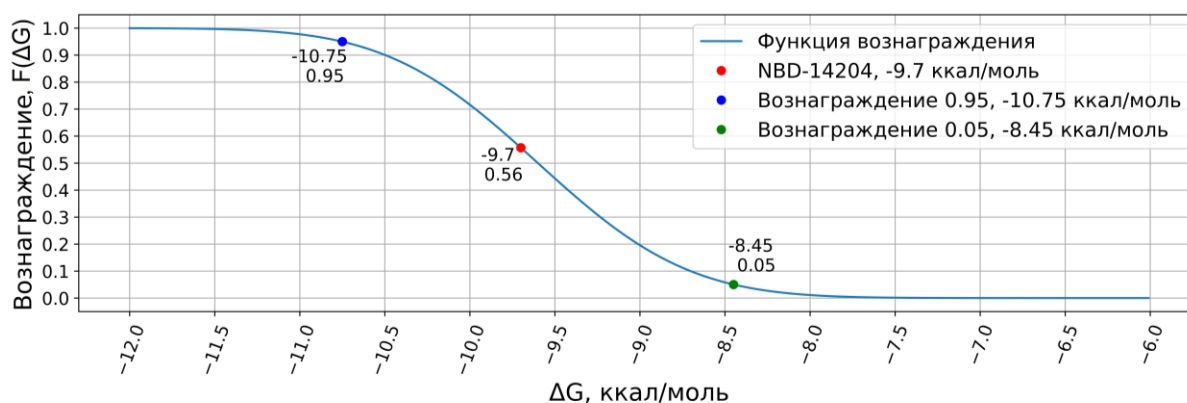


Рис. 4. Функция, использованная в процессе обучения с подкреплением для преобразования значений ΔG в вознаграждение

Fig. 4. Function, used for converting the values of binding free energy into reward during the reinforcement learning

1.8. Дополнительные ограничения. Кроме ограничений на значения ΔG , на генерируемые соединения налагались штрафы за отклонение физико-химических параметров молекул от требований, предъявляемых к ним правилом пяти Липинского [29], которое используется для предсказания молекул с большим потенциалом биодоступности при пероральном введении.

Правило пяти принимает в рассмотрение такие важные характеристики потенциального лекарства, как всасывание, распределение, метаболизм и экскреция. Кроме того, в функцию вознаграждения вводили штрафы за нарушение ограничений на топологическую площадь полярной поверхности, число вращательных связей, логарифм молярной растворимости соединений в воде и их реакционную способность (степень ненасыщенности), которые также оказывают существенное влияние на оральную биодоступность кандидатов в лекарства [29–31]. Полный список ограничений на физико-химические параметры, использованных в процессе обучения нейронной сети, представлен в табл. 2.

Таблица 2

Штрафные коэффициенты, введенные в функцию вознаграждения за нарушение ограничений на физико-химические параметры соединений

Table 2

Penalty coefficients introduced into the reward function for violating restrictions on the physicochemical parameters of compounds

Физико-химические параметры соединений <i>Physicochemical parameters of compounds</i>	Ограничения на физико-химические параметры <i>Limitations on physical and chemical parameters</i>		Штрафной коэффициент <i>Penalty coefficient</i>
Правило пяти Липински	Молекулярная масса (Да)	≤ 500	0,8
	Число доноров водородной связи	≤ 5	
	Число акцепторов водородной связи	≤ 10	
	Липофильность соединения	$-0,7 < XLOGP3 < 5,0$	
	При этом допускалось нарушение не более одного условия		
Топологическая площадь полярной поверхности (TPSA)	$20 \text{ \AA}^2 < TPSA < 130 \text{ \AA}^2$		0,8
Число вращательных связей (N)	$N \leq 10$		0,8
Логарифм молярной растворимости соединения в воде (LogS)	$-6 < \text{Log S} < 0$		0,8
Ненасыщенность соединения ^{1,2} (Csp3)	$0,25 < Csp3 < 1$		$\frac{< Csp3 >}{0,25}$

¹Ненасыщенные соединения (непредельные соединения) – органические молекулы, скелет которых содержит кратные (C=C, C≡C) связи. Ненасыщенные соединения характеризуются высокой реакционной способностью.

²Штраф за нарушение ограничений, относящихся к ненасыщенности соединений, отличается от остальных. Он может быть более строгим, так как при обучении сети без наложения штрафных санкций, свыше 85 % сгенерированных соединений не соответствовали этим ограничениям, в то время как по остальным параметрам не удовлетворяло требованиям от десятой части до двух третей соединений.

¹Unsaturated compounds (unsaturated compounds) are organic molecules whose skeleton contains multiple (C=C, C≡C) bonds. Unsaturated compounds are characterized by high reactivity.

²The penalty for violating the restrictions related to the unsaturation of compounds differs from the others. It can be more stringent, since when training the network without imposing penalties, over 85 % of the generated compounds did not meet these restrictions, while for other parameters, from a tenth to two thirds of the compounds did not meet the requirements.

Итоговое вознаграждение для каждого соединения рассчитывали путем умножения вознаграждения, полученного на основе данных молекулярного докинга, на штрафы, наложенные за отклонения физико-химических параметров от установленных ограничений:

$$\text{Reward} = F(\Delta G) \prod_i x_i, \quad (3)$$

где x_i – штрафной коэффициент для i -го параметра, равный значению в табл. 2, если ограничение нарушено, и единице, если соединение удовлетворяет ограничениям i -го параметра. $F(\Delta G)$ вычисляется по формуле (1) для величины ΔG .

2. Результаты и обсуждение. Процесс обучения нейронной сети длился около 60 ч с использованием графического процессора RTX 3080 TI и включал 500 шагов, разделенных на 10 эр, в течение которых нейронная сеть генерировала в среднем от 120 до 128 оригинальных (без учета дубликатов) соединений за шаг. Потери в количестве соединений связаны с тем, что не все сгенерированные нейронной сетью последовательности символов из словаря SMILES удалось преобразовать по правилам SMILES в структуры химических соединений. Соответственно, такие последовательности символов не могли быть корректно конвертированы программным конвейером в PDBQT-формат и их пришлось отбросить. Средняя величина итоговой функции вознаграждения за каждый шаг показана на рис. 5.

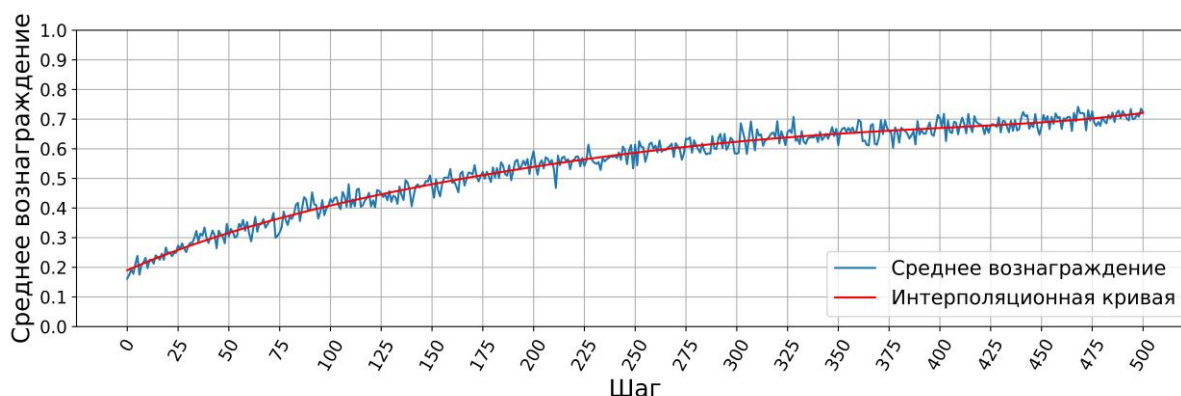


Рис. 5. Средние значения функции вознаграждения для сгенерированных на каждом шаге обучения SMILES-строк

Fig. 5. Mean values of reward function for generated SMILES strings per step of training

В результате нейронная сеть сгенерировала более 62 000 соединений, из которых свыше 52 000 молекул имели значения ΔG ниже $-9,2$ ккал/моль, сопоставимые с величиной $-9,7$ ккал/моль, полученной для контрольного ингибитора ВИЧ-1 NDB-14204. На рис. 6 изображен график зависимости числа генерируемых соединений за шаг, которые удовлетворяли ограничениям, наложенным на их физико-химические параметры, и имели величину ΔG ниже $-9,2$ ккал/моль. Итоговое число таких соединений превысило 34 000.

Следует отметить, что в работе также было проведено обучение модели с использованием ОФ AutoDock-Vina-GPU-2.1, но без учета ограничений на физико-химические параметры молекул. Когда полученные таким путем молекулы были отобраны по критерию соответствия введенным ограничениям, то их количество составило менее 3000. Это связано с тем, что при обучении сети без дополнительных ограничений штрафование соединений, которые не удовлетворяют этим ограничениям, отсутствует и единственным ориентиром для сети становятся соединения с наименьшей величиной ΔG . Однако при наложении штрафов за нарушения сеть обучается избегать молекулы с отклонениями от заданных параметров и учится ориентироваться на соединения, удовлетворяющие ограничениям, из-за чего с каждой эрой их количество возрастает.

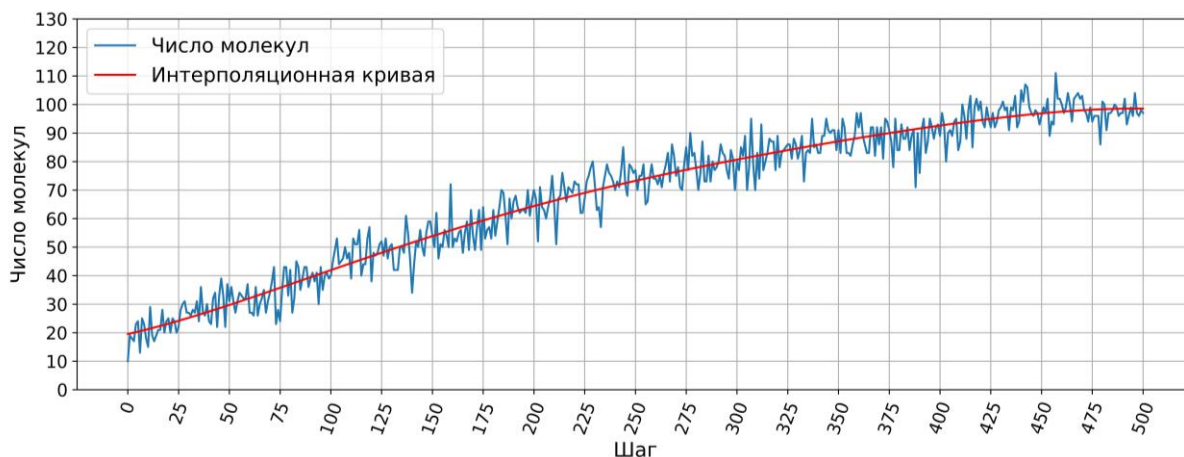


Рис. 6. Число сгенерированных за шаг обучения нейронной сети молекул, удовлетворяющих всем заданным ограничениям на физико-химические параметры и имеющих значения ΔG , более низкие или сопоставимые с величиной, рассчитанной для контрольного ингибитора NBD-14204

Fig. 6. Number of molecules that meet all requirements imposed on physicochemical parameters and exhibit values of ΔG comparable or lower than that calculated for the reference inhibitor NBD-14204

Также важно отметить, что уменьшение свободной энергии связывания для генерируемых соединений во время обучения слабо связано с тенденцией к росту их молекулярной массы. Это может существенно влиять на величину молекулярной массы по крайней мере из-за того, что с увеличением числа атомов возрастает количество возможных межмолекулярных взаимодействий между лигандом и белком-мишенью. Большинство соединений как с очень большими, так и с очень малыми молекулярными массами были получены в эры 1 и 2, т. е. когда агент был еще слабо специфичен к целевому белку gp120. Кроме того, вклад каждой эры в конечное распределение молекулярной массы скорее одинаков, с небольшим перевесом большей массы для поздних эр, в то время как для конечного распределения свободной энергии связывания ее более низкие значения явно соответствуют более поздним эрам (рис. 7).

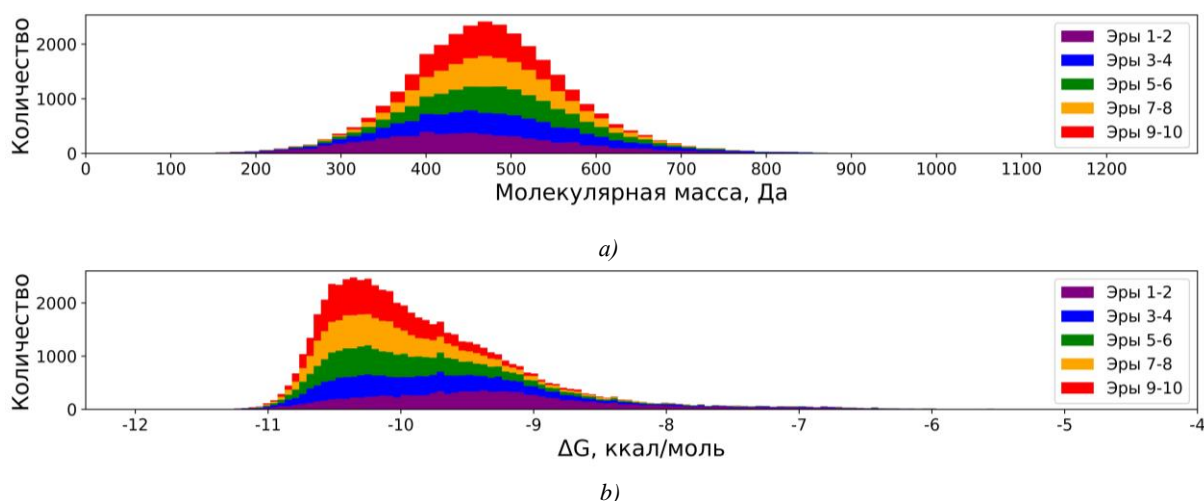


Рис. 7. Распределение молекулярных масс (a) и свободной энергии связывания (b) с указанием вклада соответствующих эр обучения

Fig. 7. The molecular weights distribution (a) and binding free energy (b) indicating the contribution of the corresponding eras of training

Заключение. В настоящем исследовании адаптирована модель глубокой генеративной нейронной сети на основе архитектуры REINVENT, использующая в процессе обучения с подкреплением графически ускоренный молекулярный докинг и учитывающая ограничения на физико-химические параметры генерируемых соединений, введенные для идентификации молекул с большим потенциалом оральной биодоступности. Для решения поставленной задачи создан программный конвейер, предназначенный для преобразования SMILES-строк молекул в формат PDBQT. Для преобразования результатов молекулярного докинга в данные, приемлемые для обучения с подкреплением, использована функция вознаграждения, основанная на кумулятивной функции нормального распределения. На наборе данных для комплексов лиганд-белок с известными экспериментальными значениями K_d проведен сравнительный анализ точности предсказания значений ΔG с помощью программ AutoDock-Vina и AutoDock-Vina-GPU-2.1 и ОФ Vina, Vina-GPU-2.1 и RFScore-4. В результате показано, что использование графически ускоренного метода AutoDock-Vina-GPU-2.1 в сочетании с ОФ RFScore-4 является наиболее продуктивным для успешного обучения предлагаемой модели. Полученная модификация нейронной сети не требует набора обучающих данных и способна генерировать большое количество соединений с высокой прогнозируемой активностью по отношению к заданной терапевтической мишени и приемлемыми фармакологическими свойствами.

Эффективность адаптированной нейронной сети продемонстрирована на примере конструирования новых потенциальных ингибиторов белка gp120 ВИЧ-1, способных блокировать CD4-связывающий сайт белка gp120 оболочки вируса и предотвращать его проникновение в клетки хозяина. В результате проведенных расчетов идентифицировано более 34 000 высокоаффинных лигандов этого гликопротеина, потенциально активных при пероральном приеме. В связи с этим развитие настоящей работы предполагает проведение дальнейших теоретических исследований, включающих молекулярную динамику комплексов лиганд-gp120, отбор соединений, перспективных для тестирования *in vitro*, и оптимизацию структур соединений-лидеров, направленную на получение их аналогов с улучшенной противовирусной активностью.

В заключение отметим, что предлагаемая модель нейронной сети может быть использована для компьютерного дизайна потенциальных кандидатов в лекарственные средства, способных ингибировать разные терапевтические мишени, играющие важную роль в развитии опасных заболеваний человека.

Вклад авторов. Д. А. Воробьев и А. Д. Карпенко разработали и реализовали модификацию модели генеративной нейронной сети, обучили, протестировали и апробировали нейронную сеть. Д. А. Воробьев провел молекулярный докинг сгенерированных соединений с белком gp120 ВИЧ-1. А. М. Андрианов и А. В. Тузиков руководили проектом и написали рукопись. Все авторы проанализировали данные расчетов, обсудили полученные результаты и внесли свой вклад в окончательную версию рукописи.

References

1. Li H., Sze K. H., Lu G., Ballester P. Machine-learning scoring functions for structure-based virtual screening. *Wiley interdisciplinary reviews: Computational Molecular Science*, 2020, vol. 11. Available at: <https://wires.onlinelibrary.wiley.com/doi/10.1002/wcms.1478> (accessed 20.05.2024). <https://doi.org/10.1002/wcms.1478>
2. Xiong G. L., Ye W. L., Shen C., Lu A. P., Hou T. J., Cao D. S. Improving structure-based virtual screening performance via learning from scoring function components. *Briefings in Bioinformatics*, 2020. Available at: <https://academic.oup.com/bib/article/22/3/bbaa094/5851268> (accessed 20.05.2024). <https://doi.org/10.1093/bib/bbaa094>
3. Stokes J. M., Yang K., Swanson K., Jin W., Cubillos-Ruiz A. A deep learning approach to antibiotic discovery. *Cell*, 2020, vol. 180, pp. 688–702. <https://doi.org/10.1016/j.cell.2020.01.021>
4. Timmons P. B., Hewage C. M. ENNAVIA is a novel method which employs neural networks for antiviral and anti-coronavirus activity prediction for therapeutic peptides. *Briefings in Bioinformatics*, 2021. Available at: <https://academic.oup.com/bib/article/22/6/bbab258/6326528> (accessed 20.05.2024). <https://doi.org/10.1093/bib/bbab258>

5. Andrianov A. M., Nikolaev G. I., Shuldov N. A., Bosko I. P., Anischenko A. I., Tuzikov A. V. Application of deep learning and molecular modeling to identify small drug-like compounds as potential HIV-1 entry inhibitors. *Journal of Biomolecular Structure and Dynamics*, 2022, vol. 40, pp. 7555–7573. <https://doi.org/10.1080/07391102.2021.1905559>
6. Zhang Y., Ye T., Xi H., Juhas M., Li J. Deep learning driven drug discovery: Tackling Severe Acute Respiratory Syndrome Coronavirus 2. *Frontiers in Microbiology*, 2021. Available at: <https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2021.739684/full> (accessed 20.05.2024). <https://doi.org/10.3389/fmicb.2021.739684>
7. Kinnings S. L., Liu N., Tonge P. J., Jackson R. M., Xie L., Bourne P. E. A Machine learning-based method to improve docking scoring functions and its application to drug repurposing. *Journal of Chemical Information and Modeling*, 2011, vol. 51, no. 5, pp. 1195–1197. <https://doi.org/10.1021/ci2001346>
8. Agastheeswaramoorthy K., Sevilimedu A. Drug REpurposing using AI/ML tools – for Rare Diseases (DREAM-RD): A case study with Fragile X Syndrome (FXS). *bioRxiv*, 2020. Available at: <https://www.biorxiv.org/content/10.1101/2020.09.25.311142v1.full.pdf> (accessed 20.05.2024). <https://doi.org/10.1101/2020.09.25.311142>
9. Heinzlmann G., Gilson M. K. Automation of absolute protein-ligand binding free energy calculations for docking refinement and compound evaluation. *Scientific Reports*, 2021, vol. 11, no. 1, p. 1116. <https://doi.org/10.1038/s41598-020-80769-1>
10. Meli R., Morris G. M., Biggin P. C. Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: A review. *Frontiers in Bioinformatics*, 2022, vol. 2. Available at: <https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2022.885983/full> (accessed 20.05.2024). <https://doi.org/10.3389/fbinf.2022.885983>
11. Metcalf D. P., Glick Z. L., Bortolato A., Jiang A., Cheney D. L., Sherrill C. D. Directional ΔG neural network (Dr ΔG -Net): A modular neural network approach to binding free energy prediction. *Journal of Chemical Information and Modeling*, 2024, vol. 64, no. 6, pp. 1907–1918. <https://doi.org/10.1021/acs.jcim.3c02054>
12. Li Y., Fan Z., Rao J., Chen Z., Chu Q., ..., Li X. An overview of recent advances and challenges in predicting compound-protein interaction (CPI). *Medical Review*, 2023, vol. 3, no. 6, pp. 465–486. <https://doi.org/10.1515/mr-2023-0030>
13. Zhavoronkov A., Ivanenkov Y. A., Aliper A., Veselov M. S., Aladinskiy V. A., ..., Aspuru-Guzik A. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, 2019, vol. 37, pp. 1038–1040. <https://doi.org/10.1038/s41587-019-0224-x>
14. Olivecrona M., Blaschke T., Engkvist O., Chen H. J. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 2017, vol. 9, p. 48. <https://doi.org/10.1186/s13321-017-0235-x>
15. Blaschke T., Arús-Pous J., Chen H., Margreitter C., Tyrchan C., ..., Patronov A. Reinvent 2.0: an AI tool for de novo drug design. *Journal of Chemical Information and Modeling*, 2020, vol. 60, no. 12, pp. 5918–5922. <https://doi.org/10.1021/acs.jcim.0c00915>
16. Loeffler H. H., He J., Tibo A., Janet J. P., Voronov A., ..., Engkvist O. Reinvent 4: Modern AI-driven generative molecule design. *Journal of Cheminformatics*, 2024, vol. 16, no. 20. Available at: <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-024-00812-5> (accessed 20.05.2024). <https://doi.org/10.1186/s13321-024-00812-5>
17. Trott O., Olson A. J. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 2010, vol. 31, no. 2, pp. 455–461. <https://doi.org/10.1002/jcc.21334>
18. Guo J., Janet J. P., Bauer M. R., Voronov A., Mervin L. H., Engkvist O. DockStream: a docking wrapper to enhance de novo molecular design. *Journal of Cheminformatics*, 2021, vol. 13. Available at: <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00563-7> (accessed 20.05.2024). <https://doi.org/10.1186/s13321-021-00563-7>
19. Tang S., Ding J., Zhu X., Wang Z., Zhao H., Wu J. Vina-GPU 2.1: towards further optimizing docking speed and precision of AutoDock Vina and its derivatives. *bioRxiv*, 2023. Available at: <https://www.biorxiv.org/content/10.1101/2023.11.04.565429v1.full.pdf> (accessed 20.05.2024). <https://doi.org/10.1101/2023.11.04.565429>
20. Svensson H., Tyrchan C., Engkvist O., Chehreghani M. H. Utilizing Reinforcement learning for de novo drug design. *Machine-Mediated Learning*, 2024, vol. 113, no. 2, pp. 4811–4843. <https://doi.org/10.1007/s10994-024-06519-w>
21. Weininger D. SMILES, a chemical language and information system. *Journal of Chemical Information and Computer Sciences*, 1998, vol. 28, pp. 31–36. <https://doi.org/10.1021/ci00057a005>

22. Mnih V., Badia A. P., Mirza M., Graves A., Lillicrap T. P., ..., Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. *Proceedings of the 33rd International Conference on Machine Learning, New York, USA, 20–22 June 2016*, vol. 48, pp. 1928–1937. <https://proceedings.mlr.press/v48/mniha16.html>
23. Berman H. M., Battistuz T., Bhat T. N., Bluhm W. F., Bourne P., ..., Zardecki C. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 2002, vol. 58, no. 6, pp. 899–907. <https://doi.org/10.1093/nar/28.1.235>
24. Curreli F., Kwon Y. D., Nicolau I., Burgos G., Altieri A., ..., Debnath A.K. Antiviral activity and crystal structures of HIV-1 gp120 antagonists. *International Journal of Molecular Sciences*, 2022, vol. 23, no. 24, p. 15999. <https://doi.org/10.3390/ijms232415999>
25. Pettersen E. F., Goddard T. D., Huang C. C., Couch G. S., ..., Ferrin T. E. UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 2004, vol. 25, no. 13, pp. 1605–1612. <https://doi.org/10.1002/jcc.20084>
26. Kiefer F., Arnold K., Künzli M., Bordoli L., Schwede T. The Swiss-MODEL Repository and associated resources. *Nucleic Acids Research*, 2009, vol. 37, pp. 387–392. <https://doi.org/10.1093/nar/gkn750>
27. Benson M. L., Smith R. D., Khazanov N. A., Dimcheff B., Beaver J., ..., Carlson H. A. Binding MOAD, a high-quality protein–ligand database. *Nucleic Acids Research*, 2007, vol. 36, pp. 674–678. <https://doi.org/10.1093/nar/gkm911>
28. Wójcikowski M., Ballester P. J., Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, 2017, vol. 7, no. 1, pp. 1–10. <https://doi.org/10.1038/srep46710>
29. Lipinski C. A., Lombardo F., Dominy B. W., Feeney P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 2001, vol. 46, no. 1–3, pp. 3–26. [https://doi.org/10.1016/S0169-409X\(00\)00129-0](https://doi.org/10.1016/S0169-409X(00)00129-0)
30. Daina A., Michielin O., Zoete V. SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports*, 2017, vol. 7. Available at: <https://www.nature.com/articles/srep42717> (accessed 20.05.2024). <https://doi.org/10.1038/srep42717>
31. Veber D. F., Johnson S. R., Cheng H. Y., Smith B. R., Ward K. W., Kopple K. D. Molecular properties that influence the oral bioavailability of drug candidates. *Journal of Medicinal Chemistry*, 2002, vol. 45, no. 12, pp. 2615–2623. <https://doi.org/10.1021/jm020017n>

Информация об авторах

Воробьев Данила Александрович, стажер младшего научного сотрудника, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: daniel.vorobiov.2002@yandex.ru

Карпенко Анна Дмитриевна, научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: rfe.karpenko@gmail.com

Тузиков Александр Васильевич, член-корреспондент, доктор физико-математических наук, профессор, заведующий лабораторией математической кибернетики, Объединенный институт проблем информатики Национальной академии наук Беларуси.

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

Андрианов Александр Михайлович, доктор химических наук, профессор, главный научный сотрудник, Институт биоорганической химии Национальной академии наук Беларуси.

E-mail: alexande.andriano@yandex.ru

Information about the authors

Danila A. Varabyeu, Trainee Junior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: daniel.vorobiov.2002@yandex.ru

Anna D. Karpenko, Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: rfe.karpenko@gmail.com

Alexander V. Tuzikov, Corresponding Member, D. Sc. (Phys.-Math.), Prof., Head of the Laboratory of Mathematical Cybernetics, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

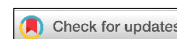
E-mail: tuzikov@newman.bas-net.by

Alexander M. Andrianov, D. Sc. (Chem.), Prof., Principal Researcher, Institute of Bioorganic Chemistry of the National Academy of Sciences of Belarus.

E-mail: alexande.andriano@yandex.ru

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

INFORMATION TECHNOLOGIES



УДК 004.912
<https://doi.org/10.37661/1816-0301-2024-21-3-94-105>

Оригинальная статья
Original Article

Обработка результатов веб-поиска в системе информационной поддержки процессов принятия решений

С. Ф. Липницкий, Л. В. Степура

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

Аннотация

Цели. Решается задача обработки результатов веб-поиска в системе информационной поддержки принятия решений с целью создания и коррекции содержательного описания проблемной ситуации. Предлагается подход к решению данной задачи на основе применения в качестве знаний о предметной области тематических корпусов текстов (совокупностей текстов по конкретной тематике), а также модели представления знаний на основе вербальных ассоциаций. При решении задач обработки результатов веб-поиска в системе информационной поддержки принятия решений преследуются пять основных целей: формирование расширенного описания проблемной ситуации, синтез поискового предписания, интернет-поиск информации о принятых решениях, синтез пересказа найденной информации, оценка качества найденных аналогов принятых решений.

Методы. Используются методы теории множеств, теории графов и математической лингвистики.

Результаты. Разработана математическая модель обработки результатов веб-поиска в системе информационной поддержки принятия решений. Формализованы понятия вербальной ассоциации слов и текстов, а также прагматически полной лексической структуры. Доказанные свойства таких структур обеспечивают алгоритмизацию информационных процессов в рассматриваемой модели.

Заключение. Подход к моделированию основывается на формализации понятий информативности слов, предложений, текстов и информативности вербальных ассоциаций между ними. В качестве реализации предложенной в статье модели разработаны алгоритмы создания словаря прагматически полных лексических структур, структурно-лексических шаблонов предложений, текстов и предметных областей, синтеза краткого пересказа найденной информации, оценки качества найденных аналогов принятых решений.

Ключевые слова: вербальные ассоциации, математическая модель, поисковое предписание, принятие решений, структурно-лексический шаблон

Для цитирования. Липницкий, С. Ф. Обработка результатов веб-поиска в системе информационной поддержки процессов принятия решений / С. Ф. Липницкий, Л. В. Степура // Информатика. – 2024. – Т. 21, № 3. – С. 94–105. <https://doi.org/10.37661/1816-0301-2024-21-3-94-105>

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 12.07.2024
Подписана в печать | Accepted 04.09.2024
Опубликована | Published 30.09.2024

Web search results processing in an information support system for decision-making processes

Stanislav F. Lipnitsky, Ludmila V. Stepura

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

Abstract

Objectives. The problem of the web search results processing in an information support system for decision-making is solved in order to create and correct a meaningful description of a problem situation. An approach to solving this problem is proposed based on the use of thematic text corpora (collections of texts on a specific topic) as knowledge about the subject area, as well as a knowledge representation model based on verbal associations. When solving problems of the web search results processing in a decision-making information support system, five main goals are pursued: the formation of an extended description of a problem situation, the synthesis of a search prescription, an Internet search for information about decisions made, the synthesis of a retelling of the information found, and an assessment of the quality of the found analogues of the decisions made.

Methods. Methods of set theory, graph theory and mathematical linguistics are used.

Results. A mathematical model has been developed for the web search results processing in an information support system for decision making. The concepts of verbal association of words and texts, as well as pragmatically complete lexical structure, are formalized. The proven properties of such structures provide algorithmization of information processes in the model under consideration.

Conclusion. The approach to modeling is based on the formalization of the concepts of the informativeness of words, sentences, texts and the informativeness of verbal associations between them. As an implementation of the model proposed in the article, algorithms have been developed for creating a dictionary of pragmatically complete lexical structures, creating structural-lexical templates for sentences, texts and subject areas, synthesizing a brief retelling of the information found, and assessing the quality of the found analogues of the decisions made.

Keywords: verbal associations, mathematical model, search prescription, decision making, structural-lexical template

For citation. Lipnitsky S. F., Stepura L. V. *Web search results processing in an information support system for decision-making processes*. *Informatika [Informatics]*, 2024, vol. 21, no. 3, pp. 94–105 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-94-105>

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

Введение. Процессы принятия решений в различных предметных областях имеют много общего и включают следующие основные этапы:

– содержательное описание проблемной ситуации. Под проблемной ситуацией понимают совокупность задач, требующих постановки и решения. Правильное ее описание является важнейшим условием принятия адекватного ситуации решения [1–5];

– сбор и анализ информации. Осуществляется поиск информации, которая находится в открытом доступе в Интернете. Проводится аналитическая обработка найденных сведений, в частности аналогов принятых решений по схожей тематике;

– коррекция описания проблемной ситуации. Корректирует описание пользователь (лицо, принимающее решение) на основе кратких пересказов найденных текстов и их оценок, генерируемых системой информационной поддержки;

– выявление вариантов (альтернатив) решения поставленной задачи. Альтернативы могут быть заданы на момент принятия решений, а могут быть неизвестны или частично известны;

– выбор критериев и оценка альтернатив. Критерии могут различаться по важности, определяемой соответствующим коэффициентом (весом);

– определение лучшей альтернативы;

– принятие решения с учетом результатов оценки альтернатив.

Реализация каждого этапа осуществляется, как правило, в итерационном режиме. В данной статье предлагается математическая модель обработки результатов веб-поиска с целью создания и коррекции содержательного описания проблемной ситуации. Формирование данного описания реализуется в режиме диалога с пользователем за пять шагов:

Шаг 1. Генерируется расширенное описание проблемной ситуации за счет пополнения исходного описания релевантными ему текстами.

Шаг 2. Формируется поисковое предписание путем индексирования исходного описания проблемной ситуации с привлечением статистических характеристик слов из расширенного описания.

Шаг 3. Осуществляется интернет-поиск информации по сформированному поисковому предписанию.

Шаг 4. Проводится аналитическая обработка результатов интернет-поиска и создаются краткие пересказы найденных текстов и их оценки.

Шаг 5. Пользователь корректирует описание проблемной ситуации.

Шаги 2–5 данного алгоритма могут повторяться по запросу пользователя.

При реализации процессов формирования и коррекции описания проблемной ситуации используются модели данных и знаний, предложенные в статье [6].

Процесс обработки результатов веб-поиска в системе информационной поддержки принятия решений состоит из пяти основных этапов, которые будут описаны в данной статье: формирования расширенного описания проблемной ситуации, синтеза поискового предписания, интернет-поиска информации о принятых решениях, синтеза пересказа найденной информации, оценки качества найденных аналогов принятых решений.

1. Формирование расширенного описания проблемной ситуации. Основой для расширения описания проблемной ситуации являются корпуса текстов. Под корпусом текстов в лингвистике понимают совокупность документов, накопленных и размеченных по определенным правилам в зависимости от назначения. В случае отсутствия разметки эти совокупности называют корпусами текстов первого порядка. Будем различать тематические и полные корпуса текстов. Тематический корпус – это множество неразмеченных текстов по некоторой конкретной тематике. Объединение всех тематических корпусов образует полный корпус текстов. Исходное описание проблемной ситуации будем дополнять текстами из полного корпуса, которые семантически связаны с этим описанием. Такие семантические связи называют вербальными ассоциациями.

Пусть Ct_i ($i = \overline{1, n}$; $n \geq 1$) – тематические корпуса текстов, а $Cf = \bigcup_{i=1}^n Ct_i$ – полный корпус, объединяющий все тематические. Обозначим через W множество всех словоформ корпуса Cf , а через \prec_w – отношение строгого порядка на W_{Cf} (транзитивное и антирефлексивное бинарное отношение). Определим, кроме того, на множестве W антирефлексивное и антисимметричное бинарное отношение Θ , такое, что любая ориентированная пара слов (a, b) из множества W является элементом отношения Θ тогда и только тогда, когда слова a и b из этой пары содержатся хотя бы в одном предложении корпуса Cf и выполняется соотношение $a \prec_w b$. Отношение Θ назовем отношением вербальной ассоциации слов в полном корпусе текстов Cf .

1.1. Информативность вербальной ассоциации текстов. Обозначим через P описание проблемной ситуации, а через T – произвольный текст из полного корпуса текстов Cf . Рассмотрим l -мерное евклидово пространство E . Для его построения лексикографически упорядочим все пары словоформ из корпуса Cf , т. е. сформируем кортеж $\langle (a_1, b_1), (a_2, b_2), \dots, (a_l, b_l) \rangle$. Пусть W_P и W_T – множества словоформ в текстах P и T соответственно, дополненные всеми синонимами и словоизменениями из следующих словарей Dic_{par} и Dic_{syn} :

$Dic_{par} = \{(a, Par_a) \mid a \in W_{Cf}, a \in Par_a\}$ – словарь словоизменительных парадигм, состоящий из пар $\langle \text{словоформа}, \text{парадигма} \rangle$. В позиции парадигмы Par_a представлены все словоизменения каждой словоформы a из множества всех словоформ W_{Cf} полного корпуса текстов;

$Dic_{syn} = \{(a, Syn_a) \mid a \in W_{Cf}, a \in Syn_a\}$ – словарь синонимичных словоформ, включающий в себя пары *⟨словоформа, синонимичные словоформы⟩*, в которых каждой словоформе a соответствует множество ее синонимов Syn_a .

Введем вектор в пространстве E :

$$\mathbf{I}^{PT} = (I^{a_1b_1}, I^{a_2b_2}, \dots, I^{a_l b_l}). \quad (1)$$

Если словоформы a_i и b_i содержатся соответственно в множествах W_P и W_T , то значение информативности $I^{a_i b_i}$ согласно [7] в формуле (1) определяется из следующего словаря вербально-ассоциативных пар слов:

$$Dic_{ab} = \{ \langle (a, b), I^{ab} \rangle \mid a, b \in \pi, \pi \in Cf \}.$$

В противном случае $I^{a_i b_i} = 0$.

С учетом принятых соглашений естественно предположить, что информативность вербальной ассоциации между текстами P и T – это длина вектора (1). Для сравнения значений информативности вербальной ассоциации различных пар текстов необходима нормализация данного понятия. С целью нормализации рассмотрим вектор, все компоненты которого равны единице. Тогда нормализованную информативность I^{PT} можно интерпретировать как проекцию вектора $\mathbf{e} = (1, 1, \dots, 1)$ размерности l на направление вектора \mathbf{I}^{PT} [5], т. е. отношение скалярного произведения векторов \mathbf{I}^{PT} и \mathbf{e} к длине вектора \mathbf{I}^{PT} :

$$I^{PT} = \frac{\mathbf{I}^{PT} \cdot \mathbf{e}}{|\mathbf{I}^{PT}|} = \frac{I_1 + I_2 + \dots + I_r}{\sqrt{(I_1)^2 + (I_2)^2 + \dots + (I_r)^2}}, \quad (2)$$

где I_1, I_2, \dots, I_r – все отличные от нуля координаты вектора \mathbf{I}^{PT} .

Если значение информативности (2) превышает некоторое пороговое значение, то описание проблемной ситуации P расширяется путем присоединения к нему всех предложений текста T . При дальнейшем присоединении текстов описание расширяется до получения статистически значимого количества предложений в нем. Обозначим через Q расширенное описание проблемной ситуации.

1.2. Информативность вербальной ассоциации слов. Будем считать, что информативность I^{ab} вербальной ассоциации между произвольными словами a и b , так же как и в формуле (1) для словоформ, – это вероятность появления в полном корпусе текстов предложения, содержащего слова a и b . При практической реализации информационной системы под указанной информативностью будем понимать дробь [8]

$$I^{ab} = n_{Cf}^{ab} / n_{Cf}, \quad (3)$$

где n_{Cf}^{ab} – количество предложений в полном корпусе текстов Cf , в которых одновременно присутствуют слова a и b или синонимы и словоизменения хотя бы одного из этих слов, а n_{Cf} – количество всех предложений в корпусе Cf .

В развернутом виде формулу (3) перепишем, используя информацию, которую содержат словари Dic_{par} и Dic_{syn} :

$$I^{ab} = \frac{n_{Cf}^{ab} + n^{Par_{ab}} + n^{Syn_{ab}}}{n_{Cf}}. \quad (4)$$

Параметр $n^{Par_{ab}}$ в формуле (4) указывает на число вхождений всех пар словоформ, являющихся словоизменениями соответственно слов a и (или) b и встречающихся в одном и том же предложении корпуса текстов Cf :

$$n^{Par_{ab}} = \sum_{c \in Par_a, d \in Par_b} n_{Cf}^{cd}.$$

Аналогичное выражение справедливо для параметра $n^{Syn_{ab}}$:

$$n^{Syn_{ab}} = \sum_{d \in Syn_a, f \in Syn_b} n_{Cf}^{df}.$$

2. Синтез поискового предписания. Из исходного описания проблемной ситуации P формируется поисковое предписание в виде

$$\text{ПП}_P = \{(a, I_Q^a); (a, I_Q^b) \dots | a, b \in P, I_Q^a \geq I^0, I_Q^b \geq I^0 \dots\}, \quad (5)$$

где I_Q^a, \dots – значения информативности слов a, b, \dots из описания P , I^0 – пороговое значение информативности слова в поисковом предписании, т. е. в поисковое предписание включаются все слова из текста P , информативность которых превышает пороговое значение I^0 .

2.1. Информативность слов. Информативность I_Q^a каждого слова a из текста Q вычислим по формуле

$$I_Q^a = n_Q^a / n_{Cf}^a, \quad (6)$$

где n_Q^a и n_{Cf}^a – частоты встречаемости (с учетом словоизменения и синонимии) словоформы a в тексте Q и полном корпусе текстов Cf соответственно [5]. Таким образом, при вычислении информативности любого слова a из текста P в силу формулы (6) используется частота n_Q^a этого слова в расширенном описании проблемной ситуации, т. е. в тексте Q . При вычислении информативности I_Q^a будем использовать частотный словарь словоформ

$$Dic_a = \{(a, n_{Cf}^a, n_{Ct_1}^a, n_{Ct_2}^a, \dots, n_{Ct_n}^a) | a \in W_{Cf}\},$$

в котором каждой словоформе из множества W_{Cf} всех словоформ корпуса Cf приписаны частоты ее встречаемости $n_{Cf}^a, n_{Ct_1}^a, n_{Ct_2}^a, \dots, n_{Ct_n}^a$ во всех тематических корпусах текстов Ct_i ($i = \overline{1, n}$; $n \geq 1$).

Учитывая словоизменения и синонимию, зафиксированные в лингвистических словарях Dic_{par} и Dic_{syn} , формулу (6) перепишем в виде

$$I_Q^a = \frac{n_Q^a + n_Q^{Par_a} + n_Q^{Syn_a}}{n_{Cf}^a + n_{Cf}^{Par_a} + n_{Cf}^{Syn_a}}. \quad (7)$$

Смысл параметров $n_Q^{Par_a}, n_{Cf}^{Par_a}, n_Q^{Syn_a}$ и $n_{Cf}^{Syn_a}$ аналогичен их смыслу в выражении (4).

2.2. Информативность текстов. При вычислении информативности любого текста M , являющегося фрагментом некоторого текста L , будем также исходить из его векторного представления $\mathbf{I}_L^M = (I_L^{a_1}, I_L^{a_2}, \dots, I_L^{a_n})$, где $I_L^{a_1}, I_L^{a_2}, \dots, I_L^{a_n}$ – значения информативности слов текста M (компонента вектора \mathbf{I}_L^M равна нулю, если соответствующего слова нет в тексте M). Тогда нор-

мализованную информативность I_L^M текста M будем вычислять по формуле, аналогичной выражению (2):

$$I_L^M = \frac{I_1 + I_2 + \dots + I_q}{\sqrt{(I_1)^2 + (I_2)^2 + \dots + (I_q)^2}}, \quad (8)$$

где I_1, I_2, \dots, I_q – значения информативности слов текста M .

3. Интернет-поиск информации о принятых решениях. Поиск проводится с целью выявления общих подходов к решению задачи и реализации синтеза кратких пересказов найденных текстов и их оценок.

Обозначим через R m -мерное евклидово пространство ($m = |W_{Cf}|$). Для каждой веб-страницы S построим вектор ее поискового образа в пространстве R : $\mathbf{F}_S = (I_S^{a_1}, I_S^{a_2}, \dots, I_S^{a_m})$. Аналогично запишем вектор поискового предписания (5): $\mathbf{F}_{\text{ПП}_p} = (I_{\text{ПП}_p}^{b_1}, I_{\text{ПП}_p}^{b_2}, \dots, I_{\text{ПП}_p}^{b_m})$. Тогда для поиска веб-страниц по поисковому предписанию $\mathbf{F}_{\text{ПП}_p}$ в качестве критерия выдачи, как показано в работе [9], используем косинус угла φ между векторами \mathbf{F}_S и $\mathbf{F}_{\text{ПП}_p}$:

$$\cos \varphi = \frac{\mathbf{F}_S \cdot \mathbf{F}_{\text{ПП}_p}}{|\mathbf{F}_S| \cdot |\mathbf{F}_{\text{ПП}_p}|} = \frac{\sum_{j=1}^m I_S^{a_j} I_{\text{ПП}_p}^{b_j}}{\sqrt{\sum_{j=1}^m (I_S^{a_j})^2} \cdot \sqrt{\sum_{i=1}^m (I_{\text{ПП}_p}^{b_j})^2}}. \quad (9)$$

Все найденные веб-страницы упорядочиваются по убыванию значения (8).

4. Синтез пересказа найденной информации. Обычно синтез речи рассматривается как процесс последовательной генерации морфем, лексем, синтаксических фраз и, наконец, предложений. Возможен также подход к синтезу предложений, основанный на использовании языковой памяти человека [10]. Согласно этой концепции фразы при синтезе речи строятся из готовых хранящихся в памяти так называемых коммуникативных фрагментов. В случае синтеза текстовых сообщений будем использовать понятие прагматически полной лексической структуры (ПП-структуры).

4.1. Определение понятия прагматически полной лексической структуры. Пусть $\pi = a_1 a_2 \dots a_n$ – произвольное предложение (или подцепочка предложения) некоторого текстового документа D . Определим формально понятие ПП-структуры:

1. Если $n = 1$, то слово a_1 цепочки π назовем ПП-структурой.
2. Если для всех m ($2 < m < n$) справедливы неравенства $I_D^{(a_1 a_2 \dots a_{m-1}) a_m} \geq I_D^{00}$ и $I_D^{(a_1 a_2 \dots a_m) a_{m+1}} < I_D^{00}$, то цепочку $a_1 a_2 \dots a_m$ назовем ПП-структурой. Значения информативности вычисляются по формуле (2).
3. Если для всех $n \geq 2$ выполняется неравенство $I_D^{(a_1 a_2 \dots a_{n-1}) a_n} \geq I_D^{00}$, то цепочку $a_1 a_2 \dots a_n$ назовем ПП-структурой.

В соответствии с данным определением работает алгоритм разбиения предложений на ПП-структуры и формирования соответствующего словаря:

Шаг 1. Проверяется условие 1. Если оно выполняется, то слово a_1 помещается в словарь ПП-структур и исключается из цепочки π . Оставшимся словам приписываются последовательно новые индексы, начиная с единицы.

Шаг 2. Проверяется условие 2. Если цепочка $a_1 a_2 \dots a_m$ является ПП-структурой, то она заносится в словарь ПП-структур и исключается из цепочки π . Оставшимся словам приписываются последовательно новые индексы, начиная с единицы.

Шаг 3. Проверяется условие 3. Если цепочка $a_1 a_2 \dots a_n$ является ПП-структурой, то она заносится в словарь ПП-структур.

Словарем ПП-структур будем называть множество

$$Dic_{str} = \{ \langle str, I_{Ct_1}^{str}, I_{Ct_2}^{str}, \dots, I_{Ct_n}^{str} \rangle \mid str \in Str \},$$

где Str – совокупность ПП-структур. Значения информативности $I_{Ct_i}^{str}$ в словаре Dic_{str} вычисляются по формуле (8).

Будем различать базовые и связующие ПП-структуры. Пусть Ct – некоторый тематический корпус текстов. Рассмотрим предметную область, определяемую корпусом Ct . Обозначим через I_{Ct}^0 пороговое значение информативности ПП-структуры. Тогда ПП-структуру str будем называть базовой, если значение ее информативности I_{Ct}^{str} удовлетворяет неравенству $I_{Ct}^{str} \geq I_{Ct}^0$. Если же $I_{Ct}^{str} < I_{Ct}^0$, то ПП-структуру str назовем связующей. Связующей, например, является ПП-структура «предлагается новый подход к решению проблемы», а базовой – ПП-структура «принятие решений в условиях неопределенности».

Обозначим через $Str_{баз.}$ множество всех базовых ПП-структур, а через $Str_{св.}$ – множество всех связующих. Тогда множество всех ПП-структур предметной области – это объединение множеств базовых и связующих ПП-структур, т. е. $Str_{Ct} = Str_{баз.} \cup Str_{св.}$.

4.2. Вербальная ассоциация прагматически полных лексических структур. Эффективность алгоритмов аналитической обработки текстовой информации существенным образом зависит от их интеллектуальности, т. е. способности работать не только с данными, но и знаниями об объектах и явлениях предметной области. При автоматизации процесса пересказа текста необходимые знания накапливаются в базе знаний о предметной области. Построим модель представления таких знаний.

Определим на множестве Str_{Ct} отношение толерантности Δ (рефлексивное и симметричное бинарное отношение), такое, что неупорядоченная пара (f, g) любых ПП-структур из множества Str_{Ct} является элементом отношения Δ , т. е. $(f, g) \in \Delta$ тогда и только тогда, когда ПП-структуры f и g из этой пары содержатся хотя бы в одном предложении корпуса Ct . Отношение Δ будем называть *вербально-ассоциативным отношением ПП-структур предметной области*, определяемой тематическим корпусом текстов Ct .

Вербально-ассоциативное отношение Δ – это отношение вербально-ассоциативной связи ПП-структур в тематическом корпусе текстов Ct .

Пусть f и g – произвольные ПП-структуры предметной области, определяемой тематическим корпусом текстов Ct . Для вычисления информативности вербальной ассоциации между ПП-структурами f и g будем использовать формулу (2).

4.3. Отношение конкатенации прагматически полных лексических структур. Для получения «хороших» предложений при их синтезе из ПП-структур будем использовать отношение их конкатенации («склейки», объединения). Понятие этого отношения введем следующим образом.

Определим на множестве Str_{Ct} всех ПП-структур в тематическом корпусе текстов Ct анти-рефлексивное бинарное отношение Λ , такое, что для любых фрагментов $f, g \in Str_{Ct}$ соотношение $(f, g) \in \Lambda$ выполняется тогда и только тогда, когда в некотором тексте $F_T \in Ct$ существует предложение π , в котором ПП-структура f непосредственно предшествует ПП-структуре g . Отношение Λ будем называть *отношением конкатенации ПП-структур* в тематическом корпусе текстов Ct .

Элементами конкатенации являются, например, следующие пары ПП-структур: «используется при» и «капитальном строительстве», «отложен ежегодный визит» и «председателя объединения».

Упорядоченные пары $(f, g) \in \Lambda$ будем хранить в специальном списке – словаре конкатенации ПП-структур:

$$Dic_{fg} = \{ (f, g) \mid f, g \in Ft, (f, g) \in \Lambda \}. \quad (10)$$

Основой для формирования словаря Dic_{fg} являются несовпадающие предложения полного корпуса текстов, представленные в виде цепочек, которые состоят из ПП-структур.

4.4. Вербально-ассоциативная сеть предметной области. Рассмотрим граф вербально-ассоциативного отношения Δ . Пометим каждую вершину f этого графа значением информативности I_{Ct}^f ПП-структуры (с учетом синонимии и словоизменения), а каждое ребро (f, g) – значением информативности I_{Ct}^{fg} вербальной ассоциации ПП-структур f и g (также учитывая синонимии и словоизменения). Пусть (f, g) – произвольное ребро этого графа. Если $(f, g) \in \Delta$, то для всех таких пар (f, g) вершины f и g соединим дугой, направленной от f к g . Обозначим полученный смешанный граф через Net_{Ct} .

Граф Net_{Ct} назовем *вербально-ассоциативной сетью предметной области*, определяемой тематическим корпусом текстов Ct . Сеть Net_{Ct} является моделью поискового образа тематического корпуса текстов Ct . Построим этот поисковый образ в виде множества ПП-структур и вербально-ассоциативных пар таких структур, в котором им приписаны значения информативности, а парам – значения информативности вербальной ассоциации:

$$PO_{Ct} = \{(f, I_{Ct}^f), \dots, ((g, h), I_{Ct}^{gh}), \dots, ((\overline{p, q}), I_{Ct}^{pq}), \dots \mid I_{Ct}^f > I_{Ct}^0; I_{Ct}^{gh}, I_{Ct}^{pq} > I_{Ct}^{00}\}, \quad (11)$$

где I_{Ct}^0, I_{Ct}^{00} – пороговые значения информативности ПП-структуры и вербальной ассоциации ПП-структур соответственно, стрелка над парой ПП-структур (p, q) означает, что $(p, q) \in \Delta$.

4.5. Вербально-ассоциативная сеть пересказываемого текста. Пусть $Q \in Cf$ – некоторый текст, а Ct – релевантный ему тематический корпус текстов. Вычислим информативность I_Q^π каждого предложения π текста Q по формуле (9).

Обозначим через I_Q^0 некоторое пороговое значение информативности предложений текста Q . Если $I_Q^\pi \geq I_Q^0$, то предложение π будем считать информативным. Исключив из текста все неинформативные предложения, т. е. такие, для которых $I_Q^\pi < I_Q^0$, получим кортеж информативных предложений $T = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$. Каждое предложение π_i ($i = \overline{1, m}$) представим в виде цепочки ПП-структур $\pi_i = f_1 f_2 \dots$. Обозначим через Δ_T сужение вербально-ассоциативного отношения Δ на множестве F_T всех ПП-структур кортежа предложений T , т. е. $\Delta_T = \Delta \cap (F_T \times F_T)$, а через Λ_T – сужение отношения конкатенации ПП-структур Λ на этом же множестве. Тогда вербально-ассоциативную сеть пересказываемого текста Q построим следующим образом.

Рассмотрим вербально-ассоциативную сеть предметной области Net . Исключим из сети Net все ребра (f, g) , такие, что $(f, g) \notin \Delta_T$, и все дуги (r, s) , для которых $(r, s) \notin \Lambda_T$. Исключим также из сети Net инцидентные исключенным ребрам и дугам вершины. Полученный граф назовем *вербально-ассоциативной сетью пересказываемого текста Q*.

Для моделирования синтеза пересказа текстов введем понятия структурно-лексических шаблонов предложения, текста и предметной области.

4.6. Структурно-лексический шаблон предложения. Пусть имеется предложение $\pi = f_1 f_2 \dots f_l$. Цепочку, полученную из предложения π заменой его базовых коммуникативных фрагментов слотами («пустыми» фрагментами), будем называть структурно-лексическим шаблоном предложения π .

Структурно-лексические шаблоны предложений создаются в автоматизированном режиме: сначала на основе специально подготовленных текстов программно формируется совокупность структурно-лексических шаблонов, а затем они корректируются экспертом-лингвистом информационной системы. При синтезе предложения слоты заменяются ПП-структурами.

4.7. Структурно-лексический шаблон текста. Пусть имеется текст в виде кортежа предложений $T = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$, такой, что каждому предложению π_i ($i = \overline{1, m}$) соответствует его структурно-лексический шаблон H_i . Рассмотрим кортеж структурно-лексических шаблонов

предложений $SH_T = \langle H_1, H_2, \dots, H_m \rangle$. В качестве характеристики связности структурно-лексических шаблонов предложений текста T определим на множестве SH_T антирефлексивное бинарное отношение Ω_T , элементами которого являются пары соседних структурно-лексических шаблонов предложений из множества SH_T , т. е. $\Omega_T = \{(H_i, H_{i+1}) | i = \overline{1, m-1}\}$. Отношение Ω_T назовем структурно-лексическим шаблоном текста T .

4.8. Структурно-лексический шаблон предметной области. Для формирования структурно-лексического шаблона предметной области необходимо предварительно подготовить множество $\{T_i | i = \overline{1, r}\}$ некоторых «хороших» текстов. Обозначим через Ω_{T_i} структурно-лексический шаблон текста T_i . Тогда объединение множеств $\Omega_{C_t} = \bigcup_{i=1}^r \Omega_{T_i}$ назовем структурно-лексическим шаблоном предметной области, определяемой тематическим корпусом текстов C_t .

4.9. Алгоритм синтеза краткого пересказа найденной информации. Пусть $T \in Cf$ – некоторый текст, C_t – релевантный ему тематический корпус текстов, а $Q = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$ – кортеж информативных предложений этого текста. Алгоритм синтеза пересказа содержания текста T работает следующим образом:

Шаг 1. Из вербально-ассоциативной сети предметной области Net_{C_t} исключаются все вершины, которые соответствуют базовым ПП-структурам, отсутствующим в предложениях кортежа Q . Из сети Net_{C_t} удаляются инцидентные исключенным вершинам ребра и дуги. Полученный граф обозначим через Net_T^+ .

Шаг 2. Из структурно-лексического шаблона предметной области Ω_{C_t} формируется множество начальных структурно-лексических шаблонов предложений $Form_1 = \{H_1, H_2, \dots\}$. Для каждого шаблона из множества $Form_1$ строится его вербально-ассоциативная сеть Net_{H_1} . Формируется множество всех орцепей $Req_{H_1}^{1,2}$ длиной 1 и 2 графа Net_{H_1} . Элементами множества $Req_{H_1}^{1,2}$ являются орцепи вида $f_1s_1, s_2g_2, f_3s_3g_3$, где f_1s_1 – конечные фрагменты предложения, а s_2g_2 – начальные; s_1, s_2 и s_3 – слоты; f_3 и g_3 – ПП-структуры, не являющиеся начальными и конечными в структурно-лексическом шаблоне H_1 . Орцепи из множества $Req_{H_1}^{1,2}$ служат запросами на поиск релевантных орцепей в графе Net_T^+ с целью заполнения найденных слотов ПП-структурами. Орцепи графов Net_T^+ и Net_{H_1} считаются совпавшими, если совпали соответствующие ПП-структуры. Например, совпавшими являются орцепи fsg и frg , где r – ПП-структура, заполняющая слот s . Если все слоты структурно-лексического шаблона заполнены, то полученное в результате предложение включается в некоторое множество Sen . Далее описанная процедура повторяется для всех остальных шаблонов из множества $Form_1$.

Шаг 3. В сформированном множестве Sen проводится поиск релевантного графу Net_T^+ предложения π_i , полученного из шаблона H_i . Оно является началом формируемого пересказа текста.

Шаг 4. Создается множество $Form_2$, состоящее из всех структурно-лексических шаблонов предложений H , таких, что $(H_i, H) \in \Omega_T$. Процессы заполнения слотами шаблонов из множеств $Form_2, Form_3$ и т. д. аналогичны такой процедуре для шаблонов из множества $Form_1$.

Пример

Принимается решение о включении некоторой статьи в отраслевую энциклопедию. На этапе поиска информации о принятых решениях найден приведенный ниже текстовый фрагмент. Требуется построить краткий его пересказ.

В ходе интерпретации воссоздается мысленный мир, в котором по презумпции интерпретатора автор конструировал дискурс и в котором описывается реальное или нереальное положение дел. При этом анализ дискурса предполагает наличие языкового инструментария, при котором исследователь обращается не только к собственным лингвистическим знаниям, но и к общему фоновому знанию

о реальном мире, поскольку в процессах понимания и порождения речи взаимодействуют все базы данных, хранящиеся в когнитивном аппарате человека. В основном анализу подвергаются не отдельные слова, а более крупные объединения (предложения или даже целые тексты), так как известно, что трансляция смысла ведется с помощью именно текстов. Поэтому текст стал объектом исследования отдельного направления языкознания, лингвистики текста, которое стремится выйти за рамки предложения. Дискурс может разделиться на высказывания, в то время как существуют другие объединения, которые складываются из последовательных предложений, например текст.

На начальном этапе генерации пересказа данного текста формируется множество структурно-лексических шаблонов предложений $Form_1$: $Form_1 = \{ \langle \text{обсуждается проблема} / \diamond \rangle, \langle \text{рассматриваются вопросы} / \diamond / \text{ путем использования} \diamond \rangle, \langle \text{речь идет о} / \diamond \rangle, \langle \text{в работе} / \text{приведены результаты} / \rangle, \dots \}$, где символом «/» обозначены разделители между коммуникативными фрагментами, а символом « \diamond » – слоты.

После построения для всех структурно-лексических шаблонов их вербально-ассоциативных сетей и поиска ПП-структур для заполнения слотов выбранного шаблона получим начальное предложение пересказа исходного текста в виде цепочки ПП-структур: *Рассматриваются вопросы / конструирования дискурса / путем использования / лингвистических знаний / о реальном мире.*

На последующих этапах синтеза выходного текста процесс поиска структурно-лексических шаблонов предложений и заполнения их слотов повторяется аналогичным образом. В результате получим следующие предложения сформированного краткого пересказа текста: *Для изучения дискурса / возникло направление / в языкознании / – / лингвистика текста. При конструировании / анализируются фрагменты / данного дискурса. Элементами дискурса / являются высказывания.*

5. Оценка качества найденных аналогов принятых решений. Для оценки качества принятых решений будем использовать совокупность оценочных тематических корпусов текстов. Каждому корпусу соответствует некоторая оценка качества. При n -балльной шкале оценок количество таких корпусов должно быть равно n . Всякий корпус включает текстовые документы одинакового оценочного качества. В простейшем случае формируются два корпуса текстов с соответствующей оценочной лексикой. Первый корпус создается для анализа положительного качества, а второй – для анализа отрицательного. Оценка качества принятых решений реализуется путем поиска наиболее релевантных им оценочных корпусов текстов.

Пусть St – тематический корпус текстов с оценочной лексикой, состоящий из n (по числу оценок в шкале оценивания) подкорпусов, т. е. $St = \{ Cp_i \mid i = \overline{1, n} \}$. Каждый подкорпус Cp_i состоит из текстов с одинаковой оценкой качества и представляет собой пару $\langle Cp_i, Ev_i \rangle$ (Ev_i – оценка качества для всех текстов из множества Cp_i). Пусть также T – текстовое сообщение, полученное в результате формирования контекстного окружения к некоторому найденному на веб-странице $S_{\text{всб}}$ информативному предложению. Построим вектор F_T сообщения T и векторы F_{Cp_i} ($i = \overline{1, n}$). Для каждой пары (F_T, F_{Cp_i}) вычислим косинус угла между этими векторами по формуле (7). Тогда сообщению T будет соответствовать оценка качества Ev_i при таком значении i , при котором $\cos(F_T, F_{Cp_i})$ принимает наибольшее значение.

Заключение. Предложена математическая модель аналитической обработки результатов веб-поиска в системе информационной поддержки процессов принятия решений на этапе текстового описания проблемной ситуации. Подход к моделированию основывается на формализации понятий информативности слов, предложений и текстов и информативности вербальных ассоциаций между ними. Формализация этих понятий обеспечила реализацию следующих алгоритмов:

- создания словаря прагматически полных лексических структур;
- создания структурно-лексических шаблонов предложений, текстов и предметных областей;
- синтеза краткого пересказа найденной информации;
- оценки качества найденных аналогов принятых решений.

Словарь прагматически полных лексических структур создается в три шага в соответствии с их формальным определением. На каждом шаге последовательно проверяются условия из данного определения.

Фрагментно-слововые шаблоны предложений формируются путем замены их базовых прагматически полных лексических структур слотами, а шаблоны текстов – как кортежи шаблонов их предложений.

Структурно-лексические шаблоны предметных областей создаются в виде реализации бинарных отношений на множествах шаблонов предложений из соответствующих тематических корпусов текстов. Каждый тематический корпус текстов определяет некоторую предметную область.

Вклад авторов. С. Ф. Липницкий предложил формальную модель обработки результатов веб-поиска в системе информационной поддержки принятия решений, Л. В. Степура разработала алгоритмы обработки текстовой информации и представила результаты исследования. Оба автора принимали участие в подготовке текста статьи.

Список использованных источников

1. Кравченко, Т. К. Системы поддержки принятия решений / Т. К. Кравченко // Информационные технологии для современного университета / под общ. ред.: А. Н. Тихонов, А. Д. Иванников. – М. : ГНИИ ИТТ «Информика», 2011. – С. 107–118.
2. Моисеенко, Е. В. Информационные технологии в экономике / Е. В. Моисеенко, Е. Г. Лаврушина ; ред.: М. А. Касаткина – М. : Софт, 2009. – С. 120–135.
3. Симанков, В. С. Методологическое обеспечение этапов поддержки принятия решений при синтезе сложных систем / В. С. Симанков, А. Н. Черкасов // Перспективы науки. – 2012. – № 12. – С. 85–89.
4. Липницкий, С. Ф. Описание проблемной ситуации в процессе информационной поддержки принятия решений / С. Ф. Липницкий, Л. В. Степура // Докл. XXI Междунар. конф. «Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2022)», Минск, 18 нояб. 2022 г. – Минск : ОИПИ НАН Беларуси, 2022. – С. 108–112.
5. Липницкий, С. Ф. Поиск и лексико-семантическая обработка научно-технической информации / С. Ф. Липницкий, Л. В. Степура // Докл. XXII Междунар. конф. «Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2023)», Минск, 16 нояб. 2023 г. – Минск : ОИПИ НАН Беларуси, 2023. – С. 181–185.
6. Липницкий, С. Ф. Модель представления знаний в информационных системах на основе вербальных ассоциаций / С. Ф. Липницкий // Информатика. – 2011. – № 4. – С. 21–28.
7. Буравкин, А. Г. Интернет-поиск альтернативных вариантов в процессе принятия решений / А. Г. Буравкин, С. Ф. Липницкий, Л. В. Степура // Докл. XX Междунар. конф. «Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2021)», Минск, 18 нояб. 2021 г. – Минск : ОИПИ НАН Беларуси, 2021. – С. 180–183.
8. Липницкий, С. Ф. Коррекция запросов в системе информационной поддержки принятия решений / С. Ф. Липницкий // Информатика. – 2023. – Т. 20, № 2. – С. 85–95. <https://doi.org/10.37661/1816-0301-2023-20-2-85-95>
9. Липницкий, С. Ф. Моделирование информационного поиска на основе динамических корпусов текстов / С. Ф. Липницкий, А. А. Мамчич // Вес. Нац. акад. навук Беларусі. Сер. фіз.-тэхн. навук. – 2011. – № 1. – С. 72–81.
10. Гаспаров, Б. М. Язык, память, образ. Лингвистика языкового существования / Б. М. Гаспаров. – М. : Новое литературное обозрение, 1996. – 352 с.

References

1. Kravchenko T. K. *Decision support systems*. Informacionnyye tehnologii dlja sovremennogo universiteta [*Information Technologies for a Modern University*]. In A. N. Tikhonov, A. D. Ivannikov (eds.). Moscow, GNIИ ИТТ "Informika", 2011, pp. 107–118 (In Russ.).
2. Moiseenko E. V., Lavrushina E. G. Informacionnyye tehnologii v jekonomike. *Information Technologies in the Economy*. In M. A. Kasatkina (ed.). Moscow, Soft, 2009, pp. 120–135 (In Russ.).
3. Simankov V. S., Cherkasov A. N. Methodological support of decision support stages in the synthesis of complex systems. *Perspektivy nauki [Prospects of Science]*, 2012, no. 12, pp. 85–89 (In Russ.).

4. Lipnitsky S. F., Stepura L. V. *Description of a problem situation in the process of information support for decision making*. Doklady XXI Mezhdunarodnoj konferencii «Razvitie informatizacii i gosudarstvennoj sistemy nauchno-tehnicheskoy informacii (RINTI-2022)», Minsk, 18 nojabrja 2022 g. [*Reports of the XXI International Conference "Development of Informatization and the State System of Scientific and Technical Information (RINTI-2022)", Minsk, 18 November 2022*]. Minsk, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 2022, pp. 108–112 (In Russ.).
5. Lipnitsky S. F., Stepura L. V. *Search and lexical-semantic processing of scientific and technical information*. Doklady XXII Mezhdunarodnoj konferencii «Razvitie informatizacii i gosudarstvennoj sistemy nauchno-tehnicheskoy informacii (RINTI-2023)», Minsk, 16 nojabrja 2023 g. [*Reports of the XXII International Conference "Development of Informatization and the State System of Scientific and Technical Information (RINTI-2023)", Minsk, 16 November 2023*]. Minsk, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 2023, pp. 181–185 (In Russ.).
6. Lipnitsky S. F. *Model of knowledge representation in information systems based on verbal associations*. Informatika [Informatics], 2011, no. 4, pp. 21–28 (In Russ.).
7. Buravkin A. G., Lipnitsky S. F., Stepura L. V. *Internet search for alternative options in the decision-making process*. Doklady XX Mezhdunarodnoj konferencii «Razvitie informatizacii i gosudarstvennoj sistemy nauchno-tehnicheskoy informacii (RINTI-2021)», Minsk, 18 nojabrja 2021 g. [*Reports of the XX International Conference "Development of Informatization and the State System of Scientific and Technical Information (RINTI-2021)", Minsk, 18 November 2021*]. Minsk, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 2021, pp. 180–183 (In Russ.).
8. Lipnitsky S. F. *Correction of queries in the information support system for decision-making*. Informatika [Informatics], 2023, vol. 20, no. 2, pp. 85–95 (In Russ.). <https://doi.org/10.37661/1816-0301-2023-20-2-85-95>
9. Lipnitsky S. F., Mamchich A. A. *Modeling information retrieval based on dynamic text corpora*. Vesci Nacyjanal'naj akademii navuk Belarusi. Seryja fizika-tjechnichnyh navuk [*Proceedings of the National Academy of Sciences of Belarus. Physical-technical Series*], 2011, no. 1, pp. 72–81 (In Russ.).
10. Gasparov B. M. Jazyk, pamjat', obraz. Lingvistika jazykovogo sushhestvovanija. *Language, Memory, Image. Linguistics of Linguistic Existence*. Moscow, Novoe literaturnoe obozrenie, 1996, 352 p. (In Russ.).

Информация об авторах

Липницкий Станислав Феликсович, доктор технических наук, главный научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: lipn@newman.bas-net.by

Степура Людмила Васильевна, научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: stepura@newman.bas-net.by

Information about the authors

Stanislav F. Lipnitsky, D. Sc. (Eng.), Chief Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: lipn@newman.bas-net.by

Ludmila V. Stepura, Researcher Scientist, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: stepura@newman.bas-net.by

УЧЕНЫЕ БЕЛАРУСИ SCIENTISTS OF BELARUS

Садыхов Рауф Хосровович
(21.10.1944–15.03.2015)

Sadykhov Rauf Khosrovovich
(21.10.1944–15.03.2015)



21 октября 2024 г. известному научному деятелю, ученому, профессору Рауфу Хосрововичу Садыхову исполнилось бы 80 лет.

Р. Х. Садыхов родился в Баку Азербайджанской ССР в семье служащих. Здесь в 1962 г. он окончил с серебряной медалью школу № 3 и в 1967 г. – Азербайджанский политехнический институт по специальности «Математические и счетно-решающие приборы и устройства». Получив диплом инженера-электрика, Р. Х. Садыхов был распределен в проектный институт «Азгоспромпроект», где до 1970 г. работал инженером-программистом, а затем старшим инженером-программистом.

С 1959 г. Р. Х. Садыхов активно занимался спортом и в течение двух лет входил в сборную Азербайджана по настольному теннису среди юношей. В 1963 г. стал призером Всесоюзного первенства Центрального совета Всесоюзного физкультурно-спортивного общества «Локомотив» среди мужчин, проходившего в Симферополе. В том же году участвовал в составе сборной Азербайджана по настольному теннису в Спартакиаде народов СССР. Будучи кандидатом в мастера спорта (1963 г.), после переезда в Минск неоднократно являлся чемпионом Академии наук БССР по настольному теннису.

Институт технической кибернетики АН БССР. В 1970 г. Рауф Хосровович поступил в аспирантуру Института кибернетики Академии наук Азербайджанской ССР. В 1971 г. был прикомандирован в аспирантуру Института технической кибернетики АН БССР и переехал в Минск. В 1975 г. здесь он защитил диссертацию на соискание ученой степени кандидата технических наук по специальности «Техническая кибернетика и теория информации» на тему «Анализ случайных процессов и динамики объектов на основе спектральных инвариантов» под руководством доктора технических наук, профессора П. М. Чеголина.

За период работы в Институте технической кибернетики АН БССР с 1972 по 1995 г. Р. Х. Садыхов прошел путь от младшего научного сотрудника до заведующего лабораторией идентификации систем. Он являлся одним из лидеров научных школ института: автоматизации научных исследований (руководитель школы П. М. Чеголин) и обработки изображений (руководитель школы С. В. Абламейко), которые оказали существенное влияние на развитие науки и техники в СССР, Беларуси и за рубежом. В 1991 г. в Институте технической кибернетики АН БССР защитил диссертацию на соискание ученой степени доктора технических наук по специальности «Применение вычислительной техники, математического моделирования и математических методов в научных исследованиях» на тему «Разработка методов и вычислительных структур цифровой обработки сигналов в системах автоматизации экспериментальных ис-

следований». В 1992 г. ему было присвоено ученое звание профессора по специальности «Применение вычислительной техники, математического моделирования и математических методов в научных исследованиях».

Белорусский государственный университет информатики и радиоэлектроники. С 1995 г. Р. Х. Садыхов работал в Белорусском государственном университете информатики и радиоэлектроники заведующим кафедрой ЭВМ факультета компьютерных систем и сетей и внес личный вклад в совершенствование организации научно-исследовательской и учебной работы в университете. Он возглавлял коллектив, разработавший учебный план, программы и общеобразовательный стандарт по специальности «Вычислительные машины, системы и сети».

На базе кафедры ЭВМ по инициативе профессора Р. Х. Садыхова создана научно-исследовательская лаборатория «Компьютерные системы идентификации образов» (НИЛ 3.6). Возглавляемый им коллектив НИЛ 3.6 в 2006 и 2007 гг. занимал первые места среди подразделений НИЧ БГУИР по научному направлению «Новые информационные технологии и системы управления». При его непосредственном участии на кафедре проведена модернизация и созданы совместные учебные лаборатории с ИТ-компаниями и предприятиями Республики Беларусь, организовано сотрудничество с различными научными центрами СНГ (Научным центром нейрокомпьютеров, Москва, совместный проект в 2000–2002 гг. в рамках программы Союзного государства Беларуси и России в области технологии обработки изображений земной поверхности на основе интеллектуальных нейронных сетей; Институтом информационных технологий, Тернополь, Украина), а также стран ближнего и дальнего зарубежья (Техническим университетом, Зелена Гура, Польша; Высшей технической школой, Равенсбург-Вайнгартен, Германия; Техническим университетом, Валенсия, Испания). Под руководством Рауфа Хосрововича на кафедре выполнен ряд проектов в рамках Республиканской программы научных исследований «Информатика», Фонда фундаментальных исследований Республики Беларусь, Министерства образования Республики Беларусь, а также постоянно велась активная научная работа с молодыми сотрудниками кафедры и студентами старших курсов.

Научная и общественная деятельность. Р. Х. Садыхов являлся известным специалистом в области создания компьютерных систем обработки и распознавания изображений, идентификации образов, защиты информации, моделирования параллельных архитектур вычислительных систем.

В 2002 г. им была создана научная школа «Цифровая обработка сигналов и изображений, распознавание образов, идентификация систем, параллельные архитектуры для цифровой обработки сигналов и изображений, системы искусственного интеллекта», которая объединяла ученых и молодых исследователей БГУИР и ОИПИ НАН Беларуси. В нее входило 45 человек (из них 6 докторов и 20 кандидатов технических наук). Им подготовлено шесть докторов технических наук: В. А. Зайка (2001 г., тема диссертации «Цифровая регистрация и экспресс-анализ многоканальной информации в системах автоматизации натурального эксперимента»); В. В. Соловьев (2001 г., «Методы и алгоритмы проектирования цифровых систем на основе программируемых логических интегральных схем»); В. А. Головки (2003 г., «Нейросетевые методы обучения и обработки информации в системах управления и прогнозирования»); В. В. Краснопрошин (2007 г., «Теоретические основы и технологии построения систем распознавания по прецедентности»); А. А. Дудкин (2010 г., «Методы обработки и анализа цифровых изображений топологических слоев интегральных микросхем»); М. М. Татур (2011 г., «Проблемно-ориентированные процессоры для поддержки параллельных вычислений в задачах классификации»). Также Р. Х. Садыховым подготовлено 20 кандидатов технических наук. Он многократно выступал оппонентом на защитах докторских и кандидатских диссертаций.

Рауф Хосровович разработал ряд аппаратно-программных комплексов стендовых испытаний сложных технических объектов (автомобилей, тракторов, ракет и их узлов) на пространственную вибрацию, которые были внедрены на предприятиях Беларуси, России и Молдовы; методы предварительной обработки изображений, включая сегментацию, скелетизацию, фильтрацию на основе технологии быстрых преобразований в различных ортогональных базисах, использующиеся еще в НПО «Интеграл»; цифровую систему распознавания рукописных символов и сигнатур (подписей), в которой реализованы методы выделения информативных при-

знаков на базе моментных функций, преобразования Хафа и усеченного преобразования Уолша – Адамара. В НИИ цифрового телевидения (Минск) была внедрена цифровая система автоматической настройки раstra телевизионных приемников, построенная на базе оригинальной модели и методов обучения на основе нейронных сетей с применением нелинейных функций активации.

Профессор Р. Х. Садыхов был членом Международной ассоциации распознавания образов (IAPR); вице-председателем Белорусского отделения международного общества нейронных сетей, вице-президентом Белорусской ассоциации по распознаванию и анализу изображений (БААРИ); членом английского общества по электротехнике и радиоэлектронике ИЕЕ; членом докторских советов по защите диссертаций (ОИПИ НАН Беларуси, БГУИР); членом экспертного совета Белорусского республиканского фонда фундаментальных исследований; председателем и заместителем председателя экспертного совета ВАК Беларуси; в 1999 г. председателем программного комитета 5-й Международной конференции «Распознавание образов и обработка информации»; с 2001 г. заместителем председателя и председателем программного комитета международных конференций «Нейронные сети и искусственный интеллект».

Научные достижения профессора Р. Х. Садыхова отмечены Государственной премией Республики Беларусь 2002 г. за научную работу «Распознавание и анализ стохастических данных и цифровых изображений». Он дважды получал стипендию Президента Республики Беларусь (в 2000 и 2011 гг.). За 2004 г. имел несколько почетных грамот: НАН Беларуси, ВАК Беларуси, Министерства образования Республики Беларусь.

В БГУИР Р. Х. Садыхов был награжден в 2005 г. Почетной грамотой за высокие достижения в научной и инновационной деятельности, активное участие в формировании и реализации государственных научных и научно-технических программ, в 2008 и 2009 гг. – Почетной грамотой за высокие показатели в труде, значительный вклад в развитие научной и инновационной деятельности. В 2007 г. получил Диплом Федерации космонавтики России за большой вклад в реализацию белорусско-российских космических программ. В 2009 г. был награжден нагрудным знаком «Отличник образования республики Беларусь» и Почетной грамотой Национального собрания Республики Беларусь за большой вклад в формирование и реализацию социальной политики Республики Беларусь и многолетнюю плодотворную научно-педагогическую деятельность.



После вручения Государственной премии Республики Беларусь, май 2003 г.

Профессором Р. Х. Садыховым было опубликовано около 500 научных работ, включая 4 монографии, 134 статьи в международных и отечественных изданиях, 80 авторских свидетельств, патент, учебное пособие, 10 препринтов, 244 доклада и тезиса на международных и республиканских конференциях, симпозиумах, совещаниях.

К сожалению, жизненный путь профессора Р. Х. Садыхова закончился 15 марта 2015 г. Друзья, коллеги и ученики помнят его высочайшую эрудицию, умение находить компромиссы, доброту, тактичность и восточную мудрость, а также веру в молодежь. Память о Р. Х. Садыкове, большом ученом, талантливом учителе и руководителе, сохраняется в его многочисленных трудах и сердцах его учеников и коллег.

*С. В. Абламейко, В. В. Голенков, В. А. Головки, А. В. Инютин,
М. М. Лукашевич, В. В. Краснопрошин, А. В. Тузиков*

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.

9. 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.

10. 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

ISSN 1816-0301 (Print)



9 771816 030000