

2025

Volume 32

No. 2

MODELING AND ANALYSIS OF INFORMATION SYSTEMS

SCIENTIFIC JOURNAL

Start date of publication — 1999

Published quarterly

FOUNDER

P.G. Demidov Yaroslavl State University

EDITORIAL OFFICE

14 Sovetskaya str., Yaroslavl 150003, Russian Federation

Website: <http://mais-journal.ru>

E-mail: mais@uniyar.ac.ru

Phone: +7 (4852) 79-77-73

2025

Том 32

№ 2

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

НАУЧНЫЙ ЖУРНАЛ

Издается с 1999 года

Выходит 4 раза в год

УЧРЕДИТЕЛЬ

федеральное государственное бюджетное образовательное учреждение высшего образования
«Ярославский государственный университет им. П. Г. Демидова»

РЕДАКЦИЯ

ул. Советская, 14, Ярославль, 150003, Российская Федерация

Website: <http://mais-journal.ru>

E-mail: mais@uniyar.ac.ru

Телефон: +7 (4852) 79-77-73

Свидетельство о регистрации СМИ ПИ №ФС77-66186 от 20.06.2016 выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Подписной индекс в каталоге «Урал-Пресс» — 31907.

Технический редактор, компьютерная вёрстка — К. В. Лагутина. Дата выхода в свет 30.06.2025. Формат 200×265 мм. Объем 130 с. Тираж 24 экз. Свободная цена. Издатель и его адрес: Ярославский государственный университет им. П. Г. Демидова; ул. Советская, 14, Ярославль, 150003, Россия. Типография и ее адрес: ИД «Канцлер»; Полушкина роща, д. 16, стр. 66-а, Ярославль, 150064, Россия.

Содержание предназначено для детей старше 12 лет.

Editor-in-Chief

Egor V. Kuzmin Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Deputy Editor-in-Chief

Vladimir A. Bashkin Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Editorial Board Secretary

Ilya V. Paramonov Ph.D., P.G. Demidov Yaroslavl State University (Russia)

The Editorial Board

- Sergei M. Abramov Professor, Doctor of Sciences, Corresponding Member of Russian Academy of Sciences, Program Systems Institute of RAS (Pereslavl-Zalesskiy, Russia)
- Lilian Aveneau Professor, XLIM Laboratory, University of Poitiers (Poitiers, France)
- Thomas Baar Professor, Doctor, Hochschule für Technik und Wirtschaft Berlin, University of Applied Sciences (Berlin, Germany)
- Olga L. Bandman Professor, Doctor of Sciences, Supercomputer Software Department, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)
- Vladimir N. Belykh Professor, Doctor of Sciences, Volga State Academy of Water Transport (Nizhny Novgorod, Russia)
- Vladimir A. Bondarenko Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Richard R. Brooks Professor, Clemson University (South Carolina, USA)
- Alex Dekhtyar Professor, California Polytechnic State University (Cal Poly, California, USA)
- Mikhail Dmitriev Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)
- Vladimir L. Dolnikov Doctor of Sciences, Moscow Institute of Physics and Technology (Moscow, Russia)
- Valery G. Durnev Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Sergey D. Glyzin Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Yuri G. Karpov Professor, Doctor of Sciences, St-Petersburg State Polytechnical University (Russia)
- Sergey A. Kashchenko Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Lev S. Kazarin Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Andrei Yu. Kolesov Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Olga Kouchnarenko Professor at the Burgundy-Franche-Comte University, The FEMTO-ST Institute (CNRS 6174) (Besancon, France)
- Nikolai A. Kudryashov Professor, Doctor of Sciences, MEPhI (Russia)
- Irina A. Lomazova Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)
- George G. Malinetskiy Professor, Doctor of Sciences, M.V. Keldysh Institute of Applied Mathematics RAS (Moscow, Russia)
- Victor E. Malyshkin Professor, Doctor of Sciences, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)
- Alexander V. Mikhailov Professor, Doctor of Sciences, University of Leeds, School of Mathematics (Leeds, Great Britain)
- Nikolai Kh. Rozov Professor, Doctor of Sciences, Lomonosov Moscow State University (Russia)
- Philippe Schnoebelen Senior Researcher, LSV, CNRS & ENS de Cachan (CACHAN, France)
- Natalia Sidorova Dr., Assistant Professor, Architecture of Information Systems group, Technische universiteit Eindhoven (Eindhoven, Netherlands)
- Ruslan L. Smeliansky Professor, Doctor of Sciences, Corresponding Member of RAS, Lomonosov Moscow State University (Russia)
- Valery A. Sokolov Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Javid Taheri Associate Professor, Ph.D., Karlstad University (Sweden)
- Eugeniy A. Timofeev Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
- Mark Trakhtenbrot Dr., Holon Institute of Technology (Holon, Israel)
- Dimitry Turaev Professor of Applied Mathematics & Mathematical Physics, Imperial College (London, Great Britain)
- Vladimir Zakharov Doctor of Sciences, Professor, Lomonosov Moscow State University (Russia)

Главный редактор

Е. В. Кузьмин д-р физ.-мат. наук, ЯрГУ (Россия)

Заместитель главного редактора

В. А. Башкин д-р физ.-мат. наук, ЯрГУ (Россия)

Ответственный секретарь

И. В. Парамонов канд. физ.-мат. наук, ЯрГУ (Россия)

Редакционная коллегия

С. М. Абрамов д-р физ.-мат. наук, чл.-корр. РАН, Институт программных систем РАН
им. А.К. Айламазяна (Россия)

L. Aveneau проф., Университет Пуатье (Франция)

T. Baar д-р наук, проф., Университет прикладных технических и экономических наук
Берлина (Германия)

О. Л. Бандман д-р техн. наук, Институт вычислительной математики и математической
геофизики СО РАН (Россия)

В. Н. Белых д-р физ.-мат. наук, проф., Волжская государственная академия водного транспорта
(Россия)

В. А. Бондаренко д-р физ.-мат. наук, проф., ЯрГУ (Россия)

R. Brooks проф., Университет Клемсона (США)

С. Д. Глызин д-р физ.-мат. наук, проф., ЯрГУ (Россия)

A. Dekhtyar проф., Калифорнийский политехнический университет, департамент
компьютерных наук (США)

М. Г. Дмитриев д-р физ.-мат. наук, проф., ВШЭ (Россия)

В. Л. Дольников д-р физ.-мат. наук, проф., МФТИ (Россия)

В. Г. Дурнев д-р физ.-мат. наук, проф., ЯрГУ (Россия)

В. А. Захаров д-р физ.-мат. наук, проф., МГУ (Россия)

Л. С. Казарин д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Ю. Г. Карпов д-р техн. наук, проф., Санкт-Петербургский государственный технический
университет (Россия)

С. А. Кащенко д-р физ.-мат. наук, проф., ЯрГУ (Россия)

А. Ю. Колесов д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Н. А. Кудряшов д-р физ.-мат. наук, проф., Засл. деятель науки РФ, МИФИ (Россия)

O. Kouchnarenko проф., Университет Бургундии - Франш-Комтэ (Франция)

И. А. Ломазова д-р физ.-мат. наук, проф., ВШЭ (Россия)

Г. Г. Малинецкий д-р физ.-мат. наук, проф., Институт прикладной математики им. М.В. Келдыша
РАН (Россия)

В. Э. Малышкин д-р техн. наук, проф., Институт вычислительной математики и математической
геофизики СО РАН (Россия)

A. Mikhailov д-р физ.-мат. наук, проф., Университет Лидса (Великобритания)

Н. Х. Розов д-р физ.-мат. наук, проф., чл.-корр. РАО, МГУ (Россия)

N. Sidorova д-р наук, университет Эйндховена (Нидерланды)

Р. Л. Смелянский д-р физ.-мат. наук, проф., член-корр. РАН, академик РАЕН, МГУ (Россия)

В. А. Соколов д-р физ.-мат. наук, проф., ЯрГУ (Россия)

J. Taheri доцент, Университет Карлстада (Швеция)

Е. А. Тимофеев д-р физ.-мат. наук, проф., ЯрГУ (Россия)

M. Trakhtenbrot д-р комп. наук, Холонский технологический институт (Израиль)

D. Turaev проф., Имперский колледж Лондона (Великобритания)

Ph. Schnoebelen проф., Национальный центр научных исследований и Высшая нормальная школа
Кашана (Франция)

Contents

Discrete Mathematics in Relation to Computer Science

Barotov D. N., Barotov R. N. On Extremal Elements and the Cardinality of the Set of Continuously Differentiable Convex Extensions of a Boolean Function.....100

Brazhnikov E. A., Panin A. A., Ratushnyi A. V. A Patterning Algorithm for the Dynamic Bin Packing Problem with Placement Groups110

Smirnov A. V. Exact Algorithm for the Problem of the Minimum Complete Spanning Tree of a Divisible Multiple Graph.....132

Theory of Computing

Kryukov O. S., Voloshko A. G., Ivutin A. N. Modeling of Parallel Program Synchronization Primitives.....150

Computing Methodologies and Applications

Bystrov L. Y., Gladkov A. N., Kuzmin E. V. Detection of Square Wave Impulse Interference in Eddy Current Rail Defectograms.....172

Glyzin S. D., Kashchenko S. A., Kosterin D. S. Piecewise Constant Modes of Operation of Fully Coupled Networks and Their Limit Integro-differential Systems.....206

Содержание

Discrete Mathematics in Relation to Computer Science

Баротов Д. Н., Баротов Р. Н. Об экстремальных элементах и мощности множества непрерывно дифференцируемых выпуклых продолжений булевой функции100

Бражников Е. А., Панин А. А., Ратушный А. В. Алгоритм шаблонизации для динамической задачи упаковки в контейнеры с группами размещения110

Смирнов А. В. Точный алгоритм для задачи о минимальном полном остовном дереве в делимом кратном графе.....132

Theory of Computing

Крюков О. С., Волошко А. Г., Ивутин А. Н. Моделирование примитивов синхронизации параллельных программ.....150

Computing Methodologies and Applications

Быстров Л. Ю., Гладков А. Н., Кузьмин Е. В. Обнаружение прямоугольных импульсных помех на вихретоковых дефектограммах рельсов172

Глызин С. Д., Кащенко С. А., Костерин Д. С. Кусочно-постоянные режимы работы полносвязных сетей и их предельных интегро-дифференциальных систем.....206

On Extremal Elements and the Cardinality of the Set of Continuously Differentiable Convex Extensions of a Boolean Function

D. N. Barotov¹, R. N. Barotov²

DOI: [10.18255/1818-1015-2025-2-100-109](https://doi.org/10.18255/1818-1015-2025-2-100-109)

¹Financial University under the Government of the Russian Federation, Moscow, Russia

²Khujand State University named after academician Bobojon Gafurov, Khujand, Tajikistan

MSC2020: 06E30, 54C20, 03E17

Research article

Full text in Russian

Received February 19, 2025

Revised April 1, 2025

Accepted April 9, 2025

In this paper we study the existence of the maximal and minimal elements of the set of continuously differentiable convex extensions to $[0, 1]^n$ of an arbitrary Boolean function $f_B(x_1, x_2, \dots, x_n)$ and the cardinality of the set of continuously differentiable convex extensions to $[0, 1]^n$ of the Boolean function $f_B(x_1, x_2, \dots, x_n)$. As a result of the study, it was established that the cardinality of the set of continuously differentiable convex extensions to $[0, 1]^n$ of an arbitrary Boolean function $f_B(x_1, x_2, \dots, x_n)$ is equal to the continuum. It is argued that for any Boolean function $f_B(x_1, x_2, \dots, x_n)$, there is no minimal element among its continuously differentiable convex extensions to $[0, 1]^n$. It is proved that for any Boolean function $f_B(x_1, x_2, \dots, x_n)$, the set of its continuously differentiable convex extensions to $[0, 1]^n$ has a maximal element only if the number of essential variables of the given Boolean function $f_B(x_1, x_2, \dots, x_n)$ is less than 2.

Keywords: continuously differentiable convex extension of a Boolean function; extremal elements of a set; cardinality of a set

INFORMATION ABOUT THE AUTHORS

Barotov, Dostonjon N. (corresponding author)	ORCID iD: 0000-0001-5047-7710 . E-mail: DNBarotov@fa.ru Senior Lecturer, Department of Mathematics and Data Analysis
---	--

Barotov, Ruziboy N.	ORCID iD: 0000-0003-3729-6143 . E-mail: DNBarotov@fa.ru Lecturer, Department Mathematical Analysis named after Professor A. Mukhsinov
---------------------	---

For citation: D. N. Barotov and R. N. Barotov, "On extremal elements and the cardinality of the set of continuously differentiable convex extensions of a Boolean function", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 100–109, 2025. DOI: [10.18255/1818-1015-2025-2-100-109](https://doi.org/10.18255/1818-1015-2025-2-100-109).

Об экстремальных элементах и мощности множества непрерывно дифференцируемых выпуклых продолжений булевой функции

Д. Н. Баротов¹, Р. Н. Баротов²DOI: [10.18255/1818-1015-2025-2-100-109](https://doi.org/10.18255/1818-1015-2025-2-100-109)¹Финансовый университет при Правительстве Российской Федерации, Москва, Россия²Худжандский государственный университет имени академика Б. Гафурова, Худжанд, Таджикистан

УДК 519.716.322+519.85+517.518.244

Научная статья

Полный текст на русском языке

Получена 19 февраля 2025 г.

После доработки 1 апреля 2025 г.

Принята к публикации 9 апреля 2025 г.

В данной статье изучается существование максимального и минимального элементов множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ и мощность множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$. В результате исследования установлено, что мощность множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ равна континууму. Аргументировано, что для любой булевой функции $f_B(x_1, x_2, \dots, x_n)$ среди её непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ нет минимального элемента. Доказано, что для любой булевой функции $f_B(x_1, x_2, \dots, x_n)$ множество её непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ имеет максимальный элемент только тогда, когда количество существенных переменных булевой функции $f_B(x_1, x_2, \dots, x_n)$ меньше 2.

Ключевые слова: непрерывно дифференцируемое выпуклое продолжение булевой функции; экстремальные элементы множества; мощность множества

ИНФОРМАЦИЯ ОБ АВТОРАХ

Баротов, Достонжон Нумонжонович (автор для корреспонденции)	ORCID iD: 0000-0001-5047-7710 . E-mail: DNBarotov@fa.ru Старший преподаватель кафедры математики и анализа данных
Баротов, Рузибой Нумонжонович	ORCID iD: 0000-0003-3729-6143 . E-mail: DNBarotov@fa.ru Преподаватель кафедры математического анализа имени профессора А. Мухсинова

Для цитирования: D. N. Barotov and R. N. Barotov, "On extremal elements and the cardinality of the set of continuously differentiable convex extensions of a Boolean function", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 100–109, 2025. DOI: [10.18255/1818-1015-2025-2-100-109](https://doi.org/10.18255/1818-1015-2025-2-100-109).

Введение

Многие труднорешаемые дискретные задачи, возникающие в разных областях, включая комбинаторику, современную кибернетику, криптографию, а также информатику, могут быть сведены к системам булевых уравнений [1–3]. Поэтому системам булевых уравнений посвящено значительное количество работ, разработано несколько направлений исследований и алгоритмов их решения [2, 4, 5], но, несмотря на это, в связи с тем, что задача решения системы булевых уравнений в общем случае остается труднорешаемой, т. е. NP-трудной, в научном сообществе и в настоящее время продолжает расти интерес к поиску новых алгоритмов их решения в различных направлениях как в классических, так и в квантовых моделях вычислений [6, 7]. Одним из таких направлений для решения систем булевых уравнений является трансформация (преобразование) заданной системы булевых уравнений путем представления некоторого вещественного продолжения (аналога) для каждой булевой функции в систему вещественных уравнений над полем действительных чисел, поскольку, во-первых, в этой области известно много методов и алгоритмов решения систем, а во-вторых, его можно использовать и при решении смешанных систем, заданных одновременно математическими и логическими операциями [8, 9]. В свою очередь, преобразованная система вещественных уравнений может быть сведена в задачу непрерывной оптимизации, так как принципиальное отличие данного подхода от «переборных» алгоритмов локального поиска состоит в том, что на каждой итерации алгоритма сдвиг по градиенту (антиградиенту) производится по всем переменным одновременно [10]. В данном направлении относительно недавно в [9, 11–15] получены некоторые важные результаты, а именно, в [9] рассмотрено конструирование полилинейного продолжения булевой функции и аргументировано, что задача решения произвольной системы булевых уравнений с n переменными может быть сведена к задаче непрерывной минимизации на $[0, 1]^n$ целевой функции, не имеющей строгих локальных минимумов внутри любой k -мерной грани куба $[0, 1]^n$, $k \in \{1, 2, \dots, n\}$, а в [11–14] построены выпуклые (вогнутые) продолжения булевых функций n переменных на $[0, 1]^n$ и на основе построенных выпуклых (вогнутых) продолжений булевых функций на $[0, 1]^n$, в частности, конструктивно доказано, что задача решения системы булевых уравнений может быть сведена к задаче минимизации (максимизации) целевой функции, любой локальный минимум (максимум) которой в искомой области является глобальным минимумом (максимумом), а также конструктивно доказано, что для любой булевой функции n переменных существует единственная вещественная функция, являющаяся максимумом (минимумом) среди всех ее выпуклых (вогнутых) продолжений на $[0, 1]^n$ и в дополнение ко всему, в [15] проведено сравнительное исследование между выпуклыми, полилинейными и вогнутыми продолжениями булевых функций. Поэтому также важным является изучение и доказательство свойств вещественных продолжений булевых функций, представляющих интерес при преобразовании систем булевых уравнений к задаче непрерывной оптимизации.

Данная статья является продолжением статей [11–13], а именно, в ней изучается существование максимального и минимального элементов множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ и мощность множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$. В результате исследования, во-первых, аргументируется, что множество непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ с точки зрения существования максимума заметно отличается от множества всех выпуклых продолжений на $[0, 1]^n$ этой булевой функции $f_B(x_1, x_2, \dots, x_n)$, а именно, доказываемая, что множество непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ данной булевой функции $f_B(x_1, x_2, \dots, x_n)$ имеет максимальный элемент лишь тогда, когда количество вещественных переменных булевой функции $f_B(x_1, x_2, \dots, x_n)$ меньше 2, а во-вторых, устанавливается,

что мощность множества непрерывно дифференцируемых выпуклых продолжений на $[0, 1]^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$ равна континууму.

1. Используемые определения и обозначения

Пусть $\mathbb{B}^n = \{0, 1\}^n$ — множество всевозможных двоичных слов (булевых векторов) длины n , $\mathbb{K}^n = [0, 1]^n$ — n -мерный куб, натянутый на булевы векторы длины n .

Пусть $\text{int}(\mathbb{K}^n) = (0, 1)^n$ — множество внутренних точек куба $\mathbb{K}^n$.

Определение 1. Отображение вида $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$ называется булевой функцией.

Определение 2. Переменная x_k , где $k \in \{1, 2, \dots, n\}$, булевой функции $f_B(x_1, x_2, \dots, x_n)$ называется существенной (булева функция $f_B(x_1, x_2, \dots, x_n)$ существенно зависит от x_k), если имеет место

$$f_B(x_1, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_n) \neq f_B(x_1, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n).$$

Определение 3. Отображение вида $f : \mathbb{K}^n \rightarrow \mathbb{R}$ называется выпуклой функцией на $\mathbb{K}^n$, если для любых $x, y \in \mathbb{K}^n$ и любого $\alpha \in [0, 1]$ выполняется

$$f(\alpha \cdot x + (1 - \alpha) \cdot y) \leq \alpha \cdot f(x) + (1 - \alpha) \cdot f(y).$$

Определение 4. Отображение вида $f_C : \mathbb{K}^n \rightarrow \mathbb{R}$ называется выпуклым продолжением на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$, если выполняются следующие два условия:

- а) отображение f_C на $\mathbb{K}^n$ является выпуклой функцией;
- б) имеет место равенство $f_C(b_1, b_2, \dots, b_n) = f_B(b_1, b_2, \dots, b_n) \quad \forall (b_1, b_2, \dots, b_n) \in \mathbb{B}^n$.

Определение 5. Отображение вида $f_{DM} : \mathbb{K}^n \rightarrow \mathbb{R}$ называется максимумом среди всех выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$, если выполняются следующие два условия:

- а) отображение f_{DM} является выпуклым продолжением булевой функции f_B на $\mathbb{K}^n$;
- б) для любого f_C — выпуклого продолжения на $\mathbb{K}^n$ булевой функции f_B и любой $(x_1, x_2, \dots, x_n) \in \mathbb{K}^n$ справедливо неравенство $f_C(x_1, x_2, \dots, x_n) \leq f_{DM}(x_1, x_2, \dots, x_n)$.

Определение 6. Отображение вида $f_R : \mathbb{K}^n \rightarrow \mathbb{R}$ назовём непрерывно дифференцируемым выпуклым продолжением на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$, если выполняются следующие два условия:

- а) отображение f_R на $\mathbb{K}^n$ является непрерывно дифференцируемой выпуклой функцией;
- б) имеет место равенство $f_R(b_1, b_2, \dots, b_n) = f_B(b_1, b_2, \dots, b_n) \quad \forall (b_1, b_2, \dots, b_n) \in \mathbb{B}^n$.

Определение 7. Отображение вида $f_{NR} : \mathbb{K}^n \rightarrow \mathbb{R}$ назовём максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$, если выполняются следующие два условия:

- а) отображение f_{NR} является непрерывно дифференцируемым выпуклым продолжением на $\mathbb{K}^n$ булевой функции f_B ;
- б) для любого f_R — непрерывно дифференцируемого выпуклого продолжения на $\mathbb{K}^n$ булевой функции f_B и любой $(x_1, x_2, \dots, x_n) \in \mathbb{K}^n$ справедливо неравенство

$$f_R(x_1, x_2, \dots, x_n) \leq f_{NR}(x_1, x_2, \dots, x_n).$$

Пусть $E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n)$ — множество непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$.

2. О мощности и минимальном элементе множества непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции

$$f_B : \mathbb{B}^n \rightarrow \mathbb{B}$$

В этом разделе обоснуем, что для любой булевой функции $f_B(x_1, x_2, \dots, x_n)$ справедливо равенство $\text{card}(E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n)) = c$ и также покажем, что среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$ нет минимума.

Утверждение 1. Для каждой булевой функции $f_B(x_1, x_2, \dots, x_n)$ множество её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ имеет мощность континуума и не имеет минимального элемента.

Доказательство. Очевидно, что имеет место вложение

$$E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n) \subset C(\mathbb{K}^n), \quad (1)$$

т. е. множество $E(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n)$ является подмножеством множества всех непрерывных функций, заданных на $\mathbb{K}^n$. В силу теоремы 2.1, приведённой в [11], имеем, что $E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n) \neq \emptyset$. Пусть $\psi_R(x_1, x_2, \dots, x_n)$ — произвольный элемент множества $E(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n)$. Тогда рассмотрим следующую функцию

$$\psi_\alpha(x_1, x_2, \dots, x_n) = \psi_R(x_1, x_2, \dots, x_n) + \alpha \cdot \sum_{k=1}^n (x_k^2 - x_k). \quad (2)$$

Обоснуем, что имеет место включение

$$\psi_\alpha(x_1, x_2, \dots, x_n) \in E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n) \quad \forall \alpha \in (0, +\infty). \quad (3)$$

Для этого покажем справедливость следующих двух свойств:

- Функция $\psi_\alpha(x_1, x_2, \dots, x_n)$ на $\mathbb{K}^n$ является непрерывно дифференцируемой и выпуклой.
- Имеет место равенство $\psi_\alpha(a_1, a_2, \dots, a_n) = f_B(a_1, a_2, \dots, a_n) \quad \forall (a_1, a_2, \dots, a_n) \in \mathbb{B}^n$.

Обоснование этих двух свойств:

- Непрерывная дифференцируемость. Из непрерывной дифференцируемости функций $\psi_R(x_1, x_2, \dots, x_n)$ и $\sum_{k=1}^n (x_k^2 - x_k)$, т. е. включений $\psi_R(x_1, x_2, \dots, x_n) \in C^1(\mathbb{K}^n)$ и $\sum_{k=1}^n (x_k^2 - x_k) \in C^1(\mathbb{K}^n)$ получаем, что функция $\psi_\alpha(x_1, x_2, \dots, x_n)$ является непрерывно дифференцируемой, т. е. $\psi_\alpha(x_1, x_2, \dots, x_n) \in C^1(\mathbb{K}^n)$.

Выпуклость. Ввиду выпуклости функции $\sum_{k=1}^n (x_k^2 - x_k)$ на $\mathbb{K}^n$ и $\alpha > 0$, имеем, что функция $\psi_\alpha(x_1, x_2, \dots, x_n)$ на $\mathbb{K}^n$ как сумма двух выпуклых функций является выпуклой.

- Пусть $(a_1, a_2, \dots, a_n) \in \mathbb{B}^n$. Тогда

$$\begin{aligned} \psi_\alpha(a_1, a_2, \dots, a_n) &= \psi_R(a_1, a_2, \dots, a_n) + \alpha \cdot \sum_{k=1}^n (a_k^2 - a_k) = \psi_R(a_1, a_2, \dots, a_n) + \\ &+ \alpha \cdot \sum_{k=1}^n (a_k - a_k) = \psi_R(a_1, a_2, \dots, a_n) + 0 = f_B(a_1, a_2, \dots, a_n). \end{aligned}$$

Ввиду (2) получаем, что $\forall \alpha_1, \alpha_2 \in (0, +\infty), \alpha_1 < \alpha_2$ и $\forall (x_1^*, x_2^*, \dots, x_n^*) \in \text{int}(\mathbb{K}^n)$ справедлива цепочка

$$\psi_R(x_1^*, x_2^*, \dots, x_n^*) > \psi_{\alpha_1}(x_1^*, x_2^*, \dots, x_n^*) > \psi_{\alpha_2}(x_1^*, x_2^*, \dots, x_n^*). \quad (4)$$

Ввиду (3) получаем, что имеет место включение

$$\bigcup_{\alpha \in [1,2]} \{\psi_\alpha(x_1, x_2, \dots, x_n)\} \subset E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n). \quad (5)$$

Отсюда, в силу $\text{card}\left(\bigcup_{\alpha \in [1,2]} \{\psi_\alpha(x_1, x_2, \dots, x_n)\}\right) = c$, вытекающего из (4), и $\text{card}(C(\mathbb{K}^n)) = c$, и также вложений (1), (5), получаем $\text{card}(E_C(f_B(x_1, x_2, \dots, x_n), \mathbb{K}^n)) = c$, т. е. мощность множества непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$ равна континууму. Ввиду произвольности $\psi_R(x_1, x_2, \dots, x_n)$ и справедливости (3) и (4) получаем, что среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$ нет минимума. $\square$

Замечание 1. Легко заметить, что утверждение 1 является уточнением (усилением) теоремы 2.1, приведённой в [11].

3. О максимальном элементе множества непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \rightarrow \mathbb{B}$

В этом разделе докажем, что множество непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ имеет максимальный элемент лишь тогда, когда количество существенных переменных этой булевой функции $f_B(x_1, x_2, \dots, x_n)$ меньше 2. Для этого сначала докажем следующую лемму.

Лемма 1. Для каждой булевой функции $f_B(x, y)$, которая существенно зависит от двух своих переменных x и y , среди её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ нет максимума.

Доказательство. Доказательство проводим от противного: пусть некоторая вещественная функция $f_{NR}(x, y)$ является максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ булевой функции $f_B(x, y)$. В силу следствия 3, приведённого в [13], получим, что $\forall (x, y) \in \mathbb{K}^2$ выполнено неравенство

$$\begin{aligned} f_{NR}(x, y) &\leq f_{DM}(x, y) = (1 - x - y) \cdot f_B(0, 0) + x \cdot f_B(1, 0) + y \cdot f_B(0, 1) + \\ &+ \frac{f_B(0, 0) - f_B(0, 1) - f_B(1, 0) + f_B(1, 1)}{4} \cdot (2x + 2y - 1 - |x - y| + |x + y - 1|) + \\ &+ \frac{|f_B(0, 0) - f_B(0, 1) - f_B(1, 0) + f_B(1, 1)|}{4} \cdot (|x - y| + |x + y - 1| - 1), \end{aligned} \quad (6)$$

так как $f_{DM}(x, y)$ является максимумом среди всех выпуклых продолжений на $\mathbb{K}^2$ булевой функции $f_B(x, y)$. Нетрудно показать, что для каждой булевой функции $f_B(x, y)$, существенно зависящей от двух своих переменных x и y выполнено неравенство

$$f_B(0, 0) - f_B(0, 1) - f_B(1, 0) + f_B(1, 1) \neq 0. \quad (7)$$

Ввиду (6) и (7) функция $f_{DM}(x, y)$ на $\mathbb{K}^2$ не является дифференцируемой и, следовательно, существует конкретная точка $(x^*, y^*) \in \mathbb{K}^2$ такая, что выполняется следующее строгое неравенство

$$f_{NR}(x^*, y^*) < f_{DM}(x^*, y^*), \quad (8)$$

так как в противном случае $f_{NR}(x, y) \equiv f_{DM}(x, y)$, а это противоречит к тому, что функция $f_{NR}(x, y)$ на $\mathbb{K}^2$ непрерывно дифференцируема. Теперь, ввиду неравенства (8), эквивалентного неравенству $f_{NR}(x^*, y^*) < f_1(x^*, y^*)$, нетрудно показать, например, рассмотрев два случая относительно знака левой части (7) и заметив, что функция $f_\beta(x^*, y^*)$ по β непрерывна на $[1, +\infty)$ и также не возрастает на $[1, +\infty)$, что существует $\beta^* \in (1, +\infty)$ такое, что выполняется следующее строгое неравенство

$$f_{NR}(x^*, y^*) < f_{\beta^*}(x^*, y^*), \quad (9)$$

где

$$\begin{aligned} f_\beta(x, y) = & (1 - x - y) \cdot f_B(0, 0) + x \cdot f_B(1, 0) + y \cdot f_B(0, 1) + \\ & + \frac{f_B(0, 0) - f_B(0, 1) - f_B(1, 0) + f_B(1, 1)}{4} \cdot (2x + 2y - 1 - |x - y|^\beta + |x + y - 1|^\beta) + \\ & + \frac{|f_B(0, 0) - f_B(0, 1) - f_B(1, 0) + f_B(1, 1)|}{4} \cdot (|x - y|^\beta + |x + y - 1|^\beta - 1), \quad \beta \geq 1. \end{aligned} \quad (10)$$

Ввиду справедливости равенства

$$|z|^\beta = |z| \quad \forall z \in \{-1, 0, 1\} \text{ и } \forall \beta > 1$$

и того, что для любого $\beta > 1$ функция, равная $|x|^\beta$, является выпуклой и непрерывно дифференцируемой, имеем, что функция $f_\beta(x, y)$ для каждого $\beta > 1$, в частности для $\beta = \beta^*$, является непрерывно дифференцируемым выпуклым продолжением на $\mathbb{K}^2$ булевой функции $f_B(x, y)$. Таким образом, получаем противоречие, т. е. полученное в (9) противоречит сделанному выше предположению о том, что вещественная функция $f_{NR}(x, y)$ является максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ булевой функции $f_B(x, y)$, существенно зависящей от x и y . $\square$

Теперь, основываясь на лемме 1, докажем следующую теорему, являющуюся усилением доказанной леммы 1.

Теорема 1. Если количество существенных переменных булевой функции $f_B(x_1, x_2, \dots, x_n)$ не меньше 2, то среди её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ нет максимума, а если оно меньше 2, то среди её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ есть максимум.

Доказательство. Рассмотрим два случая.

Случай 1. Пусть количество существенных переменных булевой функции $f_B(x_1, x_2, \dots, x_n)$ меньше 2. В этом случае достаточно рассмотреть булеву функцию $f_B(x)$, зависящую только от одной, не обязательно существенной, переменной x . Согласно следствию 2, приведённому в [13], имеем, что для булевой функции $f_B(x)$ вещественная функция

$$f_{DM}(x) = (1 - x) \cdot f_B(0) + x \cdot f_B(1) \quad (11)$$

является единственным максимумом среди всех её выпуклых продолжений на $\mathbb{K}$. Так как функция $f_{DM}(x)$, определённая формулой (11), непрерывно дифференцируема, то она также является максимумом среди непрерывно дифференцируемых выпуклых продолжений булевой функции $f_B(x)$, т. е.

$$f_{NR}(x) = (1 - x) \cdot f_B(0) + x \cdot f_B(1). \quad (12)$$

Случай 2. Пусть количество существенных переменных булевой функции $f_B(x_1, x_2, \dots, x_n)$ не меньше 2. В этом случае, без потери общности можно считать, что все переменные $x_1, x_2, \dots, x_n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$ являются существенными. Докажем от противного: пусть существует вещественная функция $f_{NR}(x_1, x_2, \dots, x_n)$, которая является максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ булевой функции $f_B(x_1, x_2, \dots, x_n)$. Тогда имеем, что $\forall (b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_{j-1}, b_{j+1}, \dots, b_n) \in \mathbb{B}^{n-2}$ суженная вещественная функция

$$f_{NR}(b_1, \dots, b_{i-1}, x_i, b_{i+1}, \dots, b_{j-1}, x_j, b_{j+1}, \dots, b_n)$$

является максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ суженной булевой функции $f_B(b_1, \dots, b_{i-1}, x_i, b_{i+1}, \dots, b_{j-1}, x_j, b_{j+1}, \dots, b_n)$. Согласно доказанному в [16, 17] имеем, что существует выделяемая пара переменных (x_i, x_j) булевой функции $f_B(x_1, x_2, \dots, x_n)$, т. е. существуют $i, j \in \{1, 2, \dots, n\}, i < j$ и $(b_1^*, \dots, b_{i-1}^*, b_{i+1}^*, \dots, b_{j-1}^*, b_{j+1}^*, \dots, b_n^*) \in \mathbb{B}^{n-2}$ такие, что переменные x_i и x_j суженной булевой функции $f_B(b_1^*, \dots, b_{i-1}^*, x_i, b_{i+1}^*, \dots, b_{j-1}^*, x_j, b_{j+1}^*, \dots, b_n^*)$ являются существенными. Отсюда получаем, что вещественная функция вида

$$g_{NR}(x_i, x_j) = f_{NR}(b_1^*, \dots, b_{i-1}^*, x_i, b_{i+1}^*, \dots, b_{j-1}^*, x_j, b_{j+1}^*, \dots, b_n^*)$$

является максимумом среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ булевой функции

$$g_B(x_i, x_j) = f_B(b_1^*, \dots, b_{i-1}^*, x_i, b_{i+1}^*, \dots, b_{j-1}^*, x_j, b_{j+1}^*, \dots, b_n^*),$$

существенно зависящей от своих двух переменных x_i и x_j . Пришли к противоречию с тем, что, согласно доказанной лемме 1, среди непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^2$ булевой функции $g_B(x_i, x_j)$, существенно зависящей от своих двух переменных x_i и x_j , нет максимума. $\square$

Заключение

В результате исследования установлено, что мощность множества непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ произвольной булевой функции $f_B(x_1, x_2, \dots, x_n)$ равна континууму. Аргументировано, что для любой булевой функции $f_B(x_1, x_2, \dots, x_n)$ среди её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ нет минимального элемента. Доказано, что для любой булевой функции $f_B(x_1, x_2, \dots, x_n)$ множество её непрерывно дифференцируемых выпуклых продолжений на $\mathbb{K}^n$ имеет максимальный элемент тогда и только тогда, когда количество существенных переменных этой булевой функции $f_B(x_1, x_2, \dots, x_n)$ меньше 2.

Отметим, что полученный в данной работе результат является усилением (обобщением) результата, приведенного в [11], и может быть использован при проведении аналогичного исследования, соответствующего выпуклым [18] и вогнутым [19] продолжениям булевоподобных дискретных функций. Кроме того, наряду с [11], он может быть применен при сведении смешанной системы булевых и булевоподобных дискретных уравнений к задаче непрерывной оптимизации, а также при последующем поиске множества решений.

Благодарности

Авторы искренне благодарят рецензента за внимательное прочтение работы и полезные замечания и рекомендации.

References

- [1] A. H. Abdel-Gawad, A. F. Atiya, and N. M. Darwish, “Solution of systems of Boolean equations via the integer domain”, *Information Sciences*, vol. 180, no. 2, pp. 288–300, 2010. DOI: [10.1016/j.ins.2009.09.010](https://doi.org/10.1016/j.ins.2009.09.010).
- [2] G. V. Bard, *Algorithms for solving linear and polynomial systems of equations over finite fields, with applications to cryptanalysis*. University of Maryland, College Park, 2007.
- [3] V. Leontiev and E. Gordeev, “On the number of solutions to a system of Boolean equations”, *Automation and Remote Control*, vol. 82, pp. 1581–1596, 2021. DOI: [10.1134/S000511792109006X](https://doi.org/10.1134/S000511792109006X).
- [4] E. Ishchukova, E. Maro, and P. Pristalov, “Algebraic analysis of a simplified encryption algorithm GOST R 34.12-2015”, *Computation*, vol. 8, no. 2, p. 51, 2020. DOI: [10.3390/computation8020051](https://doi.org/10.3390/computation8020051).
- [5] A. I. Pakhomchik, V. V. Voloshinov, V. M. Vinokur, and G. B. Lesovik, “Converting of Boolean expression to linear equations, inequalities and QUBO penalties for cryptanalysis”, *Algorithms*, vol. 15, no. 2, p. 33, 2022. DOI: [10.3390/a15020033](https://doi.org/10.3390/a15020033).
- [6] S. Ramos-Calderer *et al.*, “Solving systems of Boolean multivariate equations with quantum annealing”, *Physical Review Research*, vol. 4, no. 1, p. 013 096, 2022. DOI: [10.1103/PhysRevResearch.4.013096](https://doi.org/10.1103/PhysRevResearch.4.013096).
- [7] E. Burek, M. Wroński, K. Mańk, and M. Misztal, “Algebraic attacks on block ciphers using quantum annealing”, *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 678–689, 2022. DOI: [10.1109/TETC.2022.3143152](https://doi.org/10.1109/TETC.2022.3143152).
- [8] J. Gu, “Global optimization for satisfiability (SAT) problem”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 3, pp. 361–381, 1994. DOI: [10.1109/69.334864](https://doi.org/10.1109/69.334864).
- [9] D. N. Barotov and R. N. Barotov, “Polylinear continuations of some discrete functions and an algorithm for finding them”, *Numerical methods and programming*, vol. 24, no. 1, pp. 10–23, 2023, in Russian. DOI: [10.26089/NumMet.v24r102](https://doi.org/10.26089/NumMet.v24r102).
- [10] R. T. Faizullin, V. I. Dulkeyt, and Y. Y. Ogorodnikov, “Hybrid method for the approximate solution of the 3-satisfiability problem associated with the factorization problem”, *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, vol. 19, no. 2, pp. 285–294, 2013, in Russian.
- [11] D. N. Barotov and R. N. Barotov, “Construction of smooth convex extensions of Boolean functions”, *Russian Universities Reports. Mathematics*, vol. 29, no. 145, pp. 20–28, 2024, in Russian. DOI: [10.20310/2686-9667-2024-29-145-20-28](https://doi.org/10.20310/2686-9667-2024-29-145-20-28).
- [12] D. N. Barotov, “Convex continuation of a Boolean function and its applications”, *Journal of Applied and Industrial Mathematics*, vol. 18, no. 1, pp. 1–9, 2024. DOI: [10.1134/S1990478924010010](https://doi.org/10.1134/S1990478924010010).
- [13] D. N. Barotov, “On the existence and properties of convex extensions of Boolean functions”, *Mathematical Notes*, vol. 115, no. 3, pp. 489–505, 2024. DOI: [10.1134/S0001434624030210](https://doi.org/10.1134/S0001434624030210).
- [14] D. N. Barotov, “Concave continuations of Boolean functions and some of their properties and applications”, *Bulletin of Irkutsk State University. Series Mathematics*, vol. 49, pp. 105–123, 2024, in Russian. DOI: [10.26516/1997-7670.2024.49.105](https://doi.org/10.26516/1997-7670.2024.49.105).
- [15] D. N. Barotov and V. A. Sudakov, “On inequalities between convex, concave, and multilinear continuations of Boolean functions”, *Keldysh Institute preprints*, no. 30, pp. 1–13, 2024, in Russian. DOI: [10.20948/prepr-2024-30](https://doi.org/10.20948/prepr-2024-30).

- [16] A. Salomaa, “On essential variables of functions, especially in the algebra of logic”, *Annales Fennici Mathematici*, no. 339, pp. 1–11, 1964. DOI: [10.5186/aasfm.1964.339](https://doi.org/10.5186/aasfm.1964.339).
- [17] Y. Y. Breitbart, “Essential variables of functions of the algebra of logic”, *Doklady Akademii Nauk*, vol. 172, no. 1, pp. 9–10, 1967, in Russian.
- [18] D. N. Barotov, “Convex continuations of some discrete functions”, *Journal of Applied and Industrial Mathematics*, vol. 18, no. 3, pp. 412–423, 2024. DOI: [10.1134/S1990478924030049](https://doi.org/10.1134/S1990478924030049).
- [19] D. N. Barotov, “Concave continuations of Boolean-like functions and some of their properties”, *Bulletin of Irkutsk State University. Series Mathematics*, vol. 51, pp. 82–100, 2025, in Russian. DOI: [10.26516/1997-7670.2025.51.82](https://doi.org/10.26516/1997-7670.2025.51.82).

A Patterning Algorithm for the Dynamic Bin Packing Problem with Placement Groups

E. A. Brazhnikov¹, A. A. Panin¹, A. V. Ratushnyi¹

DOI: [10.18255/1818-1015-2025-2-110-131](https://doi.org/10.18255/1818-1015-2025-2-110-131)

¹Sobolev Institute of Mathematics of the Siberian Branch of the RAS, Novosibirsk, Russia

MSC2020: 68W25, 68R01

Research article

Full text in Russian

Received March 25, 2025

Revised May 20, 2025

Accepted May 21, 2025

We consider an NP-hard problem of dynamically distributing virtual machines to servers with placement groups. For each virtual machine, parameters such as required number of resources and creation and deletion timestamps are known. Each server is a composition of NUMA nodes and is placed in a rack. Large virtual machines that are placed on two nodes of a single server, and small virtual machines that impose additional conditions on their placement, are considered. Placement groups are associations of subsets of virtual machines with conflict conditions between the subsets. The goal is to pack all virtual machines using the minimum number of server racks within the considered time horizon. A heuristic based on the column generation method is proposed to solve this problem. We analyze a set of static problems at different time points necessary to form a common set of patterns used in the construction of upper bounds. The results of computational experiments on real open instances show a minor difference between the lower and upper bounds.

Keywords: bin packing problem; virtual machines; heuristics; placement groups; column generation

INFORMATION ABOUT THE AUTHORS

Brazhnikov, Evgeniy A.	ORCID iD: 0009-0004-3414-5176 . E-mail: brazhnikov.eugene@gmail.com Engineer
Panin, Artem A. (corresponding author)	ORCID iD: 0000-0002-1844-6276 . E-mail: aapanin1988@gmail.com Senior researcher, PhD
Ratushnyi, Alexey V.	ORCID iD: 0000-0001-5173-3536 . E-mail: alexeyratushny@gmail.com Leading engineer

Funding: FWNF-2022-0019.

For citation: E. A. Brazhnikov, A. A. Panin, and A. V. Ratushnyi, “A patterning algorithm for the dynamic bin packing problem with placement groups”, *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 110–131, 2025.

DOI: [10.18255/1818-1015-2025-2-110-131](https://doi.org/10.18255/1818-1015-2025-2-110-131).

Алгоритм шаблонизации для динамической задачи упаковки в контейнеры с группами размещения

Е. А. Бражников¹, А. А. Панин¹, А. В. Ратушный¹

DOI: [10.18255/1818-1015-2025-2-110-131](https://doi.org/10.18255/1818-1015-2025-2-110-131)

¹Институт математики им. С. Л. Соболева СО РАН, Новосибирск, Россия

УДК 519.8

Научная статья

Полный текст на русском языке

Получена 25 марта 2025 г.

После доработки 20 мая 2025 г.

Принята к публикации 21 мая 2025 г.

Рассматривается NP-трудная задача динамического распределения виртуальных машин по серверам с группами размещения. Для каждой виртуальной машины известны такие параметры, как необходимое количество ресурсов и временные метки создания и удаления. Каждый сервер представляет собой композицию NUMA-узлов и размещается в некоторой стойке. Рассматриваются большие виртуальные машины, размещаемые на два узла одного сервера, и маленькие, что накладывает дополнительные условия для их размещения. Группы размещения представляют собой объединения подмножеств виртуальных машин с условиями конфликта между подмножествами. Задача состоит в том, чтобы упаковать все виртуальные машины с использованием минимального количества стоек серверов в течение рассматриваемого временного горизонта. Для решения данной задачи предлагается эвристика, основанная на методе генерации столбцов. Анализируется набор статических задач в различные моменты времени, необходимых для формирования общего набора шаблонов, используемых при построении верхних оценок. Результаты вычислительных экспериментов на реальных открытых примерах указывают на незначительные расхождения между нижними и верхними границами.

Ключевые слова: задача упаковки в контейнеры; виртуальные машины; эвристики; группы размещения; генерация столбцов

ИНФОРМАЦИЯ ОБ АВТОРАХ

Бражников, Евгений Александрович	ORCID iD: 0009-0004-3414-5176 . E-mail: brazhnikov.eugene@gmail.com Стажёр-исследователь
Панин, Артём Александрович (автор для корреспонденции)	ORCID iD: 0000-0002-1844-6276 . E-mail: aapanin1988@gmail.com Старший научный сотрудник, канд. физ.-мат. наук
Ратушный, Алексей Владленович	ORCID iD: 0000-0001-5173-3536 . E-mail: alexeyratushny@gmail.com Инженер-исследователь

Финансирование: FWNF–2022–0019.

Для цитирования: Е. А. Brazhnikov, А. А. Panin, and А. V. Ratushnyi, “A patterning algorithm for the dynamic bin packing problem with placement groups”, *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 110–131, 2025.

DOI: [10.18255/1818-1015-2025-2-110-131](https://doi.org/10.18255/1818-1015-2025-2-110-131).

Введение

В последние десятилетия наблюдается существенный прогресс в развитии инфраструктуры облачных вычислений. Компании всё чаще обращаются к облачным сервисам для выполнения различных вычислительных задач, успешно применяя технологии виртуализации [1]. Последняя предоставляет удобство и гибкость взаимодействия с аппаратными ресурсами. С другой стороны, поставщики облачных ресурсов сталкиваются с задачей обслуживания множества виртуальных машин (ВМ). Быстрый рост числа ВМ влечет за собой рост числа используемых серверов и вызывает необходимость разработки эффективных методов распределения ВМ, т.е. методов решения динамической задачи упаковки в контейнеры (Temporal Bin Packing Problem, TBPP) [1–5]. В TBPP оптимизируется использование ресурсов, включая электроэнергию, что представляет собой критически важные аспекты с точки зрения эффективного функционирования облачных инфраструктур.

Сама по себе задача об упаковке в контейнеры является одной из самых исследованных задач в области исследования операций. Однако продолжают появляться её новые вариации, учитывающие динамику запросов на размещение ВМ и различные условия, накладывающие ограничения на распределение предметов по контейнерам. Например, авторы работы [6] фокусируются на онлайн-постановке задачи упаковки в контейнеры, которая часто возникает в компаниях, предоставляющих услуги облачных вычислений. Они проводят тщательный анализ производительности онлайн-алгоритмов как в худшем, так и среднем случаях. В исследовании [7] также рассматривается задача онлайн-планирования в распределённых вычислениях. Авторы предлагают гибридный подход, объединяющий машинное обучение и динамическое программирование, для удовлетворения ограничения рюкзака на общий вес. Для оценки прототипа использована симуляция задачи о 0–1 рюкзаке с неизвестными функциями полезности. Авторы работы [8] рассматривают задачи управления ресурсами для ВМ, используя подходы black-box и gray-box. Black-box подразумевает ограниченную информацию о внутреннем состоянии системы, тогда как gray-box предполагает более детальное знание. В работе представлен алгоритм Sandpiper, который адаптируется к различным уровням доступности информации о виртуальных машинах. В исследовании [9] рассматриваются методы снижения энергопотребления компьютерного кластера с помощью гибких стратегий управления состоянием узлов (включение и выключение) и порядком выполнения ожидающих задач.

Кроме того, продолжается разработка и исследование всё большего числа новых подходов. Так в работе [10] исследуются матэвристические методы для решения задачи TBPP. В ней рассматриваются как полиномиальная формулировка задачи, так и расширенная. Для них сравниваются различные эвристические алгоритмы, включая жадные методы и методы на основе генерации столбцов. Авторы [11] предлагают новый подход к планированию, чтобы минимизировать время выполнения задач. Они представляют алгоритмы, основанные на анализе характеристик задач, структуре кластера и требованиях пользователя. В исследовании [12] рассматриваются эвристические методы распределения ресурсов с учетом энергосбережения. Авторы стремились предложить подходы, которые учитывают особенности энергетической эффективности при принятии решений и способствуют балансу между производительностью вычислительных систем и ресурсосбережением. В работе [13] предлагается подход lookahead control, который предполагает анализ будущего состояния системы для принятия решений в текущий момент времени. Такой алгоритм позволяет предварительно реагировать на изменения в нагрузке, обеспечивая оптимальное распределение вычислительных ресурсов с учетом энергосбережения и уровня производительности. Авторы [14] исследуют задачу размещения виртуальных машин с учетом затрат на энергопотребление и миграцию. Они представляют алгоритм rMapper, который основывается на анализе энергетической эффективности и стоимости миграции. В работе [15] проводится анализ времени запуска виртуальных машин в облачных вычислениях. Авторы осуществляют эмпирическое исследование, оценивая

различные факторы, которые могут влиять на время запуска. Работа включает в себя анализ различных параметров и условий, таких как размеры ВМ, характеристики сети и оборудования, а также эффект использования различных технологий виртуализации.

Данная работа является расширенной версией работы [16] и продолжает развитие предложенных алгоритмов. Одним из основных отличий является замена постановки задачи с конфликтами между типами виртуальных машин на постановку с группами размещений, при этом задача остается NP-трудной как обобщение классической задачи упаковки в контейнеры. Для вычисления нижних границ, которые необходимы для оценки качества алгоритмов построения приближенных решений на реалистичных примерах большой размерности, мы провели адаптацию идеи генерации для учёта конфликтов. Для получения верхней оценки и приближенного решения мы предлагаем несколько алгоритмов, основанных на шаблонизации упаковки серверов. Нижние и верхние границы также могут быть использованы для оценки эффективности онлайн-алгоритмов. Основным вкладом данного исследования состоит в разработке математической модели в форме целочисленного линейного программирования (ЦЛП) и анализе нескольких вариаций алгоритма шаблонизации упаковки. Мы также приводим результаты численных экспериментов, полученных на открытых данных.

Статья организована следующим образом. В разделе 1 вводятся необходимые обозначения и математическая модель рассматриваемой задачи. В разделе 2 мы приводим описание процедуры генерации столбцов и её модификации. Раздел 3 посвящён алгоритмам построения верхних оценок и решения задачи. В разделе 4 содержатся описание используемых данных, численных экспериментов и их результаты. В заключении мы кратко подводим итоги и обсуждаем возможные направления дальнейших исследований.

1. Математическая модель

Для описания математической модели введем множество однородных серверов S , достаточное для размещения всех ВМ. Сервера используют «архитектуру с неравномерным доступом к памяти» (NUMA) [17]. Каждый сервер $s \in S$ разделен на N узлов. На узле $n \in N$ доступно количество C_{nr} ресурса $r \in R$, где R — множество всех ресурсов. Мы считаем, что все серверы идентичны, однако узлы одного сервера могут отличаться между собой, т. е. допускается, что $C_{n_1r} \neq C_{n_2r}$ при $n_1 \neq n_2$.

Каждый из серверов располагается на некоторой стойке $f \in F$. На каждой стойке нельзя размещать больше D серверов. Так как мы стремимся минимизировать число стоек, а не число серверов, без ограничения общности будем считать, что каждая из них заполнена серверами полностью. Множество серверов стойки f обозначим S_f .

Требуется разместить множество виртуальных машин M , состоящее из двух подмножеств: M^{small} и M^{large} . Множество M^{small} содержит маленькие ВМ, каждая из которых размещается на одном узле сервера. M^{large} — множество больших ВМ, которые состоят из двух идентичных частей, размещаемых на различных узлах одного сервера. Каждая виртуальная машина $m \in M$ имеет некоторое требование b_{mr} ресурса $r \in R$ для каждого узла. Таким образом, маленькая (большая) машина m занимает b_{mr} ($2 \cdot b_{mr}$) ресурса r на некотором сервере во все моменты времени $t \in T$ такие, что $\alpha_m \leq t < \omega_m$, где α_m и ω_m являются, соответственно, временами создания и удаления m .

Кроме этого, некоторые подмножества виртуальных машин объединены в группы. Каждая группа $g \in G$ разбита на множество подгрупп P_g , называемых "партициями". Каждая пара подгрупп (p_i, p_j) из множества P_g группы g конфликтует друг с другом на уровне стоек. Т.е. никакие две виртуальные машины $m_1 \in p_i$ и $m_2 \in p_j$ не могут располагаться на одной стойке f , если они пересекаются по времени. Для удобства, мы будем обозначать множество машин из подгруппы p группы g как $M_{gp} \subset M$. Также определим множества виртуальных машин $M_t = M_t^{small} \cup M_t^{large}$, существующих в определенный момент времени t .

Для определения математической модели, введём следующие переменные:

- $x_{fsmn} \in \{0, 1\}$ равняется 1, если маленькая ВМ или первая часть большой ВМ $m \in M$ располагается на узле n сервера s стойки f , и 0 иначе;
- $y_{fsmn} \in \{0, 1\}$ равняется 1, если вторая часть большой ВМ $m \in M^{large}$ располагается на NUMA-узле n сервера s стойки f , и 0 иначе;
- $z_f \in \{0, 1\}$ равняется 1, если стойка f была активна, т. е. её ресурсы использовались хотя бы в один момент времени $t \in T$, и 0 иначе.

Используя эти обозначения, математическую модель можно представить в следующем виде:

$$\sum_{f \in F} z_f \rightarrow \min, \quad (1)$$

$$\sum_{n=1}^N x_{fsmn} = \sum_{n=1}^N y_{fsmn}, f \in F, s \in S_f, m \in M^{large}, \quad (2)$$

$$x_{fsmn} + y_{fsmn} \leq 1, f \in F, s \in S_f, n \in N, m \in M^{large}, \quad (3)$$

$$\sum_{f \in F} \sum_{s \in S_f} \sum_{n \in N} x_{fsmn} = 1, m \in M, \quad (4)$$

$$\sum_{m \in M_t} b_{mr} x_{fsmn} + \sum_{m \in M_t^{large}} b_{mr} y_{fsmn} \leq C_{nr}, f \in F, s \in S_f, n \in N, t \in T, \quad (5)$$

$$\sum_{s \in S_f} \sum_{n \in N} x_{fsmn} \leq z_f, f \in F, m \in M, \quad (6)$$

$$\sum_{s \in S_f} \sum_{n \in N} (x_{fsm_1} + x_{fsm_2}) \leq 1, \quad (7)$$

$$f \in F, g \in G, t \in T, p_1, p_2 \in P_g, m_1 \in M_{gp_1} \cap M_t, m_2 \in M_{gp_2} \cap M_t, \\ x_{fsmn}, y_{fsmn}, z_f \in \{0, 1\}, f \in F, s \in S, n \in N, m \in M. \quad (8)$$

Целевая функция (1) определяет размер пула ресурсов (число используемых стоек), который мы стремимся минимизировать. Ограничения (2) запрещают размещать большие виртуальные машины на NUMA-узлы разных серверов, а ограничения (3) запрещают класть их на один NUMA-узел. Согласно ограничению (4) все виртуальные машины должны быть размещены. Ресурсные ограничения отражены в (5). Ограничения (6) запрещают использовать стойки, которые не входят в пул ресурсов. Ограничения (7) запрещают в каждый момент времени класть виртуальные машины из разных подгрупп одной группы на одну и ту же стойку, но позволяют класть на одну стойку непересекающиеся по времени ВМ. В (8) задаются области определения переменных.

2. Генерация шаблонов и нижние оценки

Каждой ВМ m сопоставим вектор $(b_{m1}, \dots, b_{mR})$, который мы назовём типом. Обозначим за L множество всех уникальных типов ВМ из M . Этот вектор представляет набор значений, описывающих требования данной ВМ. Каждая виртуальная машина m имеет некоторый интервал времени существования $[\alpha_m, \omega_m)$, который определяется пользователем. Число таких уникальных интервалов может совпадать с числом виртуальных машин. В то же время, число уникальных типов обычно очень ограничено, и считается, что $|L| \ll |M|$. Это связано с тем, что на практике чаще всего используются типовые конфигурации виртуальных машин, соответствующие стандартным сценариям использования. С примерами реальных типов виртуальных машин можно ознакомиться, например, в [18].

Для получения нижних оценок мы воспользовались модифицированной процедурой генерации столбцов, описанной в [4], для похожей задачи. Рассматривается некоторый момент времени $\hat{t}$.

Например, можно рассмотреть $\hat{t} = \arg \max_{t \in T} \sum_{m \in M_t} b_{mr}$ для некоторого r . В фиксированный момент времени виртуальные машины, имеющие одинаковый тип $l \in L$, неотличимы. Обозначим J множество всех возможных упаковок одного сервера. Элемент $j \in J$ можно представить в виде вектора $(a_{lj}, l \in L)$, где a_{lj} равняется целому числу виртуальных машин типа l . Пусть некоторое подмножество $J' \subset J$ является достаточным набором шаблонов для того, чтобы можно было упаковать все виртуальные машины данного примера.

Для описания процедуры получения нижней оценки сначала введём несколько дополнительных моделей. Первая модель называется координирующей задачей (master problem) линейного программирования и отвечает за назначение виртуальных машин на серверы по шаблонам. Переменная x_j равняется количеству серверов, которые используют шаблон j .

$$\min \sum_{j \in J'} x_j, \quad (9)$$

$$\sum_{j \in J'} a_{lj} x_j \geq n_l, l \in L, \quad (10)$$

$$x_j \geq 0, j \in J'. \quad (11)$$

Целевая функция (9) минимизирует число использованных серверов. Неравенства (10) гарантируют, что все виртуальные машины будут упакованы. Здесь $\hat{n}_l$ равняется числу виртуальных машин, имеющих тип l .

Пусть x_j^* — значения переменных x_j в оптимальном решении задачи (9)–(11), а λ_l^* — значения соответствующих двойственных переменных. Если неравенство $\sum_{l \in L} a_{lj} \lambda_l^* \leq 1$ выполняется для всех шаблонов $j \in J$, то решение x такое, что $x_j = x_j^*$ при $j \in J'$ и 0 иначе, является оптимальным решением задачи (9)–(11) для всего множества шаблонов J , а $LB = \lceil \sum_{j \in J'} x_j^* \rceil$ является нижней оценкой на оптимальное значение (1) при $D = 1$. Стоит отметить, что LB подсчитывается для числа серверов, однако её можно легко адаптировать для целевой функции (1), поделив на вместимость каждой стойки D .

Проверять неравенство выше для всех $j \in J$ достаточно сложно из-за экспоненциальной мощности множества J . Чтобы упростить задачу, можно рассмотреть задачу генерации шаблонов (pricing problem) для поиска следующего допустимого шаблона, которого ещё нет в J' и который мог бы уменьшить значение целевой функции (9).

Пусть переменные y_l^n определяют количество маленьких виртуальных машин (или первых частей больших машин) типа $l \in L$, расположенных на NUMA-узле $n \in N$, а z_l^n определяют количество вторых частей больших машин типа $l \in L^{large}$, расположенных на NUMA-узле $n \in N$. Здесь L^{small} и L^{large} являются наборами типов виртуальных машин, соответствующих множествам M^{small} и M^{large} . Тогда модель для задачи генерации шаблонов может быть записана следующим образом:

$$\max \alpha = \sum_{n \in N} \sum_{l \in L} \lambda_l^* y_l^n, \quad (12)$$

$$y_l^n \leq \sum_{\substack{k \in N \\ k \neq n}} z_l^k, l \in L^{large}, n \in N, \quad (13)$$

$$\sum_{n \in N} z_l^n = \sum_{n \in N} y_l^n, l \in L^{large}, \quad (14)$$

$$\sum_{l \in L} b_{lr} y_l^n + \sum_{l \in L^{large}} b_{lr} z_l^n \leq C_{nr}, r \in R, n \in N, \quad (15)$$

$$y_l^n \geq 0, integer, l \in L, n \in N, \quad (16)$$

$$z_l^n \geq 0, integer, l \in L^{small}, n \in N. \quad (17)$$

Целевая функция (12) отвечает за ценность упаковки. Равенства (13) и (14) гарантируют правильность расположения больших виртуальных машин. Неравенства (15) ограничивают доступность ресурсов на сервере.

Данный подход не учитывает наличие конфликтов между группами. Кажется, что их добавление способно положительно повлиять на нижнюю оценку. Будем называть тройку (l, g, p) расширенным типом виртуальных машин. Множество расширенных типов обозначим также L . Такая модификация позволяет нам достаточно удобно добавить необходимые ограничения в модель (12)–(17). Добавление всех возможных конфликтов не всегда целесообразно из-за существенного роста вычислительной сложности. Имеет смысл ограничить число групп, которые используются для подобного расширения числа типов. Например, можно рассмотреть группы с наибольшим числом виртуальных машин в определенный момент времени интервала T .

Пусть $\delta_{ij} = 1$, если два расширенных типа i и $j \in L$ не конфликтуют, и $\delta_{ij} = 0$ иначе. Для модификации модели (12)–(17) предлагается добавить следующие ограничения:

$$\chi_l \geq \sum_{n \in N} \frac{y_l^n}{M}, \quad l \in L, \quad (18)$$

$$\chi_i + \chi_j \leq 1 + \delta_{ij}, \quad i, j \in L, \quad i \neq j, \quad (19)$$

$$\chi_l \in \{0, 1\}, \quad l \in L. \quad (20)$$

Пусть α^* — значение целевой функции в оптимальном решении задачи (12)–(17). Если $\alpha^* \leq 1$, то мы считаем, что смогли найти оптимальное решение задачи (9)–(11) и соответственно нижнюю оценку для задачи (1)–(8). Если $\alpha^* > 1$, то мы получили новый шаблон упаковки, которые можно включить во множество J' . Общая схема поиска нижней оценки представлена ниже.

Алгоритм вычисления нижней оценки:

Шаг 1: Пусть $\alpha^* = +\infty$.

Шаг 2: Решаем задачу, двойственную к (9)–(11) и сохраняем значения λ_l .

Шаг 3: Решаем задачу (12)–(20). Сохраняем значение целевой функции α^* и полученный шаблон.

Шаг 4: Если $\alpha^* \leq 1$, то мы нашли нижнюю оценку $LB = \lceil \sum_{i \in L} n_i \lambda_i^* \rceil$, иначе добавляем новый шаблон в J' и возвращаемся на шаг 2.

Заметим, что добавление конфликтов в задачу (12)–(17) действительно способно увеличить значение LB . Рассмотрим две ВМ, первая из которых занимает весь ресурс $r_1 \in R$ на сервере, а вторая весь ресурс $r_2 \in R$. В случае когда они принадлежат разным партициям одной группы, приведённая выше модификация генерации столбцов позволит нам получить нижнюю оценку равную двум серверам.

3. Верхние оценки

Для построения верхней оценки для задачи (1)–(8) мы предлагаем двухэтапный эвристический алгоритм, который основан на решении статической задачи для некоторого момента времени t . Алгоритм опирается на разумное предположение о том, что плотная упаковка виртуальных машин в момент максимальной нагрузки оказывает значительный эффект на остальные моменты времени и минимизирует общее количество используемых стоек. Шаблоны, построенные методом генерации столбцов, обеспечивают высокую плотность упаковки. Отсюда вытекает основная идея решения статической задачи — упаковка ВМ по шаблонам. Кроме этого, вычислительные эксперименты показывают, что выбор подходящего момента времени часто является непростой задачей. По этой причине будем рассматривать момент времени t как параметр алгоритма, придерживаясь предположения о том, что качество упаковки в данный момент времени ключевым образом влияет на верхнюю оценку.

3.1. Базовый алгоритм

Первый этап алгоритма заключается в плотной упаковке виртуальных машин из множества M_t в заданное число стоек. Так как мы рассматриваем только один момент времени, то, не ограничивая общности, можно отказаться от информации о временах создания и удаления и работать только с типами ВМ. В начале создаются пустые сервера, на которые назначаются шаблоны, полученные генерацией столбцов без конфликтов [4] для множества M_t . На первом этапе алгоритма упаковка происходит только в соответствии с шаблонами, а количество серверов каждого шаблона можно определить, решив задачу (9)–(11) с целыми переменными. Кроме серверов на первом этапе создаётся заданное количество стоек, которые изначально пусты. Их количество определяется нижней границей.

Следующим шагом ВМ назначаются на серверы с учётом конфликтов на уровне партиций, а сами серверы размещаются на стойки. Для конфликтующих виртуальных машин предварительно создаётся порядок обработки: группы, а также подгруппы внутри групп сортируются по убыванию суммарной нагрузки. Далее подгруппы разделяются на множества P_1 и P_2 . Множество P_1 состоит из подгрупп, которые требуют для размещения более K серверов, где K – параметр алгоритма. Все остальные подгруппы составляют P_2 . В первую очередь с помощью эвристического алгоритма размещаются подгруппы из множества P_1 , а также назначаются серверы на стойки. Затем подгруппы из множества P_2 распределяются по свободным местам уже размещённых серверов. Виртуальные машины из P_2 , которые не были распределены из-за заполненности серверов или ограничения конфликтов, располагаются на оставшихся неразмещённых серверах, после чего решается модель ЦЛП для размещения данных серверов на стойки с учётом конфликтов. Свободные виртуальные машины, которые не конфликтуют с другими, размещаются на самом последнем шаге статического этапа. Конкретные значения параметра K , которые использовались при тестировании, описаны в разделе 4. Алгоритм упаковки множества P_1 приведён ниже.

Процедура упаковки подгрупп из множества P_1 :

Шаг 1: Выбираем ещё не просмотренные в порядке убывания нагрузки группу g и подгруппу p такие, что $p \in P_g \cap P_1$.

Шаг 2: Размещаем ВМ партиции p в активные серверы (на которых уже размещена хотя бы одна ВМ) согласно заданным на них шаблонам следующим образом:

1. Выделяем множество активных (с хотя бы одним размещённым сервером и хотя бы одной размещённой ВМ) стоек F_p , не содержащих ВМ из других подгрупп группы g .
2. Вычисляем вектор (k_l) , где k_l – число неразмещённых ВМ типа l в текущей подгруппе p .
3. Для каждой необработанной стойки $f \in F_p$ вычисляем вектор (s_{fl}) , где s_{fl} – число свободных мест для виртуальных машин типа $l \in L$.
4. Вычисляем значение $m_f = \sum_{l \in L} \min(s_{fl}, k_l)$, которое равняется числу неразмещённых ВМ подгруппы p , которые стойка f может вместить по шаблонам уже размещённых на ней серверов.
5. Сортируем необработанные стойки по убыванию m_f .
6. Выбираем следующую необработанную стойку f . Распределяем неразмещённые ВМ в свободные места активных серверов стойки и считаем её обработанной.

Шаг 3: Если все ВМ подгруппы p размещены, то возвращаемся на шаг 1. Иначе решаем модель (9)–(11) с целыми переменными для оставшихся ВМ из p и ограничением, что можно использовать только неразмещённые пустые серверы. Затем размещаем эти серверы с уже распределёнными машинами из p на стойки следующим образом:

1. Сортируем стойки из F_p по убыванию числа свободных мест для серверов.

2. Пока не будут размещены все серверы или не будут обработаны все стойки, обходим стойки в данном порядке и на каждую размещаем столько серверов, сколько возможно.

Возвращаемся на шаг 1.

Подгруппы из множества P_2 распределяются на частично заполненные стойки как на шаге 2. Оставшиеся после этого ВМ размещаются аналогично шагу 3, но теперь серверы размещаются на стойки посредством решения задачи ЦЛП. Для её описания введём необходимые обозначения:

- $\bar{S}$ — множество неразмещённых серверов;
- $\bar{G}$ — множество групп с подгруппами из множества P_2 ;
- $\bar{S}_{gp}$ — множество серверов, содержащих ВМ из подгруппы $p \in P_g$ группы $g \in \bar{G}$;
- $\bar{D}_f$ — количество свободных мест для серверов на стойке f .

Определим переменные:

- $x_{fs} \in \{0, 1\}$ равняется 1, если сервер s размещён на стойке f , и 0 иначе.

Получаем математическую модель:

$$\sum_{f \in F, s \in \bar{S}} x_{fs} \rightarrow \max, \quad (21)$$

$$\sum_{f \in F} x_{fs} \leq 1, s \in \bar{S}, \quad (22)$$

$$\sum_{s \in \bar{S}} x_{fs} \leq \bar{D}_f, f \in F, \quad (23)$$

$$x_{fs_1} + x_{fs_2} \leq 1, f \in F, g \in G, p_1, p_2 \in P_g : p_1 < p_2, s_1 \in \bar{S}_{gp_1}, s_2 \in \bar{S}_{gp_2}, \quad (24)$$

$$x_{fs} \in \{0, 1\}, f \in F, s \in \bar{S}. \quad (25)$$

Целевая функция (21) определяет число серверов из $\bar{S}$, которые можно разместить, не создавая конфликтов. Ограничение (22) запрещает размещать сервер более чем на одной стойке. Неравенства (23) ограничивают вместимость стоек. Наконец, ограничение (24) определяет условие на конфликты между подгруппами.

Целесообразно выбирать параметр K так, чтобы большая часть подгрупп попала во множество P_1 , то есть чтобы значение K было относительно невелико. Это связано с тем, что на практике, если большая часть подгрупп принадлежит множеству P_2 , то ограничений типа (24) становится слишком много, а время решения задачи исчисляется сутками.

Стоит отметить, что статический этап алгоритма не гарантирует размещения всех ВМ из множества M_t . Но основные шаги статического этапа на этом заканчиваются. На втором этапе алгоритма происходит упаковка ВМ из множеств $M_t^{before} = \{m \in M | \omega_m \leq t\}$ и $M_t^{after} = \{m \in M | \alpha_m > t\}$, а после этого при необходимости создаются дополнительные стойки и упаковываются неразмещённые ВМ, в том числе те, которые не были размещены на первом этапе.

Рассмотрим шаги упаковки ВМ из M_t^{after} . Предварительно создаётся порядок обработки групп и виртуальных машин. Пусть G_t^{after} — множество групп, которые содержат ВМ из M_t^{after} . Группы сортируются в порядке убывания числа ВМ из M_t^{after} , которые они содержат. Помимо этого виртуальные машины в каждой группе отдельно сортируются по возрастанию α_t . Таким же образом создаётся порядок для свободных ВМ. Идея подобной сортировки в том, чтобы в первую очередь упаковать виртуальные машины, наиболее близкие по периоду жизни к моменту времени t .

Группы из G_t^{after} и виртуальные машины из M_t^{after} , обрабатываются в предложенном порядке жадной процедурой First Fit (первый подходящий), которая размещает ВМ по серверам стоек, не создавая конфликтов. При этом шаблоны серверов на данном шаге сохраняются и каждая ВМ размещается в свободную на время её существования ячейку сервера. Сохранение шаблонов необходимо для того, чтобы поддерживать плотный вариант упаковки и в другие моменты времени

помимо t . После упаковки ВМ из групп размещений производится упаковка свободных ВМ в определённом ранее порядке. Аналогичным образом обрабатывается множество M_t^{before} , но сортировка ВМ происходит по убыванию ω_t .

В силу того, что шаблоны были сгенерированы для одного момента t , часть виртуальных машин не будет упакована. Следующим шагом происходит отказ от шаблонной структуры серверов, а стойки дополняются пустыми серверами до предельного значения. Отказ от шаблонной структуры серверов может также дать возможность упаковать некоторые из неразмещённых ВМ множества M_t .

В соответствии с идеей алгоритма в первую очередь после отказа от шаблонов должна производиться попытка упаковать неразмещённые ВМ из множества M_t . Также для уменьшения числа используемых серверов и стоек выгоднее сначала размещать конфликтующие ВМ. Аналогично множеству G_t^{after} определяется множество G_t и создаётся порядок обработки групп из этого множества. Виртуальные машины каждой обрабатываемой группы размещаются на стойки и серверы с помощью процедуры First Fit с учётом ограничения конфликтов. После этого аналогично упаковке в шаблоны производится упаковка неразмещённых конфликтующих ВМ из множеств M_t^{after} и M_t^{before} . Свободные виртуальные машины распределяются также с помощью процедуры First Fit в аналогичном порядке. При необходимости создаются новые стойки и сервера.

3.2. Модификации базового алгоритма

Помимо стандартного алгоритма были предложены и протестированы и другие, основанные на данном, но использующие альтернативные процедуры на некоторых шагах.

3.2.1. Альтернативная процедура упаковки для множества P_1

При размещении виртуальных машин из одной подгруппы на статическом этапе может оказаться важным задействовать как можно меньшее число стоек. Например, если алгоритм разместит 5 виртуальных машин одной подгруппы на 5 стойках, когда была возможность уместить их на одной, то места для других подгрупп может не остаться из-за конфликтов. Предлагается рассмотреть обратную стратегию, которая старается максимизировать плотность упаковки стоек, т. е. рассматривается все пространство стойки с учетом пустых мест под серверы. Однако даже такая процедура не лишена недостатков. Если некоторые стойки заполняются до предела, то алгоритм будет иметь меньшую ширину выбора для упаковки последующих групп. По этой причине были протестированы обе стратегии.

3.2.2. Альтернативные шаблоны упаковки для серверов

Одним из способов адаптировать шаблоны под задачу (1)–(8) является генерация столбцов с конфликтами, описанная в разделе 2. В проведённых экспериментах в качестве групп, участвующих в расширении типов, использовались 2 группы, содержащие наибольшее число ВМ. Уточнения требует только тот факт, что при переходе на второй этап алгоритмов места в шаблонах под конфликтующие типы (l, g, p) преобразуются в места под обычные типы l , то есть на втором этапе происходит отказ от шаблонов с конфликтами. Это необходимо, потому что конфликтующие шаблоны ограничивают конфликты только в момент времени t .

Ещё одной модификацией является использование универсальных шаблонов. Если прошлый подход адаптировал ограничение на конфликты, то данный частично адаптирует статические шаблоны под динамическую задачу. В некоторых случаях распределение типов виртуальных машин может значительно отличаться в различные моменты времени. Это приводит к ситуации, когда шаблоны в динамической задаче быстро теряют эффективность при последовательном отступе от ключевого момента времени. Для решения данной проблемы числа n_i из ограничения (10) можно вычислять в виде усреднённых по окрестности радиуса r_{cg} момента t .

Table 1. Analysis of instances

Таблица 1. Анализ примеров

Семейство примеров	Число моментов времени	Число ВМ	Число групп с партициями	Среднее число ВМ в группе
DMP_l_2	379.68	38074.36	354.36	69.60
DMP_l_4	311.80	42577.24	425.72	63.27
DMP_m_2	343.88	32460.80	273.60	69.51
DMP_m_4	276.00	32767.40	271.36	62.99
DMP_s_2	274.48	22066.76	124.88	68.95
DMP_s_4	235.72	23220.84	121.88	62.06
LPO_l_2	338.28	64703.12	605.92	106.85
LPO_l_4	280.40	68349.76	671.84	101.80
LPO_m_2	153.72	28825.24	272.20	105.89
LPO_m_4	115.04	27328.32	271.76	100.59
LPO_s_2	70.28	13090.92	124.56	105.20
LPO_s_4	55.24	12781.92	124.92	102.20
MP_l_2	308.88	44820.88	641.40	69.96
MP_l_4	217.60	49286.20	737.04	66.91
MP_m_2	136.16	19213.04	273.84	70.08
MP_m_4	84.28	18679.64	278.64	66.98
MP_s_2	63.32	8838.44	126.16	69.98
MP_s_4	37.24	8213.20	121.92	67.46

3.2.3. Выбор момента времени

Момент времени с максимальной нагрузкой не всегда является удачным, и в некоторых случаях имеет смысл руководствоваться другими правилами. Допустим, у нас имеется некоторое решение, которое было получено выбором момента t в качестве начального. Возможна ситуация, когда в момент времени $t_1 \neq t$ задействуется большее число активных стоек r_{t_1} , чем в момент t . Предполагается, что применение алгоритма в другие моменты t , в т. ч. $r_t = \max_{\tau \in T} r_\tau$, может улучшить итоговый результат. Рассмотрим итеративную процедуру MultiStart, которая на каждой итерации строит верхнюю оценку, но в некоторый ещё не рассмотренный момент времени. Моментов, где задействовано наибольшее число стоек, может быть очень много, обработка всех занимала бы длительное время. Пусть t_{prev} — момент времени, использованный на предыдущей итерации. На каждой следующей итерации алгоритм применяется для ближайшего к t_{prev} момента времени с максимальным r_t . Отметим, что если максимальное значение достигается в момент времени t_{prev} и сохраняется в некоторой окрестности, то выбирается ближайший момент времени, на котором достигается максимальное значение r_t вне указанной окрестности. Это позволяет избежать запуска новой итерации в момент, где распределение виртуальных машин остаётся практически неизменным. Процедура останавливается, если выбирается момент, использованный на предыдущих итерациях, или в случае, когда для данного решения максимум активных стоек единственный и достигается в момент t_{prev} . В качестве ответа выдаётся лучшее решение по всем итерациям.

4. Численные эксперименты

Алгоритмы реализованы в среде Python. Эксперимент проводился на компьютере с процессором AMD Ryzen 5 5600H и памятью 16 GB. Для проведения численных экспериментов были использованы открытые примеры [19], которые имитируют поведение реальной работы облачного сервиса. Различная информация о примерах приведена в таблице 1. Каждый пример принадлежит некоторому множеству, которое определяется 3 параметрами — семейством, размером и числом NUMA-узлов. Обозначения l, m и s отвечают за размер примера, т. е. за количество виртуальных машин

Table 2. Analysis of the parameter K Таблица 2. Анализ параметра K

Алгоритм	Best	$K = 0$	$K = 4$	$K = 9$	$K = 14$	$K = 19$	$K = 30$
Classic	6.98 %	7.50 %	7.98 %	8.26 %	8.32 %	8.33 %	8.34 %
Classic MultiStart	6.21 %	6.70 %	7.26 %	7.48 %	7.54 %	7.55 %	7.54 %
Dense	7.09 %	7.58 %	7.88 %	8.29 %	8.31 %	8.33 %	8.32 %
Dense MultiStart	6.18 %	6.71 %	7.18 %	7.49 %	7.54 %	7.54 %	7.53 %
Mean	7.02 %	7.55 %	8.00 %	8.29 %	8.30 %	8.30 %	8.29 %
Mean MultiStart	6.25 %	6.75 %	7.25 %	7.51 %	7.49 %	7.50 %	7.50 %
Dense Mean	7.13 %	7.61 %	7.90 %	8.31 %	8.27 %	8.29 %	8.28 %
Dense Mean MultiStart	6.21 %	6.74 %	7.18 %	7.49 %	7.50 %	7.50 %	7.51 %
CGC	8.23 %	8.58 %	9.52 %	9.60 %	9.69 %	9.72 %	9.74 %
CGC MultiStart	7.55 %	7.93 %	8.86 %	8.99 %	9.10 %	9.12 %	9.13 %
Dense CGC	8.63 %	8.99 %	9.66 %	9.62 %	9.74 %	9.73 %	9.76 %
Dense CGC MultiStart	7.95 %	8.35 %	9.05 %	8.98 %	9.15 %	9.14 %	9.16 %

и количество моментов времени. Так обозначения DMP, LPO, MP характеризуют семейства примеров, где LPO — примеры только с достаточно большими группами партиций, MP — примеры как с большими группами партиций, так и с маленькими, а DMP — расширение примеров MP с добавлением виртуальных машин, которые не принадлежат ни одной группе, т. е. не имеют никаких конфликтов. Буквы l, m и s обозначают размер примера, то есть количество виртуальных машин и моментов времени. Цифры 2 и 4 указывают, сколько NUMA-узлов доступно на серверах. Каждая группа состоит из 25 примеров, а суммарно бенчмарк включает 450 примеров. В таблице 1 в столбцах приводятся средние значения по всем примерам группы.

Далее в таблицах приводится анализ работы нескольких алгоритмов, которые включают в себя следующие:

1. Classic. Используются стандартные шаблоны и стандартная процедура упаковки ВМ множества P_1 .
2. Dense Packing. Используются стандартные шаблоны, но применяется альтернативная процедура упаковки ВМ множества P_1 .
3. Mean. Используются универсальные шаблоны и стандартная процедура упаковки ВМ множества P_1 .
4. Dense Mean. Используются универсальные шаблоны и альтернативная процедура упаковки ВМ множества P_1 .
5. CGC. Используются шаблоны с конфликтами и стандартная процедура упаковки ВМ множества P_1 .
6. Dense CGC. Используются шаблоны с конфликтами и альтернативная процедура упаковки ВМ множества P_1 .

В таблице 2 представлены результаты численных экспериментов, отражающие зависимость эффективности каждого алгоритма от параметра K . В клетках приводятся средние значения $GAP = (UB - LB)/UB$, вычисленные по всем примерам бенчмарка. В столбце Best приводится средний GAP , вычисленный с помощью выбора оптимального значения K для каждого конкретного примера. Было замечено, что результаты при $K > 30$ совпадают с результатами, полученными при $K = 30$, поэтому такие случаи не рассматриваются далее в анализе. В экспериментах с универсальными шаблонами использовался радиус $r_{cg} = 10$. Время работы решателя SCIP [20] версии 8.0.4, используемого для решения ЦПП проблем, здесь и далее было ограничено 5-ю минутами.

В среднем, все рассматриваемые алгоритмы показывают лучшую производительность при параметре $K = 0$. Дополнительно, для каждого значения K было подсчитано число примеров, на которых результат совпадает с лучшим из всех результатов для этого же примера. Во всех случаях при $K = 0$

Table 3. Algorithms without CGC and MultiStart**Таблица 3.** Алгоритмы без процедур CGC и MultiStart

Группа примеров	Classic		Dense		Mean		Dense Mean	
	Best	$K = 0$	Best	$K = 0$	Best	$K = 0$	Best	$K = 0$
LPO_s_2	6.93 %	7.90 %	6.83 %	7.98 %	7.15 %	8.04 %	7.61 %	8.57 %
LPO_s_4	11.59 %	12.55 %	10.51 %	10.67 %	11.59 %	12.55 %	10.51 %	10.67 %
LPO_m_2	6.01 %	7.05 %	6.39 %	7.15 %	6.01 %	7.05 %	6.39 %	7.15 %
LPO_m_4	6.30 %	6.55 %	6.26 %	6.69 %	6.39 %	6.64 %	6.26 %	6.69 %
LPO_l_2	12.55 %	12.93 %	12.58 %	12.93 %	12.55 %	12.93 %	12.58 %	12.93 %
LPO_l_4	8.51 %	8.69 %	8.44 %	8.73 %	8.51 %	8.69 %	8.44 %	8.73 %
MP_s_2	7.66 %	8.33 %	7.11 %	8.06 %	7.49 %	8.33 %	7.11 %	8.06 %
MP_s_4	10.35 %	10.52 %	11.64 %	11.85 %	11.07 %	11.24 %	11.64 %	11.85 %
MP_m_2	7.17 %	8.00 %	7.51 %	8.36 %	7.17 %	8.00 %	7.51 %	8.36 %
MP_m_4	5.67 %	6.40 %	6.67 %	7.00 %	5.57 %	6.30 %	6.67 %	7.00 %
MP_l_2	10.29 %	10.78 %	10.41 %	10.82 %	10.29 %	10.78 %	10.41 %	10.82 %
MP_l_4	5.42 %	5.73 %	5.49 %	5.80 %	5.38 %	5.69 %	5.49 %	5.80 %
DMP_s_2	4.46 %	4.93 %	4.53 %	5.48 %	4.46 %	4.93 %	4.53 %	5.48 %
DMP_s_4	2.78 %	3.15 %	2.61 %	2.89 %	2.78 %	3.23 %	2.61 %	2.89 %
DMP_m_2	7.13 %	7.67 %	7.40 %	7.84 %	7.13 %	7.67 %	7.40 %	7.84 %
DMP_m_4	1.98 %	2.26 %	2.20 %	2.42 %	1.98 %	2.26 %	2.20 %	2.42 %
DMP_l_2	8.65 %	8.97 %	8.78 %	9.31 %	8.65 %	8.97 %	8.78 %	9.31 %
DMP_l_4	2.22 %	2.56 %	2.22 %	2.48 %	2.22 %	2.56 %	2.22 %	2.48 %
GAP_{Mean}	6.98 %	7.50 %	7.09 %	7.58 %	7.02 %	7.55 %	7.13 %	7.61 %

достигнуто наибольшее значение. Например, для Classic MultiStart при $K = 0$ это число составляет 326, а минимально оно при $K = 19$ и равняется 228. С увеличением значения параметра K практически монотонно происходит ухудшение средних результатов алгоритмов. Это означает, что алгоритмы имеют достаточно высокую эффективность, когда подгруппы не разделяются на большие и маленькие по количеству занимаемых серверов. Значение параметра $K = 0$ может быть предпочтительным, когда мы не обладаем достаточным временем для перебора различных случаев.

Рассмотрим пример статической задачи, который может объяснить ухудшение. Пусть имеется 2 группы, в каждой по 2 подгруппы. Подгруппа 1.1 для размещения требует $3/4$ ресурса сервера, а подгруппа 1.2 требует $1/4$. Подгруппа 2.1 требует $2/4$ ресурса, а подгруппа 2.2 требует $1/4$. Имеется 2 стойки f_1 и f_2 , на каждой к моменту упаковки данных групп осталось по 1 ячейке для пустого сервера. При $K = 0$ для упаковки групп данных стоек хватит, согласно базовому алгоритму для 1.1 будет выделен пустой сервер и подгруппа будет упакована на f_1 , аналогично 1.2 на f_2 , а 2.1 и 2.2 займут свободное место на f_2 и f_1 , соответственно. Имеет место такая последовательность упаковки, потому что группы сортируются по нагрузке, после чего внутри самих групп сортируются подгруппы. Теперь пусть $K = 1/2$, тогда подгруппа 1.1 будет добавлена в P_1 и упакована на первом этапе в f_1 , а подгруппы 1.2, 2.1 и 2.2, напротив, будут добавлены в P_2 . На втором этапе в первую очередь происходит упаковка в размещённые на стойках сервера. На f_1 для 1.1 размещён сервер, а на f_2 осталась только пустая ячейка для сервера. Группы и подгруппы внутри снова сортируются по нагрузке, поэтому сначала будет упаковываться группа 2 и подгруппа 2.1, которая займёт только оставшуюся часть сервера на f_1 в размере $\frac{1}{4}$, но не будет упакована полностью. В итоге к следующему шагу алгоритма не будут полностью упакованы 2.1 и 2.2. Для их упаковки потребуется дополнительная стойка, так как f_1 полностью заполнена, а на одной f_2 подгруппы размещаться не могут из-за ограничений.

Результаты сравнения алгоритмов детализированы в трех таблицах: 3, 4 и 5, где также приведено разбиение на группы примеров. В каждой таблице столбцы содержат усреднённые значения

Table 4. Algorithms with MultiStart

Таблица 4. Алгоритмы с процедурой MultiStart

Группа примеров	Classic MultiStart		Dense MultiStart		Mean MultiStart		Dense Mean MultiStart	
	Best	$K=0$	Best	$K=0$	Best	$K=0$	Best	$K=0$
LPO_s_2	6.36 %	7.21 %	6.09 %	7.20 %	6.41 %	7.81 %	6.64 %	7.79 %
LPO_s_4	10.25 %	11.29 %	9.32 %	9.34 %	10.42 %	11.13 %	9.32 %	9.34 %
LPO_m_2	5.75 %	6.46 %	5.95 %	6.53 %	5.75 %	6.46 %	5.95 %	6.53 %
LPO_m_4	5.49 %	5.74 %	5.21 %	5.55 %	5.40 %	5.74 %	5.21 %	5.55 %
LPO_l_2	11.05 %	11.71 %	11.14 %	11.78 %	11.05 %	11.71 %	11.14 %	11.78 %
LPO_l_4	7.03 %	7.18 %	6.88 %	7.33 %	7.03 %	7.18 %	6.88 %	7.33 %
MP_s_2	6.52 %	7.56 %	6.38 %	6.73 %	6.48 %	7.56 %	6.38 %	6.73 %
MP_s_4	10.19 %	10.15 %	10.08 %	10.43 %	10.70 %	10.51 %	10.08 %	10.43 %
MP_m_2	5.85 %	6.57 %	6.10 %	7.32 %	5.85 %	6.57 %	6.10 %	7.32 %
MP_m_4	4.87 %	5.25 %	5.54 %	5.64 %	4.99 %	5.37 %	5.54 %	5.64 %
MP_l_2	8.92 %	9.58 %	9.01 %	9.52 %	8.92 %	9.58 %	9.01 %	9.52 %
MP_l_4	4.63 %	4.94 %	4.71 %	5.10 %	4.59 %	4.90 %	4.71 %	5.10 %
DMP_s_2	4.09 %	4.45 %	4.24 %	5.32 %	4.09 %	4.45 %	4.24 %	5.32 %
DMP_s_4	2.62 %	2.85 %	2.45 %	2.68 %	2.62 %	2.93 %	2.45 %	2.68 %
DMP_m_2	6.63 %	7.17 %	6.62 %	7.28 %	6.63 %	7.17 %	6.62 %	7.28 %
DMP_m_4	1.76 %	2.04 %	1.88 %	2.21 %	1.76 %	2.04 %	1.88 %	2.21 %
DMP_l_2	7.83 %	8.19 %	7.79 %	8.49 %	7.83 %	8.19 %	7.79 %	8.49 %
DMP_l_4	1.97 %	2.27 %	1.88 %	2.31 %	1.97 %	2.27 %	1.88 %	2.31 %
GAP_{Mean}	6.21 %	6.70 %	6.18 %	6.71 %	6.25 %	6.75 %	6.21 %	6.74 %

GAP по всем примерам из соответствующей группы. Кроме того, в последней строке каждой таблицы представлены значения GAP , усреднённые по всем примерам, что позволяет оценить общую эффективность алгоритмов. Таблица 3 содержит результаты алгоритмов, которые не применяют генерацию столбцов с учётом конфликтов, в то время как таблица 4 отражает результаты тех же алгоритмов с применением MultiStart. Данная процедура позволяет улучшить результат в среднем на 0.78 %.

В результатах всех алгоритмов наблюдается схожая зависимость от групп примеров. Влияние числа NUMA-узлов на результаты может быть обусловлено устройством генератора и требует более глубокого исследования. При фиксированном K наилучший результат $GAP_{Mean} = 6.70\%$ показал алгоритм Classic MultiStart, но при условии настройки параметра для каждого примера отдельно выиграл алгоритм Dense MultiStart с $GAP_{Mean} = 6.18\%$. Согласно данным таблицы 5, алгоритмы, использующие конфликтующие шаблоны, показали относительно плохие результаты. Отчасти такие результаты связаны с тем, что в генерации шаблонов участвовали всего 2 группы, тогда как в самых маленьких примерах число групп с конфликтами достигает 100. Тем не менее ранее был проведён эксперимент на примерах сильно меньшей размерности, где существовало в среднем 12 моментов времени, 20 групп с подгруппами и 6000 ВМ. Такой размер эксперимента позволил задействовать все группы в генерации шаблонов, однако средний результат алгоритмов с такими шаблонами также значительно проигрывал аналогам. Подобный результат может свидетельствовать о непригодности шаблонов с конфликтами для данного типа алгоритмов.

В таблице 6 указаны максимальное, минимальное и среднее время работы каждого алгоритма в секундах на всём множестве примеров. Алгоритмы, использующие генерацию столбцов с конфликтами, уступили аналогам без неё как в плане GAP , так и по времени работы. Дольше всех в среднем решаются примеры из семейства LPO, так как при больших группах партиций значительно увеличивается число конфликтов. Примеры из семейства DMP, напротив, решаются быстрее

Table 5. Algorithms based on the column generation with conflicts**Таблица 5.** Алгоритмы, основанные на генерации столбцов с конфликтами

Группа примеров	CGC		CGC MultiStart		Dense CGC		Dense CGC MultiStart	
	Best	$K=0$	Best	$K=0$	Best	$K=0$	Best	$K=0$
LPO_s_2	10.29 %	11.10 %	9.18 %	9.79 %	9.99 %	11.03 %	9.11 %	10.26 %
LPO_s_4	12.83 %	13.32 %	12.19 %	12.85 %	13.44 %	13.84 %	12.75 %	12.84 %
LPO_m_2	6.93 %	7.61 %	6.30 %	6.99 %	7.45 %	7.91 %	6.50 %	7.19 %
LPO_m_4	7.12 %	7.13 %	6.67 %	6.50 %	7.66 %	7.90 %	7.15 %	7.48 %
LPO_l_2	12.79 %	13.23 %	11.38 %	11.83 %	12.76 %	13.17 %	11.26 %	11.63 %
LPO_l_4	8.58 %	8.88 %	7.45 %	7.74 %	8.73 %	8.87 %	7.37 %	7.70 %
MP_s_2	9.81 %	10.53 %	8.88 %	9.59 %	10.42 %	11.63 %	9.38 %	10.22 %
MP_s_4	14.40 %	15.19 %	14.08 %	14.86 %	14.91 %	15.19 %	14.91 %	15.19 %
MP_m_2	8.05 %	8.66 %	6.81 %	7.56 %	8.20 %	8.69 %	6.90 %	7.72 %
MP_m_4	8.08 %	8.19 %	7.91 %	8.11 %	9.62 %	9.45 %	9.45 %	9.36 %
MP_l_2	10.59 %	10.94 %	9.01 %	9.52 %	10.47 %	10.72 %	8.98 %	9.51 %
MP_l_4	5.65 %	5.85 %	4.91 %	5.06 %	5.65 %	5.84 %	5.03 %	5.22 %
DMP_s_2	4.80 %	5.17 %	4.50 %	4.93 %	5.09 %	5.32 %	4.79 %	5.13 %
DMP_s_4	4.33 %	4.22 %	4.33 %	4.22 %	5.39 %	5.53 %	5.39 %	5.53 %
DMP_m_2	7.51 %	7.67 %	7.05 %	7.29 %	7.56 %	7.86 %	7.25 %	7.41 %
DMP_m_4	3.59 %	3.49 %	3.59 %	3.49 %	4.65 %	4.91 %	4.65 %	4.91 %
DMP_l_2	8.94 %	9.38 %	7.86 %	8.48 %	8.96 %	9.35 %	7.89 %	8.38 %
DMP_l_4	3.88 %	3.91 %	3.83 %	3.87 %	4.43 %	4.67 %	4.35 %	4.59 %
GAP_{Mean}	8.23 %	8.58 %	7.55 %	7.93 %	8.63 %	8.99 %	7.95 %	8.35 %

Table 6. Running time**Таблица 6.** Время работы алгоритмов

Алгоритм	Максимальное время	Минимальное время	Среднее время
Classic	346	3	105
Classic Iter	3047	3	330
Dense	344	3	107
Dense Iter	3019	3	330
Mean	558	3	118
Mean Iter	4727	3	367
Dense Mean	373	3	111
Dense Mean Iter	3252	3	349
CGC	3478	10	969
CGC iter	13476	13	1919
Dense CGC	6716	9	996
Dense CGC iter	15141	9	1941

всех из-за большого числа неконфликтующих ВМ. Примеры с 2 NUMA-узлами решаются в среднем по времени приблизительно одинаково для всех алгоритмов. Однако на примерах с 4 NUMA-узлами алгоритмы с CGC могут работать в среднем в 27 раз медленнее, чем на примерах с 2 NUMA-узлами, тогда как остальные алгоритмы замедляются лишь в 1.2–2 раза. Замедление происходит из-за роста числа переменных и ограничений в модели (12)–(17). Из-за наличия расширенных типов в модели (12)–(20) число переменных и ограничений растёт с увеличением количества NUMA-узлов в несколько раз быстрее.

Заключение

В настоящем исследовании была подробно рассмотрена динамическая задача упаковки виртуальных машин в серверные стойки с учетом групповых размещений и конфликтов между ними. Мы разработали математическую модель для данной NP-трудной задачи. Для её решения была предложена эвристическая процедура, основанная на методе генерации столбцов и распространении статического решения. Мы также провели анализ нескольких эвристик и сравнили их по времени выполнения и качеству работы на обширном бенчмарке. Кроме того, мы изучили влияние параметров на эффективность работы разработанных алгоритмов. Предложенные методы и модели могут быть обобщены на другие варианты задачи упаковки с группами размещения.

В дальнейших работах планируется более тщательное исследование предложенных алгоритмов. Например, планируется протестировать разделение групп целиком на большие и маленькие, вместо разделения подгрупп. Также интересно было бы адаптировать алгоритмы для шаблонов с конфликтами, т. к. пока что они показывают относительно плохой результат.

References

- [1] Z. Farahmandpour, M. Seyedmahmoudian, and A. Stojcevski, “A review on the service virtualisation and its structural pillars”, *Applied Sciences*, vol. 11, no. 5, p. 2381, 2021.
- [2] M. de Cauwer, D. Mehta, and B. O’Sullivan, “The temporal bin packing problem: An application to workload management in data centres”, in *IEEE 28th International Conference on Tools with Artificial Intelligence*, 2016, pp. 157–164.
- [3] M. Dell’Amico, F. Furini, and M. Iori, “A branch-and-price algorithm for the temporal bin packing problem”, *Computers & Operations Research*, vol. 114, p. 104825, 2019.
- [4] A. Ratushnyi and Y. Kochetov, “A column generation based heuristic for a temporal bin packing problem”, in *Mathematical Optimization Theory and Operations Research*, 2021, pp. 96–110.
- [5] M. Sakhno, “A grouping genetic algorithm for the temporal vector bin packing problem”, in *19th International Asian School-Seminar on Optimization Problems of Complex Systems*, 2023, pp. 94–99.
- [6] D. Lazarev and N. Kuzyurin, “On online algorithms for bin, strip, and box packing, and their worst-case and average-case analysis”, *Programming and Computer Software*, vol. 45, no. 4, pp. 448–457, 2020, in Russian.
- [7] V. Toporkov, D. Yemelyanov, and A. Bulkhak, “Machine learning-based online scheduling in distributed computing”, in *Parallel Processing and Applied Mathematics*, 2023, pp. 248–259.
- [8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Sandpiper: Black-box and gray-box resource management for virtual machines”, *Computer Networks*, vol. 53, no. 17, pp. 2923–2938, 2009.
- [9] V. Ivannikov, D. Grushin, N. Kuzyurin, A. Pospelov, and S. A., “Software for improving the energy efficiency of a computer cluster”, *Programming and Computer Software*, vol. 36, pp. 327–336, 2010.
- [10] F. Furini and X. Shen, “Matheuristics for the temporal bin packing problem”, in *Recent Developments in Metaheuristics*. 2018, pp. 333–345.
- [11] D. Grushin and N. Kuzyurin, “On effective scheduling in computing clusters”, *Programming and Computer Software*, vol. 45, pp. 398–404, 2019.
- [12] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing”, *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.
- [13] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, and G. Jiang, “Power and performance management of virtualized computing environments via lookahead control”, *Cluster Computing*, vol. 12, pp. 1–15, 2009.

- [14] A. Verma, P. Ahuja, and A. Neogi, “pMapper: Power and migration cost aware application placement in virtualized systems”, in *Middleware*, 2008, pp. 243–264.
- [15] M. Mao and M. Humphrey, “A performance study on the VM startup time in the cloud”, in *IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 423–430.
- [16] A. Ratushnyi, “A pattern-based heuristic for a temporal bin packing problem with conflicts”, in *Mathematical Optimization Theory and Operations Research: Recent Trends*, 2023, pp. 161–175.
- [17] C. Lameter, “NUMA (non-uniform memory access): An overview”, *Queue*, vol. 11, no. 7, pp. 40–51, 2013.
- [18] *Placement groups for your Amazon EC2 instances*. [Online]. Available: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>.
- [19] *Temporal bin packing problem with placement groups*. [Online]. Available: <http://old.math.nsc.ru/AP/benchmarks/Temporal%20Bin%20Packing/binpack.html>.
- [20] *SCIP. Solving constraint integer programs*. [Online]. Available: <https://www.scipopt.org/>.

Приложение. Псевдокод используемых алгоритмов

Sort. Sorting a list of objects Input: Список объектов <i>List</i>, список ключей <i>Keys</i>, флаг порядка сортировки <i>IsDescendingOrder</i>; Output: Отсортированный список объектов; if <i>isDescendingOrder</i> = <i>false</i> then Сортируем <i>List</i> по возрастанию соответствующих ключей из списка <i>Keys</i>; end else Сортируем <i>List</i> по убыванию соответствующих ключей из списка <i>Keys</i>; end return <i>List</i>	Sort. Сортировка списка объектов Input: Упорядоченный список предметов <i>Items</i>, упорядоченный список контейнеров <i>Containers</i>, флаг возможности добавлять новые стойки <i>ExpansionAllowed</i>; for every <i>item</i> ∈ <i>Items</i> do <i>IsUnplaced</i> ← <i>item</i> не упакован; for every <i>container</i> ∈ <i>Containers</i> do <i>ItemFits</i> ← <i>item</i> помещается в <i>container</i>; <i>NoPlacementConflict</i> ← не нарушается ограничение на группы размещений; if <i>IsUnplaced</i> = <i>true</i> and <i>ItemFits</i> = <i>true</i> and <i>NoPlacementConflict</i> = <i>true</i> then Пакуем <i>item</i> в <i>container</i>; // если <i>container</i> – стойка, а <i>item</i> – ВМ, то пакуем в первый подходящий сервер <i>IsUnplaced</i> ← <i>false</i>; end end if <i>IsUnplaced</i> = <i>true</i> and <i>ExpansionAllowed</i> = <i>true</i> then <i>f</i> ← новая стойка, заполненная пустыми серверами; <i>F.append(f)</i>; // добавляем в множество стоек Пакуем <i>item</i> в <i>f</i>; end end
---	---

SelectServers. Procedure for selecting suitable servers

SelectServers. Процедура выделения подходящих серверов

Input: Список виртуальных машин VM , список серверов $Servers$

Output: Список выделенных серверов

Добавляем в (9)-(11) ограничение на целочисленность переменных;

Добавляем в (9)-(11) ограничение, что можно использовать только сервера из $Servers$;

Определяем матрицу (a_{lj}) в соответствии с шаблонами серверов из $Servers$;

Определяем числа n_l в соответствии с типами ВМ из списка VM ;

$SelectedServers \leftarrow$ получаем подходящие неразмещённые серверы из решения модели;

return $SelectedServers$

Algorithm 1. Static stage of the basic algorithm

Algorithm 1. Статический этап базового алгоритма

Input: множество виртуальных машин M_t , которые существуют в момент $t \in T$, параметр

$K \geq 0$

$Servers, LB \leftarrow$ применяем метод генерации столбцов без конфликтов для множества M_t ;

// получили список серверов с шаблонами и нижнюю границу на число серверов

$LB \leftarrow \lceil LB/RackCapacity \rceil$;

// делим на вместимость одной стойки по серверам

и получаем нижнюю границу на число стоек

Создаём LB пустых стоек без серверов;

for every $g \in G$ **do**

for every $p \in P_g$ **do**

$w_{gp} \leftarrow$ суммарная нагрузка всех ВМ $m \in M_t$, которые принадлежат подгруппе p ;

end

$w_g \leftarrow \sum_{p \in P_g} w_{gp}$;

end

$G^{sorted} \leftarrow \text{Sort}(G, (w_g)_{g \in G}, \text{true})$; // группы с нулевой суммарной нагрузкой в момент t не учитываются

for every $g \in G$ **do**

$P_g^{sorted} \leftarrow \text{Sort}(P_g, (w_{gp})_{g \in G, p \in P_g}, \text{true})$;

end

$P_1, P_2 \leftarrow \text{SplitSubgroups}(G, (P_g)_{g \in G}, K)$; // Разделяем подгруппы в зависимости от K : если для упаковки подгруппы требуется $> K$ серверов, то подгруппа добавляется в P_1 , иначе в P_2

$\text{Algorithm1}(G^{sorted}, (P_g^{sorted})_{g \in G}, P_1)$;

// упаковка подгрупп из P_1

Пакуем подгруппы из P_2 с помощью решения задачи (21)-(25);

$Unconstrained \leftarrow$ свободные ВМ;

$\text{FirstFit}(Unconstrained, F, \text{false})$;

// F - множество стоек

SortGroups. Sorting groups by number of VMs

SortGroups. Сортировка групп по числу ВМ

Input: множество групп $\tilde{G}$, множество виртуальных машин $\tilde{M}$

for every $g \in \tilde{G}$ **do**

$h_g \leftarrow$ кол-во ВМ из $\tilde{M}$ в этой группе g ;

end

$\text{Sort}(\tilde{G}, (h_g)_{g \in \tilde{G}}, \text{true})$;

// сортировка по убыванию числа ВМ

Algorithm 2. The procedure for packing partitions
from the set P_1

Algorithm 2. Процедура упаковки подгрупп
из множества P_1

Input: Отсортированный по убыванию суммарной нагрузки ВМ список групп G^{sorted} ,
отсортированные по убыванию суммарной нагрузки ВМ списки подгрупп P_g^{sorted} ,
множество подгрупп P_1 ;

```

for every  $g \in G^{sorted}$  do
  for every  $p \in P_g^{sorted}$  do
    if  $p \in P_1$  then
      Выделяем множество активных стоек  $F_p$ ;    // на которых размещена хотя бы
      одна ВМ
      for every  $p' \in P_g^{sorted}$  do
        if  $p' \neq p$  then
          for every  $f \in F_p$  do
            if  $f$  содержит ВМ из  $p'$  then
              Удаляем  $f$  из  $F_p$ ;    // для соблюдения ограничения на группы
              размещений
            end
          end
        end
      end
      for every  $l \in L$  do
         $k_l \leftarrow$  число ВМ типа  $l$  в подгруппе  $p$ ;
      end
      for every  $f \in F_p$  do
        for every  $l \in L$  do
           $s_{fl} \leftarrow$  число незанятых мест для ВМ типа  $l$  на стойке  $f$ ;
        end
         $m_f \leftarrow \sum_{l \in L} \min(s_{fl}, k_l)$ ;
      end
       $F_p^{sorted} \leftarrow \text{Sort}(F_p, (m_f)_{f \in F_p}, \text{true});$     // По убыванию
      for every  $f \in F_p^{sorted}$  do
         $\text{UnplacedItems} \leftarrow$  неразмещённые ВМ из  $p$ ;
         $\text{FreeSlots} \leftarrow$  незанятые места серверов стойки  $f$ ;
         $\text{FirstFit}(\text{UnplacedItems}, \text{FreeSlots}, \text{false});$ 
      end
      if в подгруппе  $p$  остались неразмещённые ВМ then
        for every  $f \in F_p$  do
           $c_f \leftarrow$  число незанятых для серверов мест стойки  $f$ ;
        end
         $F_p^{sorted} \leftarrow \text{Sort}(F_p, (c_f)_{f \in F_p}, \text{true});$     // По убыванию
         $\text{Servers} \leftarrow$  неразмещённые серверы;
         $\text{SelectedServers} \leftarrow \text{SelectServers}(p, \text{Servers});$ 
        Распределяем оставшиеся ВМ из  $p$  на серверы из  $\text{SelectedServers}$ ;    // Все ВМ
        будут распределены
        for every  $f \in F_p^{sorted}$  do
          Размещаем на стойку  $f$  столько серверов из  $\text{SelectedServers}$ , сколько
          возможно;    // ВМ из неразмещённых серверов будут доупакованы
          на динамическом этапе
        end
      end
    end
  end
end

```

PackGroups. Packing a list of groups **PackGroups.** Упаковка списка групп

Input: упорядоченный список групп $\tilde{G}$, упорядоченный список виртуальных машин $\tilde{M}$,
флаг упаковки свободных виртуальных машин $PackUnconstrained$, флаг
возможности добавлять новые стойки $ExpansionAllowed$;

```

for every  $g \in \tilde{G}$  do
  for every  $m \in \tilde{M}$  do
    if  $m$  содержится в  $g$  then
       $FirstFit((m), F, ExpansionAllowed)$ ;    // сначала учитывается порядок групп,
      потом порядок ВМ; если ВМ уже была упакована, это учитывается внутри
      процедуры  $FirstFit$ 
    end
  end
end
if  $PackUnconstrained = true$  then
  for every  $m \in \tilde{M}$  do
    if  $m$  свободная ВМ then
       $FirstFit((m), F, ExpansionAllowed)$ ;
    end
  end
end

```

Algorithm 3. Dynamic stage of the basic algorithm

Algorithm 3. Динамический этап базового алгоритма

Input: разделённые по моменту $t \in T$ множества виртуальных машин M_t^{before} , M_t , M_t^{after} ;
 $G_t \leftarrow$ группы, в которых есть ВМ из M_t ;
 $G_t^{before} \leftarrow$ группы, в которых есть ВМ из M_t^{before} ;
 $G_t^{after} \leftarrow$ группы, в которых есть ВМ из M_t^{after} ;
 $SortGroups(G_t, M_t)$;
 $SortGroups(G_t^{before}, M_t^{before})$;
 $SortGroups(G_t^{after}, M_t^{after})$;
 $Sort(M_t^{after}, (\alpha_m)_{m \in M_t^{after}}, false)$;
 $Sort(M_t^{before}, (\omega_m)_{m \in M_t^{before}}, true)$;
 $PackGroups(G_t^{after}, M_t^{after}, false, false)$; // сохранение шаблонов
 $PackGroups(G_t^{before}, M_t^{before}, false, false)$; // сохранение шаблонов
Возвращаемся к изначальной структуре серверов без шаблонов;
 $PackGroups(G_t, M_t, false, true)$; // отказ от шаблонов
 $PackGroups(G_t^{after}, M_t^{after}, false, true)$; // отказ от шаблонов
 $PackGroups(G_t^{before}, M_t^{before}, false, true)$; // отказ от шаблонов
 $PackGroups(G_t, M_t, true, true)$; // упаковка свободных ВМ
 $PackGroups(G_t^{after}, M_t^{after}, true, true)$; // упаковка свободных ВМ
 $PackGroups(G_t^{before}, M_t^{before}, true, true)$; // упаковка свободных ВМ

Algorithm 4. Alternative packing procedure for P_1 **Algorithm 4.** Альтернативная процедура упаковки для P_1

Input: Отсортированный по убыванию суммарной нагрузки ВМ список групп G^{sorted} ,
 отсортированные по убыванию суммарной нагрузки ВМ списки подгрупп P_g^{sorted} ,
 множество подгрупп P_1 ;

```

for every  $g \in G^{sorted}$  do
  for every  $p \in P_g^{sorted}$  do
    if  $p \in P_1$  then
      Выделяем множество активных стоек  $F_p$ ;    // на которых размещена хотя бы
      одна ВМ
      for every  $p' \in P_g^{sorted}$  do
        if  $p' \neq p$  then
          for every  $f \in F_p$  do
            if  $f$  содержит ВМ из  $p'$  then
              Удаляем  $f$  из  $F_p$ ;    // для соблюдения ограничения на группы
              размещений
            end
          end
        end
      end
      for every  $l \in L$  do
         $k_l \leftarrow$  число ВМ типа  $l$  в подгруппе  $p$ ;
      end
      for every  $f \in F_p$  do
        for every  $l \in L$  do
           $s_{fl} \leftarrow$  число незанятых мест для ВМ типа  $l$  на стойке  $f$ ;
        end
         $m_f \leftarrow \sum_{l \in L} \min(s_{fl}, k_l)$ ;
      end
       $F_p^{sorted} \leftarrow \text{Sort}(F_p, (m_f)_{f \in F_p}, \text{true});$     // По убыванию
      for every  $f \in F_p^{sorted}$  do
         $UnplacedItems \leftarrow$  неразмещённые ВМ из  $p$ ;
         $FreeSlots \leftarrow$  незанятые места серверов стойки  $f$ ;
         $\text{FirstFit}(UnplacedItems, FreeSlots, \text{false});$ 
        if в подгруппе  $p$  остались неразмещённые ВМ then
           $Servers \leftarrow$  неразмещённые серверы;
           $SelectedServers \leftarrow \text{SelectServers}(p, Servers);$ 
          Распределяем оставшиеся ВМ из  $p$  на серверы из  $SelectedServers$ ;    // Все
          ВМ будут распределены
          Размещаем на стойку  $f$  столько серверов из  $SelectedServers$ , сколько
          возможно;    // ВМ из неразмещённых серверов будут записаны
          в  $UnplacedItems$  на следующей итерации
        end
      end
    end
  end
end

```

CalculateActiveRacks. Calculating the number of active racks

CalculateActiveRacks. Вычисление числа активных стоек

Input: данные об упаковке всех виртуальных машин $Solution$;

Output: Количество активных стоек в каждый момент времени (список) $(r_t)_{t \in T}$;

for every $t \in T$ **do**

$r_t \leftarrow 0$;

for every $f \in F$ **do**

if B $Solution$ есть ВМ, которая существует в момент t и упакована на стойке f **then**

$r_t \leftarrow r_t + 1$;

end

end

end

return $(r_t)_{t \in T}$

Algorithm 5. MultiStart procedure

Algorithm 5. Процедура MultiStart

Input: множество виртуальных машин M , параметр $K \geq 0$, вариация алгоритма получения верхней оценки $Algorithm$;

Output: верхняя оценка на число стоек, допустимое решение задачи

$\hat{t} \leftarrow \arg \max_{t \in T} (\max_{r \in R} \sum_{m \in M_t} b_{mr})$; // вычисляем момент максимальной нагрузки

$prevMoments \leftarrow \emptyset$;

// список предыдущих моментов

$minValue \leftarrow -\infty$;

$bestSolution \leftarrow \emptyset$;

while $\hat{t} \notin prevMoments$ **do**

$value, solution \leftarrow Algorithm(M, \hat{t}, K)$;

if $value > minValue$ **then**

$minValue \leftarrow value$;

$bestSolution \leftarrow solution$;

end

$(r_t)_{t \in T} \leftarrow CalculateActiveRacks(solution)$;

if $r_{\hat{t}}$ - единственный максимум в $(r_t)_{t \in T}$ или принадлежит единственному плато максимумов **then**

break;

// сработал один из критериев остановки

end

$prevMoments.Append(\hat{t})$;

$\hat{t} \leftarrow$ ближайший к $\hat{t}$ момент времени, в котором достигается максимум по числу активных стоек из $(r_t)_{t \in T}$;

end

return $minValue, bestSolution$

Algorithm 6. Basic algorithm for getting the upper bound

Algorithm 6. Базовый алгоритм получения верхней оценки

Input: множество виртуальных машин M , момент времени $t \in T$, параметр $K \geq 0$

Output: верхняя оценка на число стоек, допустимое решение задачи

$M_t \leftarrow \{m \in M | \alpha_m \leq t < \omega_m\}$;

$M_t^{before} \leftarrow \{m \in M | \omega_m \leq t\}$;

$M_t^{after} \leftarrow \{m \in M | \alpha_m > t\}$;

$Algorithm2(M_t, K)$;

// статический этап

$Algorithm3(M_t^{before}, M_t, M_t^{after})$;

// динамический этап

$Placements \leftarrow$ информация об упаковке всех ВМ из M ;

return $|F|, Placements$

Exact Algorithm for the Problem of the Minimum Complete Spanning Tree of a Divisible Multiple Graph

A. V. Smirnov¹DOI: [10.18255/1818-1015-2025-2-132-149](https://doi.org/10.18255/1818-1015-2025-2-132-149)¹P.G. Demidov Yaroslavl State University, Yaroslavl, Russia

MSC2020: 05C85, 05C65, 05C05

Research article

Full text in Russian

Received April 23, 2025

Revised May 27, 2025

Accepted May 28, 2025

We study undirected multiple graphs of any natural multiplicity $k > 1$. There are edges of three types: ordinary edges, multiple edges, and multi-edges. Each edge of the last two types is a union of k linked edges, which connect 2 or $(k + 1)$ vertices, correspondingly. The linked edges should be used simultaneously. Divisible graphs form a special class of multiple graphs. The main peculiarity of them is a possibility to divide the graph into k parts, which are adjusted on the linked edges and which have no common edges. Each part is an ordinary graph.

The multiple tree is a multiple graph with no multiple cycles. The number of edges may be different for multiple trees with the same number of vertices. Also we can consider spanning trees of a multiple graph. A spanning tree is complete if a multiple path joining any two selected vertices exists in the tree if and only if such a path exists in the initial graph. The problem of the minimum complete spanning tree of a multiple graph is NP-hard even in the case of a divisible graph. In this article, we obtain an exact algorithm for the problem of the minimum complete spanning tree of a divisible multiple graph. Also we define a subclass of divisible graphs, for which the algorithm runs in polynomial time.

Keywords: multiple graph; divisible graph; multiple tree; complete spanning tree; exact algorithm

INFORMATION ABOUT THE AUTHORS

Smirnov, Alexander V. (corresponding author)	ORCID iD: 0000-0002-0980-2507 . E-mail: alexander.sm@mail.ru PhD, Associate Professor, Department of Theoretical Computer Science
---	--

Funding: Yaroslavl State University (project VIP-016).

For citation: A. V. Smirnov, "Exact algorithm for the problem of the minimum complete spanning tree of a divisible multiple graph", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 132–149, 2025. DOI: [10.18255/1818-1015-2025-2-132-149](https://doi.org/10.18255/1818-1015-2025-2-132-149).

Точный алгоритм для задачи о минимальном полном остовном дереве в делимом кратном графе

А. В. Смирнов¹

DOI: [10.18255/1818-1015-2025-2-132-149](https://doi.org/10.18255/1818-1015-2025-2-132-149)

¹Ярославский государственный университет им. П.Г. Демидова, Ярославль, Россия

УДК 519.17+519.161

Научная статья

Полный текст на русском языке

Получена 23 апреля 2025 г.

После доработки 27 мая 2025 г.

Принята к публикации 28 мая 2025 г.

В статье рассматриваются неориентированные кратные графы произвольной натуральной кратности $k > 1$. Кратный граф содержит ребра трех типов: обычные, кратные и мультиребра. Ребра последних двух типов представляют собой объединение k связанных ребер, которые соединяют 2 или $(k + 1)$ вершину соответственно. Связанные ребра могут использоваться только согласованно. Делимые графы представляют собой специальный класс кратных графов. Их основная особенность состоит в возможности разделить граф на k частей, которые будут согласованы на связанных ребрах и не будут иметь общих ребер. Каждая часть является обычным графом.

Кратное дерево представляет собой кратный граф без кратных циклов. Количество ребер может быть разным для кратных деревьев с одинаковым количеством вершин. Также можно рассмотреть остовные деревья в кратном графе. Остовное дерево является полным, если кратный путь, соединяющий любые две выбранные вершины, существует в дереве тогда и только тогда, когда такой путь существует в исходном графе. Задача о минимальном полном остовном дереве в кратном графе NP-трудна даже в случае делимого графа. В данной статье мы получим точный алгоритм для задачи о минимальном полном остовном дереве в делимом кратном графе. Также мы определим подкласс делимых графов, для которых алгоритм будет выполняться за полиномиальное время.

Ключевые слова: кратный граф; делимый граф; кратное дерево; полное остовное дерево; точный алгоритм

ИНФОРМАЦИЯ ОБ АВТОРАХ

Смирнов, Александр Валерьевич
(автор для корреспонденции)

ORCID iD: [0000-0002-0980-2507](https://orcid.org/0000-0002-0980-2507). E-mail: alexander_sm@mail.ru
Канд. физ.-мат. наук, доцент, кафедра теоретической информатики

Финансирование: ЯрГУ (проект VIP-016).

Для цитирования: A. V. Smirnov, "Exact algorithm for the problem of the minimum complete spanning tree of a divisible multiple graph", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 132–149, 2025. DOI: [10.18255/1818-1015-2025-2-132-149](https://doi.org/10.18255/1818-1015-2025-2-132-149).

Введение

В данной статье мы рассмотрим *остовные деревья* в делимом кратном графе. Ранее мы определили кратный граф и кратное дерево произвольной натуральной кратности $k > 1$ (см. [1–3]). Кратные графы содержат три типа ребер (обычные, кратные и мультиребра) и являются обобщением обычных графов — по сути, обычный граф имеет кратность $k = 1$. Делимые графы представляют собой специальный класс кратных графов. Их основная особенность состоит в возможности разделить граф на k частей, которые будут согласованы на связанных ребрах и не будут иметь общих ребер. Каждая часть является обычным графом.

Также мы рассмотрели обобщения для кратных графов ряда классических задач теории графов. В частности, в статье [4] мы доказали что задача о минимальном остовном дереве является NP-трудной как для произвольного, так и для делимого кратного графа, если кратность $k \geq 3$.

Среди других известных обобщений графов наиболее близкими нам концепциями являются мультиграфы, гиперграфы (см., например, [5, 6]), а также метаграфы (см. [7, 8]). Действительно, как и в мультиграфах, в кратных графах допускается наличие нескольких ребер между парой вершин (набор таких ребер мы будем в дальнейшем называть *кратным ребром*), однако в случае кратного графа количество таких ребер должно быть строго равным k . В кратных графах присутствуют *мультиребра*, соединяющие между собой $(k + 1)$ вершину. Но в отличие от гиперребер гиперграфа, мультиребро представляется в виде k связанных ребер, имеющих один общий конец, причем все эти k ребер должны использоваться согласованно, то есть все характеристики ребра должны иметь одинаковые значения для всех связанных ребер мультиребра. Кроме того, мультиребро может включаться в какую-либо новую структуру только как единое целое. По сути, понятие мультиребра близко понятию ребра между вершиной и метавершиной в метаграфе. При этом в метаграфе, напомним, метапуть между двумя метавершинами фактически моделирует причинно-следственные связи в некоторой предметной области. Однако в кратном графе используется принципиально иной подход к определению пути: *кратный путь* должен состоять ровно из k обычных путей, проходящих по обычным ребрам, а также по связанным ребрам кратных и мультиребер; при этом пути должны быть согласованы (одинаковы) на кратных и мультиребрах. Поэтому кратный граф нельзя считать частным случаем метаграфа.

Отметим, что частным случаем кратного графа является кратная сеть (см. [9, 10]), представляющая собой ориентированный кратный граф с одним источником и одним стоком. Задача о наибольшем потоке в кратной сети обобщает классическую задачу (см. [11]) и имеет ряд приложений в сфере экономики, управления, финансов.

В частности, кратные сети и потоки используются для поиска решения NP-трудной задачи целочисленного сбалансирования трех- и четырехмерной матрицы (см., например, [12, 13]), которая возникает при планировании железнодорожных грузоперевозок. Имеется матричный план по отправке вагонов, который группируется по некоторым показателям (например, направление, тип вагона, владелец вагона и т.п.). Данный план составляется на месяц и естественно является целочисленным. Однако вагоны необходимо отправлять ежедневно. При делении на количество дней в месяце план перестает быть целочисленным. Поэтому возникает проблема такого округления основных параметров, чтобы суммирующие показатели не выходили за определенные рамки. Данный план может быть представлен в виде s -мерной матрицы, где s — это число показателей, по которым ведется суммирование. Задача целочисленного сбалансирования может быть сведена к задаче о наибольшем кратном потоке (кратность сети равна 2 для трехмерной матрицы и 5 — для четырехмерной).

В разделе 1 мы дадим необходимые определения относительно кратных графов и деревьев.

В разделе 2 мы поставим задачи о минимальном остовном дереве и о минимальном полном остовном дереве. Затем мы сформулируем и обсудим точный алгоритм для второй задачи для случая делимого кратного графа.

1. Кратные графы и кратные деревья: определения

Напомним некоторые определения относительно кратных графов и деревьев, которые были ранее сформулированы в статьях [1–3].

Определение 1. *Кратный граф* G произвольной натуральной кратности $k > 1$ — это граф, вершины которого могут соединяться ребрами одного из 3 видов:

1. *Обычное ребро* e^o ; множество обычных ребер обозначим через E^o .
2. *Кратное ребро* e^k между двумя вершинами, которое состоит из k одинаковых связанных ребер; связанные ребра кратного ребра могут использоваться только согласованно; множество кратных ребер обозначим через E^k .
3. *Связанное ребро* e между двумя вершинами, имеющее один общий конец с другим $(k - 1)$ ребром (у любых двух из k связанных ребер только один конец является общим); множество связанных общей вершиной ребер будем называть *мультиребром* e^m ; связанные ребра мультиребра могут использоваться только согласованно; множество мультиребер обозначим через E^m .

Если вершина инцидентна какому-либо кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом какого-либо мультиребра.

Если вершина является общим концом какого-либо мультиребра, то она не может быть общим концом никакого другого мультиребра.

Если вершина является отдельным концом мультиребра или инцидентна обычному ребру, то она не может быть общим концом мультиребра и не может быть инцидентна кратному ребру.

Множества вершин и ребер графа G обозначим через V и E соответственно ($E = E^o \cup E^k \cup E^m$). Обычное или кратное ребро, соединяющее две вершины x и y , обозначается стандартным образом: $\{x, y\}$. Мультиребро, соединяющее общую вершину x с k отдельными вершинами $y_1, \dots, y_k$, обозначается так: $e_x^m = \{x, \{y_1, \dots, y_k\}\}$. Далее мы будем рассматривать только неориентированные кратные графы.

Рис. 1 иллюстрирует определение 1. На рис. 1 (а) кратное ребро представлено в виде объединения k одинаковых ребер между двумя вершинами, что показано штрихами. Равенство (или согласованность) связанных ребер предполагает, что все характеристики этих ребер (например, длина) одинаковы, и эти ребра могут использоваться только одновременно. Так, если осуществляется проход в определенном направлении по одному из связанных ребер, то одновременно с этим оставшееся $(k - 1)$ ребро проходит в том же самом направлении. Кратное ребро может включаться в какие-либо новые структуры только целиком. В дальнейшем мы будем обозначать кратные ребра жирными линиями, как на рис. 1 (b).

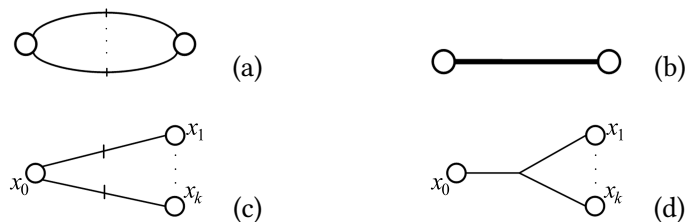


Fig. 1. Multiple and multi-edge

Рис. 1. Кратное и мультиребро

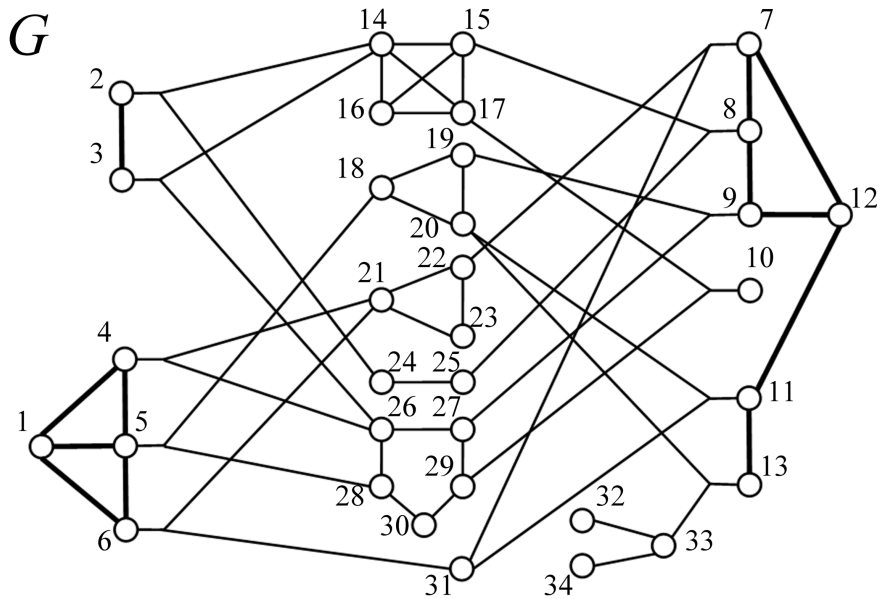


Fig. 2. Divisible graph of multiplicity 2

Рис. 2. Делимый граф кратности 2

На рис. 1 (с) мультиребро $\{x_0, \{x_1, \dots, x_k\}\}$ представлено в виде объединения k одинаковых ребер, связывающих общую вершину x_0 с k попарно различными вершинами $x_1, \dots, x_k$. Как и на рис. 1 (а), равенство ребер показано штрихами. Согласованность связанных ребер имеет тот же смысл, что и для кратных ребер. Если мы проходим связанное ребро $\{x_0, x_i\}$ от или к общему концу x_0 , мы должны одновременно проходить все остальные связанные ребра в том же направлении. В дальнейшем мультиребра мы будем изображать при помощи расщепляющихся на k частей линий, как на рис. 1 (d).

Определение 2. Обычной вершиной назовем вершину, которая инцидентна обычному ребру или является отдельным концом мультиребра.

Кратной вершиной назовем вершину, которая инцидентна кратному ребру или является общим концом мультиребра.

Из определения 1 следует, что множества обычных и кратных вершин не пересекаются. При этом кратная вершина может быть соединена с обычными только посредством мультиребра.

Определение 3. Делимым кратным графом назовем такой кратный граф $G(V, E)$, в котором все связанные ребра кратных и мультиребер можно пронумеровать от 1 до k таким образом, что граф представляется в виде объединения $G = \cup_{i=1}^k G_i$. При этом каждый подграф G_i ($i \in \overline{1, k}$) содержит связанные ребра с номером i всех кратных и мультиребер, а также компоненты связности, состоящие из обычных ребер и инцидентные i -ым связанным ребрам всех мультиребер.

Подграфы G_i будем называть частями делимого графа.

При удалении всех мультиребер делимый граф распадется на $n \geq 1$ компонент связности (связность здесь понимается в том же смысле, что и для обычных графов), каждая из которых содержит только кратные ребра либо только обычные ребра. Очевидно, что каждая часть G_i является обычным графом. При этом возможность выделения частей G_i является особенностью делимых графов. В общем случае получить части G_i не удастся.

Пример 1. Рассмотрим делимый кратный граф G кратности 2, показанный на рис. 2. Для лучшей читаемости рисунка будем обозначать вершины их номерами без буквы «х».

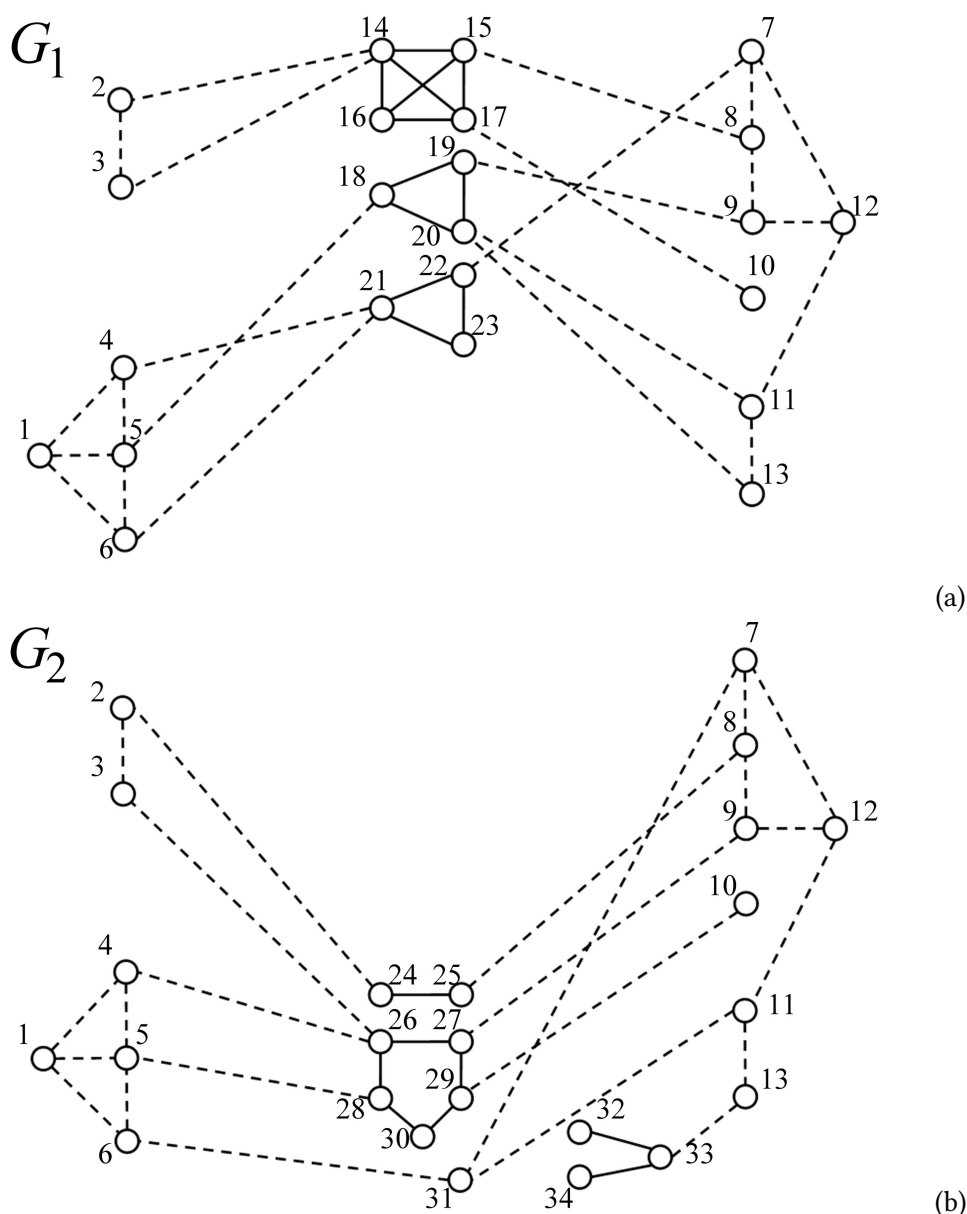


Fig. 3. Partition of a divisible graph

Рис. 3. Части делимого графа

На рис. 3 показаны части G_1 и G_2 делимого кратного графа G . Все связанные ребра обозначены пунктирными линиями.

Часть G_1 этого графа (рис. 3 (a)) состоит из всех обычных вершин с номерами от 14 до 23, всех соединяющих их обычных ребер, всех связанных ребер инцидентных им мультиребер, а также из всех кратных вершин и первых связанных ребер каждого мультиребра.

Часть G_2 (рис. 3 (b)) состоит из всех обычных вершин с номерами от 24 до 34, всех соединяющих их обычных ребер, всех связанных ребер инцидентных им мультиребер, а также из всех кратных вершин и вторых связанных ребер каждого мультиребра.

Дадим теперь определение кратного пути. Основное отличие кратного пути от пути в обычном графе состоит в том, что связанные ребра каждого кратного и мультиребра должны проходиться в этом пути согласованно.

Определение 4. $S(x, y) = \cup_{i=1}^k S^i(x, y)$ является *кратным путем* из вершины x в вершину y в графе $G(V, E)$, если выполнены следующие условия:

1. $S^i(x, y) = \left(\{x, v_1^i\}, \{v_1^i, v_2^i\}, \dots, \{v_{l_i-1}^i, v_{l_i}^i\}, \{v_{l_i}^i, y\} \right)$, где $l_i \geq 0$, — последовательность ребер, представляющая собой обычный (некратный) путь из x в y , где каждое ребро $\{a, b\}$ является либо обычным ребром в графе $G(V, E)$, либо i -ым связанным ребром кратного или мультиребра. Значения l_i и l_j ($i \neq j$) не согласовываются и могут быть как равными, так и различными. Если в путь $S(x, y)$ не входит ни одного кратного или мультиребра, то $S^2(x, y) = S^3(x, y) = \dots = S^k(x, y) = \emptyset$.
2. Любая обычная вершина может встретиться в $S^i(x, y)$ несколько раз, то есть $S^i(x, y)$ может содержать циклы.
3. Никакая кратная вершина не может встретиться в $S^i(x, y)$ дважды.
4. Любое обычное ребро может встречаться в $S^i(x, y)$ несколько раз, причем направления, в которых оно проходится в разных вхождении, могут не совпадать.
5. Обычное ребро, входящее в $S^i(x, y)$, может также входить в любой $S^j(x, y)$, $j \neq i$.
6. Все пути $S^i(x, y)$ согласованы (одинаковы) на общей части. Это условие означает, что если связанное ребро какого-то кратного или мультиребра входит в некоторый путь $S^i(x, y)$, то остальные связанные ребра должны входить во все $S^j(x, y)$, $j \neq i$ (по одному связанному ребру в каждый $S^j(x, y)$). При этом порядок вхождения всех кратных и мультиребер во все $S^i(x, y)$ одинаков.
Фактически это значит, что если e_1 и e_2 — это два ребра пути $S(x, y)$, каждое из которых либо кратное, либо мультиребро, и в проекции $S^i(x, y)$ связанное ребро из e_1 проходится раньше связанного ребра из e_2 , то во всех остальных проекциях $S^j(x, y)$ связанные ребра из e_2 могут проходиться только после связанных ребер из e_1 .
7. Если $S(x, y)$ содержит мультиребро $\{x_0, \{x_1, \dots, x_k\}\}$, проходимое в направлении от общего конца, то он не может содержать никакого другого мультиребра $\{y_0, \{x_1, \dots, x_k\}\}$, проходимого в том же направлении. Аналогичное условие должно выполняться и в случае движения к общему концу.

Определение 5. Кратный путь $S(x, y)$ является *кратным циклом*, если $x = y$ и $S(x, y) \neq \emptyset$.

Пример 2. Рассмотрим граф G , показанный на рис. 2, и получим следующий кратный путь $S(x_8, x_{10})$:

$$S^1(x_8, x_{10}) = (\{x_8, x_{15}\}, \{x_{15}, x_{16}\}, \underline{\{x_{16}, x_{14}\}}, \underline{\{x_{14}, x_2\}}, \underline{\{x_2, x_3\}}, \underline{\{x_3, x_{14}\}}, \underline{\{x_{14}, x_{16}\}}, \{x_{16}, x_{17}\}, \{x_{17}, x_{10}\}),$$

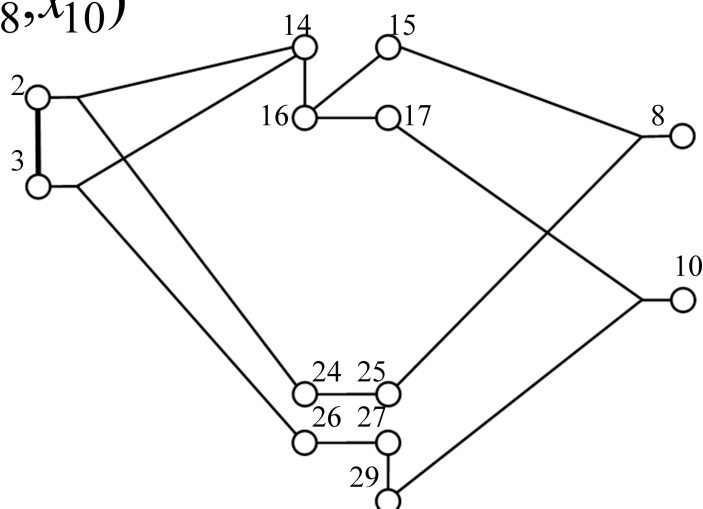
$$S^2(x_8, x_{10}) = (\{x_8, x_{25}\}, \{x_{25}, x_{24}\}, \underline{\{x_{24}, x_2\}}, \underline{\{x_2, x_3\}}, \underline{\{x_3, x_{26}\}}, \{x_{26}, x_{27}\}, \{x_{27}, x_{29}\}, \underline{\{x_{29}, x_{10}\}}).$$

На рис. 4 (а) показан этот кратный путь $S(x_8, x_{10})$. На рис. 4 (б) и (с) показаны части кратного пути $S^1(x_8, x_{10})$ и $S^2(x_8, x_{10})$ соответственно.

Все ребра общей части S^1 и S^2 согласованы (эти ребра отмечены подчеркиванием). Пример демонстрирует особенности кратного пути. Действительно, у нас есть обычное ребро $\{x_{14}, x_{16}\}$ (отмечено двойным подчеркиванием), которое входит в путь S^1 дважды и проходится в противоположных направлениях. Более того, в S^1 есть обычный цикл, однако кратного цикла в пути S нет.

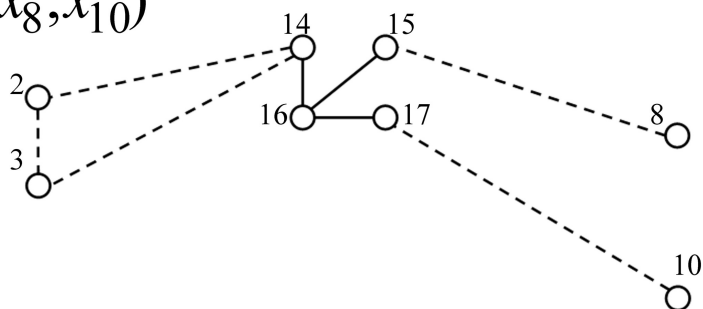
Определение 6. Множеством *достижимости по кратным ребрам* для некоторой кратной вершины x назовем множество R_x^k всех вершин y таких, что существует путь из x в y , проходящий только по кратным ребрам.

$S(x_8, x_{10})$



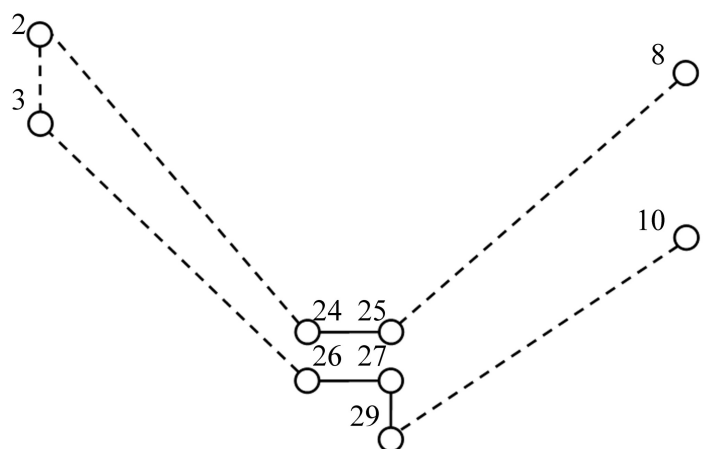
(a)

$S^1(x_8, x_{10})$



(b)

$S^2(x_8, x_{10})$



(c)

Fig. 4. Multiple path and its parts

Рис. 4. Кратный путь и его части

Определение 7. Множеством достижимости по обычным ребрам для некоторой обычной вершины x назовем множество R_x^o всех вершин y таких, что существует путь из x в y , проходящий только по обычным ребрам.

Определение 8. Множества достижимости R_x^k и R_y^k являются смежными, если для произвольных вершин $a \in R_x^k$, $b \in R_y^k$ существует соединяющий их кратный путь $S(a, b)$.

Определение 9. Кратный граф $G(V, E)$ является связным, если одновременно выполнены два условия:

1. Кратный путь $S(x, y)$ существует для любых двух кратных вершин $x \in V$, $y \in V$.
2. Невозможно выделить такой подграф $G' \subset G$, который будет содержать только обычные ребра, и при этом подграфы G' и $G \setminus G'$ не будут соединены ни одним ребром (обычным ребром или связанным ребром мультиребра).

В отличие от обычных графов, связность кратного графа не предполагает наличие кратных путей из каждой вершины в каждую. Фактически в связном кратном графе между каждой парой вершин должен существовать обычный (некратный) путь, использующий связанные ребра кратных и мультиребер несогласованно, а кратные пути обязательно должны существовать только для пар кратных вершин.

Определение 10. Делимый кратный граф $G(V, E)$ является связным, если одновременно выполнены два условия:

1. Кратный путь $S(x, y)$ существует для любых двух кратных вершин $x \in V$, $y \in V$.
2. Каждая из частей G_i является связным (некратным) графом.

Определение 11. Кратное дерево — это связный кратный граф без циклов.

В статьях [2, 3] мы показали, что два кратных дерева с одинаковым количеством кратных и обычных вершин могут содержать разное количество ребер.

Определение 12. Остовным деревом в кратном графе $G(V, E)$ называется кратное дерево $T(V, E')$, для которого $E' \subseteq E$.

Заметим, что в остовном дереве заведомо будут существовать кратные пути $S(x, y)$ для всех кратных вершин x, y . Однако, если хотя бы одна из вершин x, y является обычной, существование пути $S(x, y)$ не гарантировано даже в том случае, когда такой путь существует в исходном графе $G(V, E)$.

Определение 13. Остовное дерево $T(V, E')$ в кратном графе $G(V, E)$ является полным, если для любой пары вершин $x \in V$, $y \in V$ кратный путь $S_T(x, y)$ в дереве $T(V, E')$ существует тогда и только тогда, когда существует кратный путь $S_G(x, y)$ в исходном графе $G(V, E)$.

2. Задача о минимальном полном остовном дереве в делимом кратном графе и алгоритм ее решения

2.1. Постановка задачи

Напомним определение длины ребра из [2, 3].

Определение 14. Целочисленная функция $l(e)$, определенная для всех ребер $e \in E$, является длиной (весом) ребра в кратном графе $G(V, E)$, если выполнено следующее:

1. $l(e) > 0$ для любого ребра e .

2. Если e является кратным или мультиребром, то $l(e_1) = l(e_2) = \dots = l(e_k)$ и $l(e) = k \cdot l(e_1)$, где $e_1, \dots, e_k$ — это связанные ребра данного ребра e .

Тогда вес кратного графа $G(V, E)$ определяется по формуле

$$w(G) = \sum_{e \in E} l(e).$$

Поставим две задачи о минимальном остовном дереве.

Задача 1 (минимальное остовное дерево). В кратном графе $G(V, E)$ требуется найти такое остовное дерево $T^{\min}(V, E')$, что для любого другого остовного дерева $T(V, E'')$ выполнено

$$w(T^{\min}(V, E')) \leq w(T(V, E'')).$$

Задача 2 (минимальное полное остовное дерево). В кратном графе $G(V, E)$ требуется найти такое полное остовное дерево $T_{complete}^{\min}(V, E')$, что для любого другого полного остовного дерева $T_{complete}(V, E'')$ выполнено

$$w(T_{complete}^{\min}(V, E')) \leq w(T_{complete}(V, E'')).$$

Задача 2 является более интересной, поскольку кратные пути существуют в полном остовном дереве для всех пар вершин, для которых соответствующие пути существуют в исходном графе. Кроме того, очевидно, что

$$w(T^{\min}) \leq w(T_{complete}^{\min}).$$

Существование полного остовного дерева в связном кратном графе доказывается в статьях [2] (для делимого графа) и [3] (для произвольного кратного графа). В статье [4] мы доказали, что задачи 1 и 2 являются NP-трудными даже в случае делимого графа, если кратность $k \geq 3$. Предположительно, задачи являются NP-трудными и для кратности $k = 2$.

2.2. Точный алгоритм для задачи 2

В статьях [2, 3] мы получили эвристические алгоритмы для задачи 2 для случаев делимого и произвольного кратного графа. Построим теперь точный алгоритм для задачи 2 для случая делимого графа. Будем формулировать алгоритм для графов кратности 2 (далее будет показано, что аналогичный алгоритм может быть получен и для любой другой кратности $k > 2$).

Пусть имеется взвешенный делимый граф $G(V, E)$ кратности 2.

Алгоритм 1 (редукция графа).

1. Находим все множества достижимости по обычным ребрам R_x^o (используется полиномиальный алгоритм 2 из статьи [1]).
2. С помощью алгоритма Краскала (см. [14]) находим минимальное остовное дерево T_x^o в каждом подграфе G_x^o , образованном множеством вершин R_x^o и всеми обычными ребрами, которые соединяют вершины из R_x^o в графе $G(V, E)$. Множеством ребер дерева T_x^o будет $E_x^o \subseteq E^o$. Получение таких деревьев является необходимым и достаточным условием полноты кратного остовного дерева в делимом графе (теорема 3 из статьи [2]).
3. Обозначим каждое найденное дерево через T_r^p , где r — это номер дерева, а p — это номер части G_p исходного графа G .
4. Получим новый делимый граф $G_{red}(V_{red}, E_{red})$ следующим образом:
 - (а) Поместим все кратные вершины из множества V в множество V_{red} . Поместим обычные вершины T_r^p в множество V_{red} . Каждая из них соответствует обычному дереву с предыдущего шага.

- (b) Установим $E_{red}^k = E^k$ и $E_{red}^o = \emptyset$.
- (c) Получим мультиребра $\{x_i, \{T_r^1, T_s^2\}\}$ на основе всех мультиребер $\{x_i, \{x_t, x_u\}\} \in E^m$, заменяя обычные вершины x_t и x_u на обычные вершины T_r^1 и T_s^2 , которые соответствуют обычным деревьям, содержащим x_t и x_u . Добавим все такие мультиребра $\{x_i, \{T_r^1, T_s^2\}\}$ в множество E_{red}^m .
5. Граф G_{red} является делимым и не содержит обычных ребер. Редуцируем его до обычного графа $G_{ord}(V_{ord}, E_{ord})$. Некоторые вершины нового графа будут соответствовать сразу паре вершин графа G_{red} . Такие вершины будем называть *квазивершинами* здесь и далее.
- (a) Создадим *квазивершины* $q_{rs} = \{T_r^1, T_s^2\}$ для каждого мультиребра $\{x_i, \{T_r^1, T_s^2\}\} \in E_{red}^m$ и включим эти квазивершины в множество V_{ord} .
- (b) Преобразуем каждую кратную вершину $x_i \in V_{red}$ в обычную вершину $x_i \in V_{ord}$.
- (c) Преобразуем каждое кратное ребро $\{x_i, x_j\} \in E_{red}^k$ в обычное ребро $\{x_i, x_j\} \in E_{ord}$ той же длины.
- (d) Преобразуем каждое мультиребро $\{x_i, \{T_r^1, T_s^2\}\} \in E_{red}^m$ в обычное ребро $\{x_i, q_{rs}\} \in E_{ord}$ той же длины.

Заметим, что каждое минимальное полное остовное дерево графа G_{red} соответствует минимальному полному остовному дереву графа G . Последнее может быть получено заменой всех вершин T_r^p в минимальном полном остовном дереве графа G_{red} на обычные минимальные остовные деревья, найденные на шаге 2 алгоритма 1. Более того, каждое остовное дерево графа G_{red} является полным, поскольку в графе G_{red} нет обычных ребер.

Граф G_{ord} строится таким образом, что каждое остовное дерево в нем соответствует полному остовному дереву в графе G_{red} . Однако минимальное остовное дерево в графе G_{ord} может не соответствовать минимальному полному остовному дереву в графе G_{red} . Причина в том, что каждая квазивершина представляет собой пару обычных деревьев. Эти пары могут пересекаться, поэтому какие-то квазивершины будут «избыточными» для минимального остовного дерева в графе G_{red} . Тем не менее, эти «избыточные» вершины должны присутствовать в остовном дереве в графе G_{ord} .

Вышесказанное приводит к идее исключения каких-то квазивершин при поиске минимального остовного дерева в графе G_{ord} . Мы можем перебрать все возможные варианты такого исключения, получить минимальные остовные деревья в обычном графе для тех вариантов, в которых графы G_{ord} и G_{red} остаются связными, и выбрать кратчайшее из деревьев, которое и будет соответствовать минимальному полному остовному дереву в графе G_{red} .

Но этот подход требует перебора 2^q вариантов, где q — это количество квазивершин. Заметим, что мы можем уменьшить глубину перебора, если предварительно определим квазивершины, которые являются или не являются обязательными для графа производного от графа G_{ord} .

Разобьем множество квазивершин на четыре непересекающихся подмножества:

- Q_A — это множество квазивершин, которые обеспечивают смежность каких-то множеств достижимости по кратным ребрам R_x^k и R_y^k в графе G_{red} . Это значит, что в графе G_{red} невозможно получить остовное дерево, не включив в него некоторое мультиребро, соединяющее вершину из множества R_x^k с вершинами из q_{rs} , и мультиребро, соединяющее вершину из множества R_y^k с вершинами из q_{rs} , где $q_{rs} \in Q_A$;
- Q_B — это множество квазивершин, которые *могут* обеспечивать смежность каких-то множеств достижимости по кратным ребрам R_x^k и R_y^k в графе G_{red} . Однако можно обеспечить указанную смежность и альтернативным способом, если определенная вершина из множества Q_B будет исключена из графа G_{ord} ;
- Q_C — это множество квазивершин, которые являются листьями в графе G_{ord} ; при этом хотя бы одна из компонент T_r^1, T_s^2 квазивершины $q_{rs} \in Q_C$ не является компонентой никакой квазивершины из $Q_A \cup Q_B$;

- Q_D — это множество квазивершин, которые являются листьями в графе G_{ord} ; при этом обе компоненты T_r^1, T_s^2 квазивершины $q_{rs} \in Q_D$ являются компонентами каких-то квазивершин из $Q_A \cup Q_B$.

Очевидно, $Q_A \cup Q_B$ содержит все квазивершины из V_{ord} , не являющиеся листьями.

Мы можем найти все множества Q_A, Q_B, Q_C и Q_D с помощью следующего полиномиального алгоритма. Также мы можем профильтровать Q_D , исключив квазивершины, которые гарантированно «избыточны». В то же время, множество Q_A состоит из квазивершин, которые не могут быть исключены, поскольку минимальное остовное дерево в графе G_{red} должно содержать все пары их компонент (как отдельные концы мультиребер).

Алгоритм 2 (классификация и фильтрация квазивершин графа G_{ord}).

1. Последовательно просмотрим все квазивершины q_{rs} , для которых $\deg q_{rs} > 1$. Если $\deg q_{rs} = n$, у нас есть n ребер $\{x_{rs}^i, q_{rs}\} \in E_{ord}$ ($i \in \overline{1, n}$). Временно исключим все эти ребра из E_{ord} и проверим существование n путей $S(x_{rs}^1, x_{rs}^2), \dots, S(x_{rs}^{n-1}, x_{rs}^n)$ и $S(x_{rs}^n, x_{rs}^1)$ (фактически мы проверяем связность графа G_{ord} после исключения вершины q_{rs}). Если хотя бы один из путей не существует, включаем q_{rs} в Q_A . Иначе включаем q_{rs} в Q_B .
2. Последовательно просмотрим все квазивершины q_{rs} , для которых $\deg q_{rs} = 1$. Возьмем их компоненты T_r^1 и T_s^2 и проверим, являются ли они компонентами каких-то квазивершин из $Q_A \cup Q_B$. Если для обеих компонент ответ «да», включим q_{rs} в Q_D . Иначе включим q_{rs} в Q_C .
3. Профильтруем Q_D . Последовательно просмотрим все квазивершины $q_{rs} \in Q_D$. Возьмем их компоненты T_r^1 и T_s^2 и проверим, являются ли они обе компонентами каких-то квазивершин из Q_A . Если для обеих компонент ответ «да», исключим q_{rs} из Q_D и из V_{ord} . Также исключим единственное инцидентное этой квазивершине ребро $\{x_i, q_{rs}\}$ из E_{ord} .

Далее будем рассматривать граф G_{ord} в виде, полученном на шаге 3 алгоритма 2.

Определение 15. Обычный граф G_{der} с квазивершинами называется *производным от графа G_{ord}* , если он получен из графа G_{ord} в результате исключения каких-то квазивершин из V_{ord} и исключения всех инцидентных этим квазивершинам ребер из E_{ord} . Граф G_{der} должен также соответствовать следующим условиям:

1. G_{der} связан.
2. Каждое остовное дерево в графе G_{der} соответствует полному остовному дереву в графе G_{red} .

Заметим, что только квазивершины из Q_B, Q_C или Q_D могут исключаться из V_{ord} при получении производного графа, поскольку все квазивершины из Q_A обязательны для связности графа. Мы можем получить все возможные производные графы G_{der} и найти минимальные остовные деревья в них. Очевидно, кратчайшее из этих деревьев порождает минимальное полное остовное дерево в делимом графе G_{red} .

Обозначим через DER множество всех графов производных от G_{ord} .

Алгоритм 3 (генерация всех графов производных от G_{ord}).

1. Определим *базовую часть* G_{base} с множествами вершин и ребер V_{base} и E_{base} . Она содержит вершины и ребра, которые являются обязательными для каждого производного графа. Поэтому базовая часть должна содержать все обычные вершины из V_{ord} , все квазивершины из Q_A и все ребра из E_{ord} , связывающие указанные вершины.
2. Рассмотрим множество $Q_B = \{q_{r_1 s_1}, \dots, q_{r_p s_p}\}$. Найдем все его подмножества:

$$Q_B^1 = \emptyset, \quad Q_B^2 = \{q_{r_1 s_1}\}, \quad \dots, \quad Q_B^{p+1} = \{q_{r_p s_p}\},$$

$$Q_B^{p+2} = \{q_{r_1 s_1}, q_{r_2 s_2}\}, \quad \dots, \quad Q_B^{2p} = Q_B.$$

Обозначим через E_B^j множество ребер инцидентных вершинам из Q_B^j в графе G_{ord} ($j \in \overline{1, 2^p}$). Положим $DER = \emptyset$, $i = 1$.

3. Сгенерируем граф G_{der}^i производный от G_{ord} . Его множества вершин и ребер обозначим через V_{der}^i и E_{der}^i . Сначала положим $V_{der}^i = V_{base} \cup Q_B^i$ и $E_{der}^i = E_{base} \cup E_B^i$. Если граф не связен, производный граф G_{der}^i не существует, переходим на шаг 7. Иначе переходим на шаг 4.
4. Сформируем множество Q_D^i . Для этого произведем дополнительную фильтрацию множества Q_D . Последовательно просмотрим все квазивершины $q_{rs} \in Q_D$. Возьмем их компоненты T_r^1 и T_s^2 и проверим, являются ли они компонентами каких-то квазивершин из $Q_A \cup Q_B^i$. Если хотя бы для одной компоненты ответ «нет», включим q_{rs} в Q_D^i .
5. Рассмотрим множество $Q_C \cup Q_D^i$. Каждая квазивершина q_{rs} из этого множества содержит хотя бы одну компоненту T_r^1 или T_s^2 , которая не является компонентой никакой квазивершины из множества $Q_A \cup Q_B^i$. Все такие компоненты образуют два множества:

$$T^1 = \{T_{r_1}^1, \dots, T_{r_l}^1\}, \quad T^2 = \{T_{s_1}^2, \dots, T_{s_n}^2\}.$$

Для каждой квазивершины $q_{rs} \in Q_C \cup Q_D^i$ определим ее вес $w(q_{rs}) = l(\{x_t, q_{rs}\})$, где $\{x_t, q_{rs}\}$ — это единственное ребро, инцидентное q_{rs} .

Затем выделим подмножество Q_{CD}^i минимального суммарного веса $\sum_{q_{rs} \in Q_{CD}^i} w(q_{rs})$ среди всех

подмножеств множества $Q_C \cup Q_D^i$ такое, что оно покрывает T^1 и T^2 , то есть каждый элемент T^1 и T^2 является компонентой хотя бы одной квазивершины из подмножества. Очевидно, что

$$\max\{l, n\} \leq |Q_{CD}^i| \leq l + n.$$

Определим $E_{CD}^i \subseteq E_{ord}$ как множество ребер инцидентных квазивершинам из множества Q_{CD}^i .

6. Граф G_{der}^i определяется множествами вершин и ребер:

$$V_{der}^i = V_{base} \cup Q_B^i \cup Q_{CD}^i, \quad E_{der}^i = E_{base} \cup E_B^i \cup E_{CD}^i.$$

Добавим этот граф G_{der}^i в множество DER .

7. Положим $i = i + 1$. Если $i > 2^p$, выход. Иначе переходим на шаг 3.

Заметим, что алгоритм 3 выполняется за экспоненциальное число шагов в общем случае. Действительно, мы ищем $2^{|Q_B|}$ подмножеств на шаге 2 и выполняем шаги 3–7 для каждого подмножества. Более того, на шаге 5 требуется выбрать одно из $\left(C_{|Q_C \cup Q_D^i|}^{\max\{l, n\}} + \dots + C_{|Q_C \cup Q_D^i|}^{l+n} \right)$ подмножеств в худшем случае, что также требует экспоненциального числа операций.

Но задача 2 NP-трудная, поэтому любой точный алгоритм для нее потребует экспоненциального числа шагов, если $P \neq NP$. Тем не менее, алгоритм 3 будет выполняться за приемлемое время, если $|Q_B|$, $|Q_C|$ и $|Q_D|$ невелики в то время, как $|Q_A|$ может быть большим.

Будем использовать алгоритмы 1–3 на шагах точного алгоритма для задачи о минимальном полном остовном дереве в делимом кратном графе. Пусть $G(V, E)$ — взвешенный связный делимый кратный граф. Будем искать его минимальное полное остовное дерево $T_{complete}^{\min}(V, E^T)$, где $E^T \subseteq E$.

Алгоритм 4 (точный алгоритм для задачи 2 для делимого графа).

1. С помощью алгоритма 1 находим все обычные минимальные остовные деревья T_r^p на множествах достижимости по обычным ребрам R_x^o , редуцированный кратный граф G_{red} и обычный граф G_{ord} с квазивершинами.
2. Классифицируем и фильтруем квазивершины графа G_{ord} с помощью алгоритма 2.
3. Рассмотрим граф G_{ord} , полученный после фильтрации на шаге 2. С помощью алгоритма 3 находим множество DER всех графов G_{der}^i производных от G_{ord} .

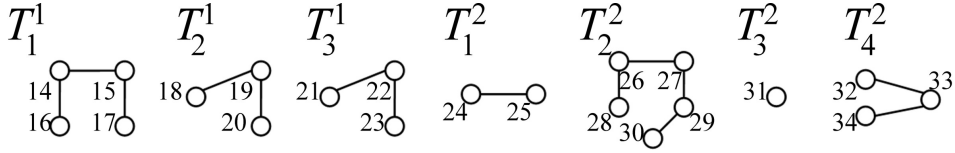


Fig. 5. The minimum spanning trees for the ordinary edges reachability sets

Рис. 5. Минимальные остовные деревья на множествах достижимости по обычным ребрам

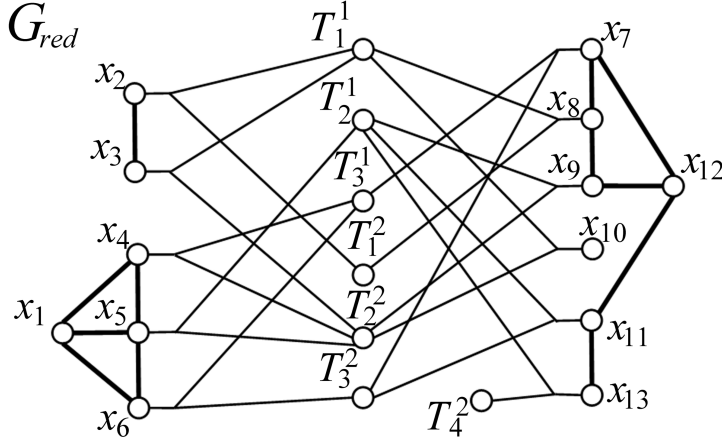


Fig. 6. Reduced divisible graph G_{red}

Рис. 6. Редуцированный кратный граф G_{red}

4. В каждом графе $G_{der}^i \in DER$ найдем минимальное остовное дерево T_{der}^i с помощью алгоритма Краскала. Выберем кратчайшее из них и обозначим его через $T_{der}^{\min}(V_{der}^{\min}, E_{der}^{\min})$.
5. Преобразуем дерево $T_{der}^{\min}$ в минимальное полное остовное дерево $T_{red}^{\min}(V_{red}, E_{red}^T)$ графа G_{red} , где $E_{red}^T \subseteq E_{red}$. Получим E_{red}^T следующим образом.
 - (а) Если для каких-то обычных вершин $x_t \in V_{der}^{\min}$, $x_u \in V_{der}^{\min}$ ребро $\{x_t, x_u\} \in E_{der}^{\min}$, включаем кратное ребро $\{x_t, x_u\} \in E_{red}$ в множество E_{red}^T .
 - (б) Если для какой-то обычной вершины $x_t \in V_{der}^{\min}$ и для какой-то квазивершины $q_{rs} \in V_{der}^{\min}$ ребро $\{x_t, q_{rs}\} \in E_{der}^{\min}$, включаем мультиребро $\{x_t, \{T_r^1, T_s^2\}\} \in E_{red}$ в множество E_{red}^T .
6. Преобразуем дерево $T_{red}^{\min}$ в минимальное полное остовное дерево $T_{complete}^{\min}$ графа G . Для этого заменим все вершины $T_r^p \in V_{red}$ на соответствующие обычные деревья T_r^p , полученные на шаге 1.

2.3. Пример работы алгоритма

Пример 3. Рассмотрим делимый граф G кратности 2, показанный на рис. 2. Установим длины всех кратных ребер равными 2, длины всех обычных ребер равными 1, $l(\{x_4, \{x_{21}, x_{26}\}\}) = 2$, $l(\{x_6, \{x_{21}, x_{31}\}\}) = 2$, $l(\{x_7, \{x_{22}, x_{31}\}\}) = 6$, $l(\{x_{11}, \{x_{20}, x_{31}\}\}) = 2$, а длины всех остальных мультиребер равными 4.

Применим алгоритм 4 к графу G . Сначала найдем минимальные остовные деревья на множествах достижимости по обычным ребрам (рис. 5). Затем заменим их на соответствующие вершины T_r^p и получим редуцированный кратный граф G_{red} (рис. 6).

Далее сформируем квазивершины, чтобы получить граф G_{ord} (рис. 7 (а), квазивершины закрашены черным). Классификация квазивершин дает следующие множества:

$$Q_A = \{q_{11}, q_{12}\}, \quad Q_B = \{q_{22}, q_{33}\}, \quad Q_C = \{q_{24}\}, \quad Q_D = \{q_{23}, q_{32}\}.$$

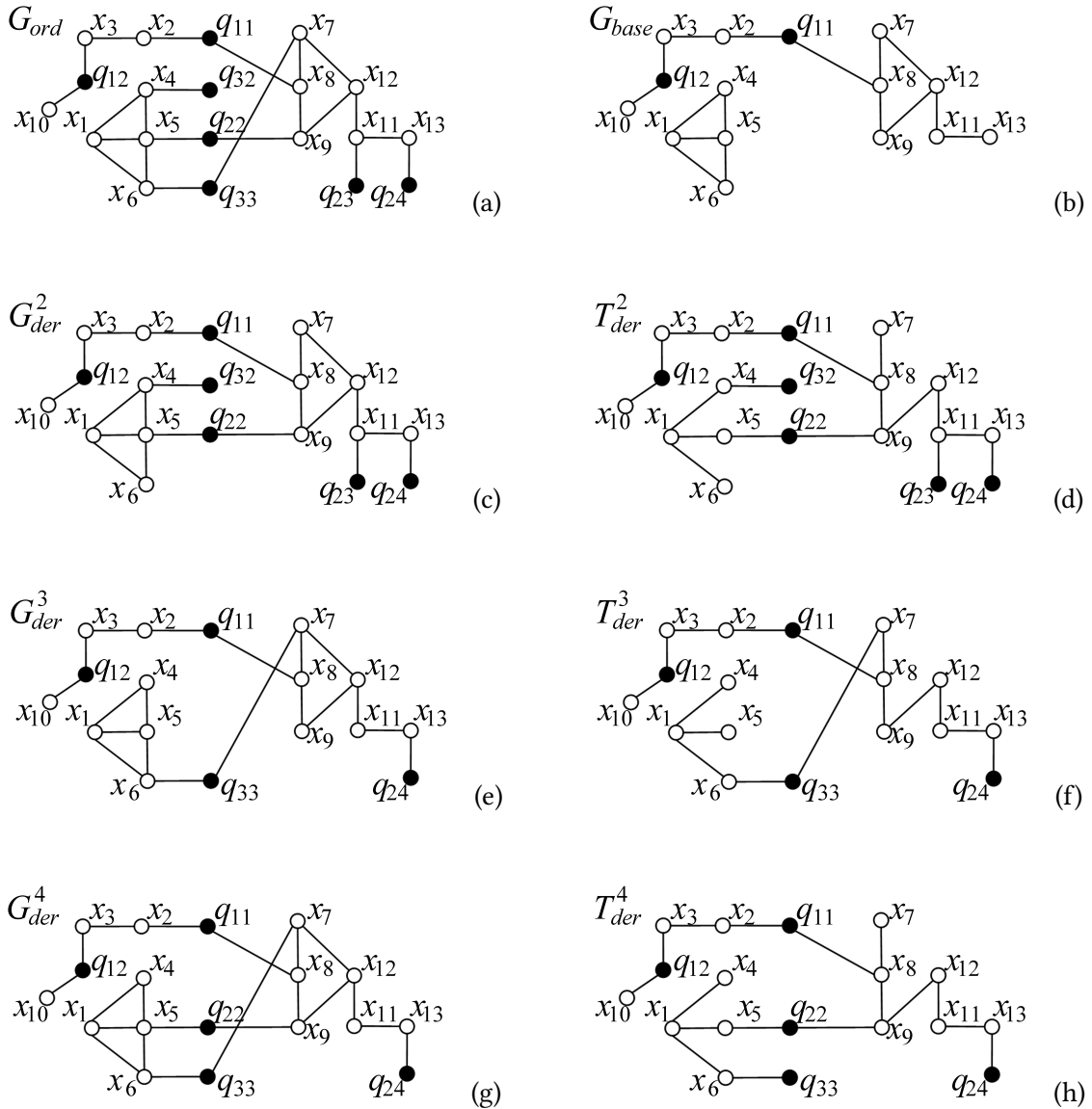


Fig. 7. Ordinary graph G_{ord} with quasi-vertices, its derivative graphs and theirs minimum spanning trees

Рис. 7. Обычный граф G_{ord} с квазивершинами, его производные графы и их минимальные остовные деревья

Базовая часть G_{base} графа G_{ord} состоит из всех обычных вершин, всех квазивершин из множества Q_A и из всех ребер, соединяющих эти вершины, см. рис. 7 (b). Сформируем подмножества Q_B^i ($i \in \overline{1, 4}$):

$$Q_B^1 = \emptyset, \quad Q_B^2 = \{q_{22}\}, \quad Q_B^3 = \{q_{33}\}, \quad Q_B^4 = \{q_{22}, q_{33}\},$$

и попробуем получить производные графы G_{der}^i ($i \in \overline{1, 4}$). Граф G_{der}^1 не существует, потому что добавление Q_B^1 в G_{base} приводит к несвязному графу. Остальные производные графы определяются следующими подмножествами:

$$Q_D^2 = \{q_{23}, q_{32}\}, \quad Q_{CD}^2 = \{q_{23}, q_{24}, q_{32}\};$$

$$Q_D^3 = \{q_{23}\}, \quad Q_{CD}^3 = \{q_{24}\};$$

$$Q_D^4 = \emptyset, \quad Q_{CD}^4 = \{q_{24}\}.$$

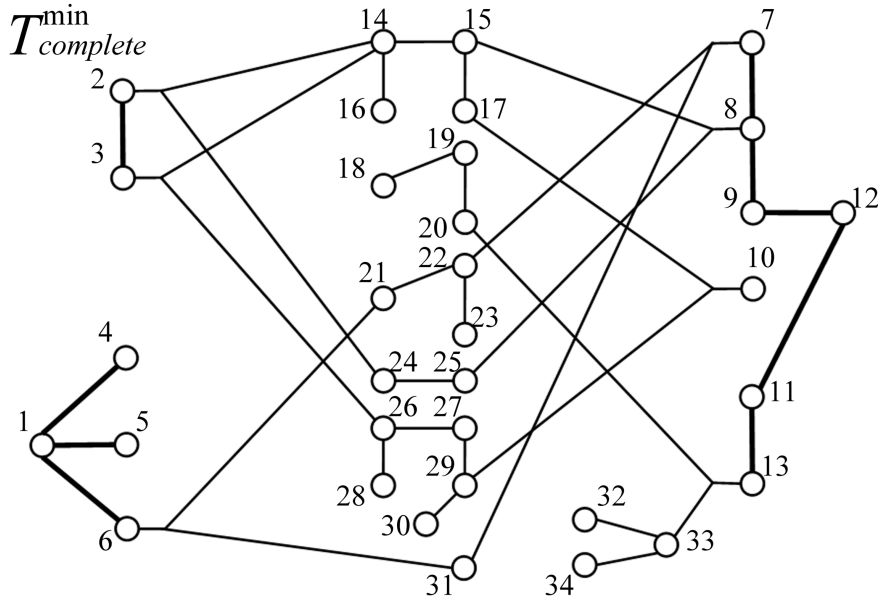


Fig. 8. The minimum complete spanning tree of the divisible graph G

Рис. 8. Минимальное полное остовное дерево делимого графа G

Производные графы показаны на рис. 7 (с), (е), (г). Чтобы получить минимальное остовное дерево в каждом графе, мы должны исключить ребра $\{x_4, x_5\}$, $\{x_5, x_6\}$ и $\{x_7, x_{12}\}$. Также мы должны исключить ребро $\{x_7, q_{33}\}$ из G_{der}^4 . В итоге мы получим минимальные остовные деревья T_{der}^2 , T_{der}^3 и T_{der}^4 этих графов; они показаны на рис. 7 (d), (f), (h). Их веса:

$$w(T_{der}^2) = 50, \quad w(T_{der}^3) = 46, \quad w(T_{der}^4) = 48.$$

Кратчайшее из них — T_{der}^3 , оно порождает минимальное полное остовное дерево $T_{complete}^{min}$ исходного графа G веса $w(T_{complete}^{min}) = 60$ (рис. 8).

2.4. Анализ алгоритмов

Алгоритмы 1–4 сформулированы для делимых графов кратности 2. Однако они могут быть легко переформулированы для любой другой кратности $k \geq 3$. Действительно, отличия будут только в количестве частей G_p делимого графа ($p \in 1, k$), в диапазоне значений индекса p для обычных деревьев T_r^p , а также в количестве компонент квазивершины. В общем случае у каждой квазивершины будет ровно k компонент, поэтому нужно рассматривать квазивершины вида $q_{r_1 \dots r_k} = \{T_{r_1}^1, \dots, T_{r_k}^k\}$. Все остальное остается прежним.

Пусть теперь G — делимый граф произвольной кратности $k \geq 2$. Применим алгоритм 1 к графу G и получим обычный граф G_{ord} с квазивершинами.

Определение 16. Делимый кратный граф G назовем *графом простой структуры*, если в соответствующем обычном графе G_{ord} с квазивершинами $|Q_B| = |Q_C| = 0$.

Теорема 1. Задача о минимальном полном остовном дереве в делимом кратном графе простой структуры может быть решена за полиномиальное время.

Доказательство. Во-первых, из условия $|Q_B| = 0$ следует, что все компоненты $T_{r_1}^1, \dots, T_{r_k}^k$ каждой квазивершины $q_{r_1 \dots r_k} \in Q_D$ являются компонентами каких-то квазивершин из Q_A . Поэтому все квазивершины будут исключены из $|Q_D|$ на шаге 3 алгоритма 2. Значит, мы можем пропустить

шаги 4–5 алгоритма 3, поскольку $Q_D = \emptyset$ после фильтрации и $Q_C = \emptyset$ в силу простоты структуры графа.

Во-вторых, из условия $|Q_B| = 0$ следует, что только одно подмножество $Q_B^1 = \emptyset$ будет найдено на шаге 2 алгоритма 3. Поэтому шаги 3–7 этого алгоритма будут выполнены ровно один раз и ровно один граф G_{der}^1 будет производным от графа G_{ord} . Очевидно, что $V_{der}^1 = V_{ord}$ и $E_{der}^1 = E_{ord}$.

Следовательно, алгоритм 3 выполняется за полиномиальное время для любого делимого графа простой структуры. Шаг 4 алгоритма 4 выполняется ровно один раз, поскольку у нас есть только один граф производный от графа G_{ord} .

В итоге, все шаги алгоритма 4 выполняются однократно и каждый шаг полиномиален для данного класса графов. Таким образом, задача 2 полиномиальна в случае делимого кратного графа простой структуры.

□

Если мы будем применять алгоритм 4 на практике, мы можем столкнуться с проблемой нехватки памяти из-за того, что в алгоритме мы получаем экспоненциальное количество производных графов. Чтобы преодолеть указанную проблему, мы можем объединить шаги 3 и 4 алгоритма 4 следующим образом.

1. Зададим переменную $w_{cur} = \infty$ для хранения текущего веса минимального остовного дерева в производном графе. Обозначим текущее минимальное остовное дерево веса w_{cur} через T_{cur} . В начальный момент $T_{cur} = \emptyset$. Таким образом, мы будем хранить дерево T_{cur} и переменную w_{cur} вместо множества DER .
2. После шага 6 алгоритма 3 (поиск очередного графа G_{der}^i) переходим на шаг 4 алгоритма 4 для поиска минимального остовного дерева T_{der}^i в графе G_{der}^i . Если $w(T_{der}^i) < w_{cur}$, полагаем $T_{cur} = T_{der}^i$ и $w_{cur} = w(T_{der}^i)$. Переходим на шаг 7 алгоритма 3.
3. После последней итерации указанной процедуры полагаем $T_{der}^{min} = T_{cur}$.

Также мы можем оптимизировать алгоритм для случая графа с большим количеством вершин и ребер. Мы можем сравнивать текущий вес $w(T_{der}^i)$ с весом w_{cur} на шаге 4 алгоритма 4 после добавления каждого нового ребра алгоритмом Краскала. Если $w(T_{der}^i) \geq w_{cur}$, можно прервать построение дерева и перейти к следующей итерации.

References

- [1] A. V. Smirnov, “The shortest path problem for a multiple graph”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 625–633, 2018. doi: [10.3103/S0146411618070234](https://doi.org/10.3103/S0146411618070234).
- [2] A. V. Smirnov, “The spanning tree of a divisible multiple graph”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 871–879, 2018. doi: [10.3103/S0146411618070325](https://doi.org/10.3103/S0146411618070325).
- [3] A. V. Smirnov, “Spanning tree of a multiple graph”, *Journal of Combinatorial Optimization*, vol. 43, no. 4, pp. 850–869, 2022. doi: [10.1007/s10878-021-00810-5](https://doi.org/10.1007/s10878-021-00810-5).
- [4] A. V. Smirnov, “NP-completeness of the minimum spanning tree problem of a multiple graph of multiplicity $k \geq 3$ ”, *Automatic Control and Computer Sciences*, vol. 56, no. 7, pp. 788–799, 2022. doi: [10.3103/S0146411622070173](https://doi.org/10.3103/S0146411622070173).
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd. The MIT Press, McGraw-Hill Book Company, 2009.
- [6] C. Berge, *Graphs and Hypergraphs*. North-Holland Publishing Company, 1973.
- [7] A. Basu and R. W. Blanning, “Metagraphs in workflow support systems”, *Decision Support Systems*, vol. 25, no. 3, pp. 199–208, 1999. doi: [10.1016/S0167-9236\(99\)00006-8](https://doi.org/10.1016/S0167-9236(99)00006-8).

- [8] A. Basu and R. W. Blanning, *Metagraphs and Their Applications* (Integrated Series in Information Systems). Springer US, 2007, vol. 15.
- [9] V. S. Rublev and A. V. Smirnov, “Flows in multiple networks”, *Yaroslavsky Pedagogichesky Vestnik*, vol. 3, no. 2, pp. 60–68, 2011, in Russian.
- [10] A. V. Smirnov, “The problem of finding the maximum multiple flow in the divisible network and its special cases”, *Automatic Control and Computer Sciences*, vol. 50, no. 7, pp. 527–535, 2016. DOI: [10.3103/S0146411616070191](https://doi.org/10.3103/S0146411616070191).
- [11] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.
- [12] V. S. Rublev and A. V. Smirnov, “The problem of integer-valued balancing of a three-dimensional matrix and algorithms of its solution”, *Modeling and Analysis of Information Systems*, vol. 17, no. 2, pp. 72–98, 2010, in Russian.
- [13] A. V. Smirnov, “Network model for the problem of integer balancing of a four-dimensional matrix”, *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 558–566, 2017. DOI: [10.3103/S0146411617070185](https://doi.org/10.3103/S0146411617070185).
- [14] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem”, *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956. DOI: [10.1090/S0002-9939-1956-0078686-7](https://doi.org/10.1090/S0002-9939-1956-0078686-7).

Modeling of Parallel Program Synchronization Primitives

O. S. Kryukov¹, A. G. Voloshko¹, A. N. Ivutin¹

DOI: [10.18255/1818-1015-2025-2-150-171](https://doi.org/10.18255/1818-1015-2025-2-150-171)

¹Tula State University, Tula, Russia

MSC2020: 68Q85

Research article

Full text in Russian

Received September 29, 2024

Revised April 1, 2025

Accepted May 7, 2025

This article is devoted to the problem of verifying parallel programs that may contain special types of errors associated with the synchronization of parallel executed threads and access to shared memory. Such errors include deadlocks and data races. There is a division of parallel program verification methods into static and dynamic. The second ones require running the code and allow to check only the current implementation of the program for races, which, if there are a large number of branches, can lead to missing races. Among static methods, analytical methods (for example, based on deductive analysis) and model checking methods are most widely used. However, they are difficult to implement, and model checking still require a significant amount of manual work from the programmer to build such a model. In this regard, it is necessary to use models that can be built automatically. Previously, the authors developed a model based on an extension of Petri nets, which allows automatic creation based on sequential code and its conversion into parallel code. Automatic creation of a model of a parallel program introduces new, previously unused requirements related to the interaction of parallel threads. Thus, this article discusses the features of modeling using extended Petri nets with semantic relations of the main synchronization primitives implemented in most languages and parallel programming technologies for shared memory systems. In the future, these models will be used to search for data races and deadlocks in parallel programs.

Keywords: verification; parallel program; Petri nets; synchronization primitives

INFORMATION ABOUT THE AUTHORS

Kryukov, Oleg S. (corresponding author)	ORCID iD: 0000-0003-3155-6266 . E-mail: ol.kryukov97@mail.ru Postgraduate Student
Voloshko, Anna G.	ORCID iD: 0000-0002-4304-2513 . E-mail: atroshina@mail.ru Ph.D, Docent, Associate Professor of the Department of Computer Engineering
Ivutin, Alexey N.	ORCID iD: 0000-0003-2970-2148 . E-mail: alexey.ivutin@gmail.com Dr. Sc., Professor, Head of the Department of Computer Engineering

For citation: O. S. Kryukov, A. G. Voloshko, and A. N. Ivutin, “Modeling of parallel program synchronization primitives”, *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 150–171, 2025. DOI: [10.18255/1818-1015-2025-2-150-171](https://doi.org/10.18255/1818-1015-2025-2-150-171).

Моделирование примитивов синхронизации параллельных программ

О. С. Крюков¹, А. Г. Волошко¹, А. Н. Ивутин¹

DOI: [10.18255/1818-1015-2025-2-150-171](https://doi.org/10.18255/1818-1015-2025-2-150-171)

¹Тульский государственный университет, Тула, Россия

УДК 519.876.5

Научная статья

Полный текст на русском языке

Получена 29 сентября 2024 г.

После доработки 1 апреля 2025 г.

Принята к публикации 7 мая 2025 г.

Данная статья посвящена проблеме верификации параллельных программ, которые могут содержать особые виды ошибок, связанных с синхронизацией параллельно исполняемых потоков и доступом к общей памяти. К таким ошибкам относятся тупики и гонки данных. Существует разделение методов верификации параллельных программ на статические и динамические. Последние требуют запуска кода и позволяют проверить на гонки лишь текущую реализацию программы, что при наличии большого числа ветвлений может привести к пропуску гонок. Среди статических методов наибольшее применение нашли аналитические методы (например, на основе дедуктивного анализа) и методы проверки моделей. Однако они сложны в реализации, а последние по-прежнему требуют от программиста значительного объёма ручной работы для построения модели. В этой связи необходимо использование моделей, которые могут быть построены автоматически. Ранее авторами была разработана модель на основе расширения сетей Петри, позволяющая автоматическое построение на основе последовательного кода и преобразование её в параллельный код. Автоматическое построение модели параллельной программы вводит новые, ранее не использовавшиеся требования, связанные со взаимодействием параллельных потоков. Таким образом, в данной статье рассматриваются особенности моделирования с использованием расширенных сетей Петри с семантическими связями основных примитивов синхронизации, реализуемых в большинстве языков и технологий параллельного программирования для систем с общей памятью. В дальнейшем на основе этих моделей будет проводиться поиск гонок данных и тупиков для параллельных программ.

Ключевые слова: верификация; параллельная программа; сети Петри; примитивы синхронизации

ИНФОРМАЦИЯ ОБ АВТОРАХ

Крюков, Олег Сергеевич
(автор для корреспонденции)

ORCID iD: [0000-0003-3155-6266](https://orcid.org/0000-0003-3155-6266). E-mail: ol_kryukov97@mail.ru
Аспирант

Волошко, Анна Геннадьевна

ORCID iD: [0000-0002-4304-2513](https://orcid.org/0000-0002-4304-2513). E-mail: atroshina@mail.ru
Канд. тех. наук, доцент, доцент кафедры вычислительной техники

Ивутин, Алексей Николаевич

ORCID iD: [0000-0003-2970-2148](https://orcid.org/0000-0003-2970-2148). E-mail: alexey.ivutin@gmail.com
Доктор тех. наук, профессор, заведующий кафедрой вычислительной техники

Для цитирования: O. S. Kryukov, A. G. Voloshko, and A. N. Ivutin, “Modeling of parallel program synchronization primitives”, *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 150–171, 2025. DOI: [10.18255/1818-1015-2025-2-150-171](https://doi.org/10.18255/1818-1015-2025-2-150-171).

Введение

Современный этап развития вычислительных систем характеризуется использованием параллельных архитектур. И если простые задачи все еще пишутся для работы на одном ядре, то для более сложных задач уже желательно использовать все возможные мощности. Одним из классов таких сложных систем являются интеллектуальные системы, которые часто имеют дело с обработкой больших объемов информации, что достаточно хорошо может быть выполнено в параллельном режиме. В последнее время разрабатывается все больше параллельных алгоритмов для различных интеллектуальных задач, например, для задач анализа данных [1], для задач нечёткой логики [2], нейросетей [3] и т. д.

Однако, программирование для параллельных систем представляет значительную сложность в связи с необходимостью координации выполнения потоков и управления общими данными. Для этого применяются различные механизмы синхронизации. Более того, многопоточные программы могут содержать дополнительно ошибки, связанные с совместным использованием ресурсов, поиск которых сопряжен со значительными сложностями и требует применения вычислительно затратных методов и алгоритмов. Краткий обзор методов проверки параллельных программ представлен в разделе 1. На наш взгляд, наиболее перспективным является метод статического анализа на основе проверки модели. Статические методы в отличие от динамических рассматривают весь код программы, а не только один из вариантов её исполнения, когда множество альтернативных путей развития алгоритма не может быть проанализировано. К сожалению, недостатком любых средств статического анализа кода, и в особенности метода проверки моделей (включая поиск гонок данных), является наличие большего по сравнению с динамическими средствами количества ложноположительных срабатываний, когда анализатор указывает на ошибку при её реальном отсутствии [4]. Чаще всего такое происходит при наличии чередований потоков, организованных нетиповыми способами, которые не учтены в анализаторе.

В качестве модели программного кода для статического анализа параллельных программ можно использовать расширенные сети Петри с семантическими связями (РСПСС) [5]. Как было показано ранее [6], с помощью таких сетей можно представлять параллельные алгоритмы и анализировать наличие семантических отношений между параллельными потоками. В дополнение к операциям создания и завершения потоков, которые представлены соответствующими управляющими переходами `fork` и `join`, модель обеспечивает переход типа `synchro` для барьерной синхронизации. Этих примитивов синхронизации было достаточно для решения задач автоматизации распараллеливания с помощью РСПСС. Анализ изначально параллельных программ требует наряду с проверкой каждого последовательного потока решения более сложной проблемы исследования межпоточкового взаимодействия. Вопросы представления последовательных участков кода, а также процедур ветвления и завершения потоков в этой модели несложны и аналогичны представленным в [6]. Представление примитивов синхронизации связано с рядом проблем. Одна из них, учёт особенностей различных способов синхронизации доступа к ресурсам, помимо барьеров в реальных программах, и их представление в выбранной модели. И мы постараемся описать основные из них в этой статье.

1. Обзор методов анализа параллельных программ

Для начала всё же более подробно рассмотрим методы анализа параллельных программ. Двумя наиболее сложными проблемами являются тупики и гонки данных. Тупики возникают, когда ни один поток из некоторой группы потоков параллельной программы не может завершить свою работу по причине ожидания освобождения ресурсов другим потоком этой группы [7, 8]. Обнаружение тупиковых ситуаций чаще всего проводится на основе построения путей выполнения программы (трасс) и анализа отношений ожидания [9, 10], что позволяет определить, существует

ли трасса, содержащая тупиковое исполнение. Кроме того, обнаружение тупиковых ситуаций встроено в некоторые системы разработки и применяет механизм тайм-аутов, заключающийся в анализе времени, проведённого потоком в ожидании: если оно превысило определённый лимит на всех потоках, то выдаётся предупреждение о возникновении взаимной блокировки [7]; но данный способ применим только для обнаружения полной взаимоблокировки программы.

Несмотря на то, что проблема обнаружения тупиков является актуальной, но как было сказано, уже разработаны методы их выявления. Более сложной для обнаружения ошибкой в многопоточных программах являются гонки данных. Это связано с тем, что гонки данных зачастую связаны с трудновоспроизводимыми путями выполнения, что в свою очередь также усложняет и поиск тупиковых состояний. Для поиска гонок данных применяются различные средства, которые разделяют на статические и динамические методы обнаружения гонок данных.

Динамические методы используют информацию, полученную в результате конкретного выполнения параллельной программы. Пути выполнения ограничены теми, которые могут возникнуть при данных конкретных входных данных. Это приводит к уменьшению проверяемых на наличие гонок данных вариантов. Динамические методы обнаружения гонок данных обычно классифицируют на методы *post-mortem* и методы детектирования «на лету».

Первые анализируют отслеживаемую информацию или воспроизводят программу после выполнения. Например, при работе программы может динамически осуществляться сбор информации о ссылках на память, а при завершении работы выполняться их анализ на наличие конфликтов в параллельных участках (RecPlay [11]).

Методы «на лету» могут быть основаны на анализе отношения «произошло до», при котором проверяется, что операции взаимодействия с памятью при нескольких последовательных запусках осуществляются в одном и том же порядке (например, FastTrack [12], Djit+ [13], ThreadSanitizer [14], и т. д.), анализе блокировок, когда проверяется, что два конфликтующих события удерживают одну блокировку, что говорит об отсутствии гонок данных (например, Eraser [15]), или гибридном анализе (например AccuLock [16], SimpleLock [17], и т. д.).

Статические методы используют только исходный код для обнаружения гонок данных. Эти методы должны учитывать все возможные входные данные и все возможные пути выполнения при задании этих входных данных. Этот подход включает дедуктивный анализ и метод проверки моделей.

Дедуктивный анализ предлагает генерировать набор математических доказательств в соответствии с программным кодом и его спецификациями и выполнять эти спецификации с помощью либо вспомогательных средств доказательства (интерактивных средств доказательства теорем), либо автоматических средств доказательства теорем, в том числе SMT-решателей [18]. Истинность представленных доказательств означает, что система соответствует своей спецификации. Однако этот метод имеет значительную сложность, а объём доказательства корректности может значительно превышать объём исследуемой программы [19]. Метод проверки моделей является одним из широко используемых статических методов анализа параллельных программ [20]. Проверка модели — автоматизированный подход, позволяющий проверить, соответствует ли модель программы заданной спецификации (формальному описанию свойств системы). При этом в качестве описания спецификации и самого программного обеспечения могут использоваться как логико-алгебраические модели [21], так и исполняемые модели [22]. Проверка модели может выполняться путём перебора всех возможных состояний программы или путём задания булевых отношений, связывающих начальное и конечное состояния. Отдельно стоит отметить широкое использование графовых моделей для представления и анализа параллельных программ, отличающихся как наглядностью для человека, так и простотой компьютерного анализа. Среди них стоит отметить модель «опера-

ции — операнды» [23], а также различные модификации сетей Петри [24], в особенности, цветные сети Петри, например, [25–27].

Для использования этого метода требуется согласование между исходным кодом параллельной программы и сгенерированной моделью программы для обнаружения гонки данных и взаимоблокировок с использованием метода проверки модели. Применимость модели РСПСС для этого представления обсуждается далее в этой статье.

2. Расширенные сети Петри с семантическими связями

Приведём описание модели, используемой для представления параллельных программ. Расширенная сеть Петри с семантическими связями является иерархической сетью Петри, позиции данной сети являются математическими объектами, моделирующими операторы алгоритма. Структура сети задается следующим множеством:

$$\Pi = \left\{ A, \left\{ Z^C, \tilde{R}^C, \hat{R}^C \right\}, \left\{ Z^S, \tilde{R}^S, \hat{R}^S \right\} \right\},$$

где A — конечное множество позиций, представляющих операции программы; Z^C — конечное множество переходов по управлению, представляющих последовательность выполнения программы; Z^S — конечное множество переходов по семантическим связям, представляющих информационные зависимости между переменными; $\tilde{R}^C$ — матрица смежности, отображающая множество позиций в множество переходов по управлению; $\hat{R}^C$ — матрица смежности, отображающая множество переходов по управлению в множество позиций; $\tilde{R}^S$ — матрица смежности, отображающая множество позиций в множество переходов по семантическим связям; $\hat{R}^S$ — матрица смежности, отображающая множество переходов по семантическим связям в множество позиций. Кроме того, на сеть для анализа могут быть наложены дополнительные параметры. $M = \{h^C(t), \Lambda^C, \Lambda^S\}$, определяющие временные (вектор времени исполнения $h^C(t)$) и логические характеристики РСПСС (вектора условий срабатывания соответствующих переходов Λ^C, Λ^S).

Позиции РСПСС являются математическими объектами, моделирующими процесс выполнения операции исполнительным устройством в ЭВМ. Следует отметить, что сеть строится по принципу декомпозиции, и одна позиция РСПСС может сама представлять собой подсеть. В ЭВМ структура сети может быть представлена с использованием матриц смежности в разреженном виде, так как они преимущественно заполнены нулевыми элементами, что снижает требования к компьютерной памяти.

Переход по управлению в РСПСС является математическим объектом, моделирующим процесс передачи управления по выполнению операций с одного процесса на другой. В случае параллельного исполнения алгоритма возможна передача управления сразу нескольким процессам, что выражается в соответствующем срабатывании переходов и появлении соответствующих фишек в нескольких позициях сети.

Переход по семантическим связям в РСПСС является математическим объектом, устанавливающим семантические ограничения на последовательность смены состояний одного или нескольких элементов системы. Наличие перехода по семантическим связям между двумя позициями a_i и a_j указывает на то, что для выполнения операции, представленной в позиции a_j необходимо наличие информации, порождаемой в позиции a_i . Например, a_i — присвоение значения в переменную x , a_j — считывание значения переменной x .

Графически позиции сети представляются кругами, переходы по управлению — утолщёнными линиями, переходы по семантическим связям — треугольниками. Пример РСПСС для последовательного линейного алгоритма приведен на рисунке 1. Элементы представленной модели можно интерпретировать следующим образом: позиции 0 и 1 можно расценивать как память (например, переменные), хранящую некоторые значения, позиции 2 и 3 — чтение значения из памяти

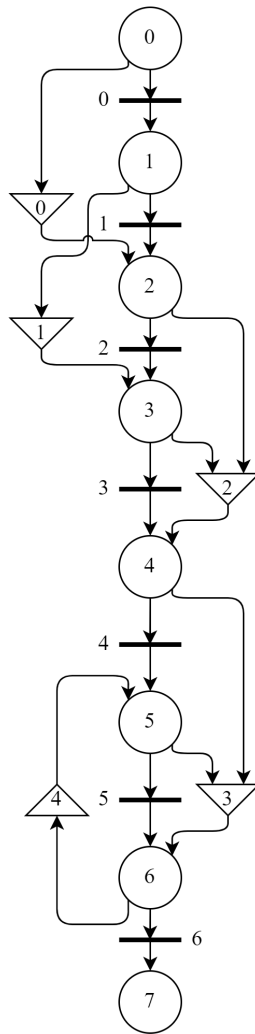


Fig. 1. ExPNR of sequential linear algorithm

Рис. 1. PCПСС последовательного линейного алгоритма

в позициях 0 и 1 соответственно, позиция 3 — выполнение операции со значениями из 0 и 1, позиция 4 — выполнение некоторой операции со считанными значениями, позиция 5 — некоторая память, отличная от ассоциированной с позициями 0 и 1, позиция 6 — запись результатов операции в позиции 4 в память на позиции 5, позиция 7 — терминатор алгоритма. Таким образом, переход по семантическим связям 0 и 1 показывают чтение конкретной памяти, переход по семантическим связям 2 — использование считанных значений в операции, переход по семантическим связям 3 — использование результата операции для записи в конкретную память, а переход по семантическим связям 4 — зависимость памяти от операции записи. Так как алгоритм линейный, то все позиции последовательно соединены переходами по управлению.

Этому может соответствовать следующая последовательность операций.

Начало;
 $a = 10$;
 $b = 5$;
 $c = a + b$;
 Конец.

Преимуществом такой модели является разделение потока управления и семантических связей. В отличие от классических методов на основе графов операции-операнды, где невозможно четко различить разные ветви условий и параллельные процессы, здесь такое отличие становится очевидным. Кроме того, четко определяются параллельные области, можно варьировать число потоков и анализировать балансировку нагрузки между параллельными потоками. Это позволяет более простыми методами анализировать и проектировать параллельные алгоритмы [28].

Для анализа модели часто необходимо анализировать входные ($I_A(z^S), I_A(z^C)$), описывающие множество позиций, отображаемых в данный переход, и выходные ($O_A(z^S), O_A(z^C)$), описывающие множество позиций, отображаемых из данного перехода, функции переходов.

В РСПСС возможны следующие типы переходов в зависимости от мощности соответствующих множеств $I_A(z)$ и $O_A(z)$: — примитивные переходы, которые имеют одну входную и одну выходную позицию, то есть $|I_A(z_i)| = |O_A(z_i)| = 1$. Так как примитивным может быть любой тип перехода, то его в формуле не специфицируем. Непримитивные переходы по управлению:

- fork, $|I_A(z^C)| = 1, |O_A(z^C)| > 1$, представляющий начало параллельного вычислительного процесса, когда из одного потока/процесса порождается несколько;
- join, $|I_A(z^C)| > 1, |O_A(z^C)| = 1$, представляющий завершение параллельного вычислительного процесса, когда несколько параллельных потоков/процессов завершаются и далее продолжает работу только один;
- synchro, $|I_A(z^C)| = n, |O_A(z^C)| = m, n, m > 1$, представляющий синхронизацию, когда несколько потоков/процессов ожидают завершения определённых операций друг от друга для продолжения своей работы. В общем случае какие-то потоки или процессы могут полностью завершить своё выполнение, при этом $m \neq n$.

Для переходов по семантическим связям возможен только один непримитивный переход s-join, $|I_A(z^S)| > 1, |O_A(z^S)| = 1$, представляющий объединение информации от разных операций для выполнения следующей.

Пример РСПСС для параллельного алгоритма приведён на рисунке 2. По аналогии с РСПСС последовательного алгоритма, здесь также позицию 0 можно расценивать как память, хранящую некоторое значение, позиция 2 — некоторая операция, для вычисления которой не используются сторонние значения, позиции 3 и 5 — операции записи значений в память в позиции 0 (позиция 5 в таком виде может означать запись константы, так как не зависит от какой-либо операции вычисления значений), позиция 8 — чтение значения из памяти в позиции 0, позиция 9 — память отличная от ассоциированной с позицией 0, позиция 10 — запись в память на позиции 9 считанного в позиции 8 значения из позиции 0, позиции 4 и 6 — терминаторы функций соответствующих потоков, позиция 9 — терминатор основной функции. Так как алгоритм параллельный то в нём присутствуют специальные позиции: позиция 1 — порождение параллельного участка, и позиция 7 — объединение параллельных путей алгоритма в один. Помимо примитивных переходов по управлению 0, 2–4, 6–9 в модели присутствуют переход типа Fork под номером 1 и переход типа Join под номером 5.

Этому может соответствовать следующий псевдокод.

```
Начало;
a = 10;
создание потоков 1 и 2;
поток 1:
    Начало;
    a = foo();
    Конец.
поток 2:
    Начало;
```

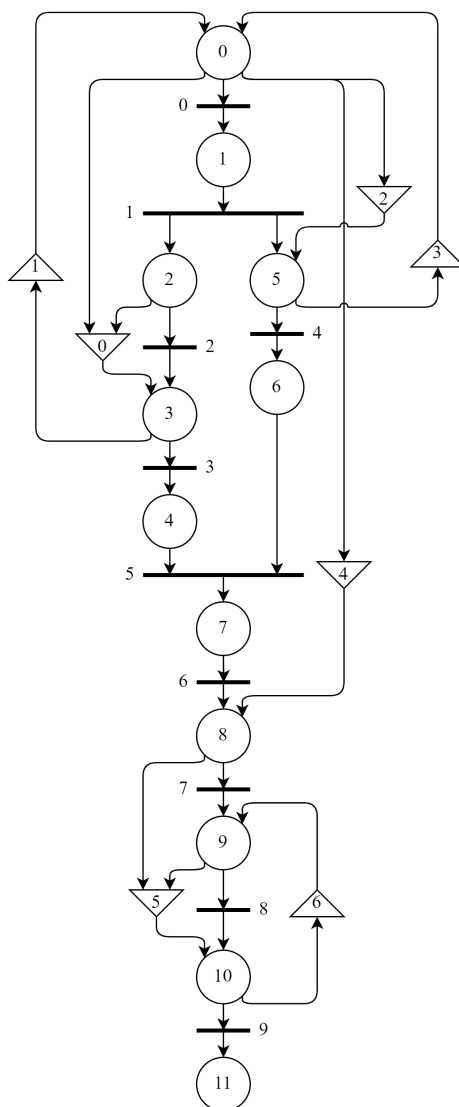


Fig. 2. ExPNRS of concurrent algorithm

Рис. 2. РСПСС для параллельного алгоритма

$a = 10;$

Конец.

ожидание завершение потоков 1 и 2;

$b = a;$

Конец.

Каждая позиция РСПСС имеет не более одной дуги, связывающей её с любым переходом. При этом из позиции может выходить несколько дуг, ведущих к разным переходам в случае ветвления, и входить несколько дуг при завершении альтернативных путей алгоритма.

Наличие двух типов переходов приводит к изменению понятия фишки. В классической теории сетей Петри фишка является атомарной структурой. Однако в ряде работ (например, [29]) фишки могут сами представлять собой сложные объекты с сетевой структурой. В РСПСС используется составная фишка T , разделяемой на управляющую фишку (множество фишек) — T^C и семантическую фишку (множество фишек) — T^S . Каждый из данных подтипов фишки формирует поведение соот-

ветствующего типа переходов, объединяясь в позициях в единую фишку $T = T^S + T^C$. Рассмотрим поведение каждого типа фишек в процессе функционирования сети.

Прежде всего следует отметить, что при моделировании вычислений на ЭВМ в любой позиции сети в любой момент времени не может быть более одной объединённой фишки. Наличие иной ситуации свидетельствовало бы, что в одно и то же время на одном исполнительном устройстве (процессоре, ядре) проводится несколько одинаковых вычислений, что не имеет смысла. В системе может быть более одной позиции с фишкой только в том случае, если эти позиции принадлежат разным параллельным путям, т. е. их пути по управлению от начальной позиции сети имеют больше общих переходов типа `fork`, чем общих переходов типа `join`.

Для срабатывания перехода по управлению необходимо, чтобы во всех входных позициях перехода было по одной объединённой фишке, если иное не установлено в векторе условий срабатывания перехода Λ^C . При срабатывании управляющая фишка извлекается из соответствующих позиций и во все выходные позиции перехода добавляется управляющая фишка. В выходных позициях перехода должна уже находиться или появиться в этом же такте работы сети семантическая фишка. При наличии в позиции семантической и управляющей фишки они соединяются в единую фишку. Дополнительные условия срабатывания могут накладываться на переход по управлению при использовании при распараллеливании конструкций типа `nowait`, то есть, когда потоки завершаются и достаточно одного завершённого потока для продолжения работы, остальные в это время могут продолжать свое исполнение. В этом случае, наличия хотя бы одной объединённой фишки во входных позициях перехода `join` достаточно для его срабатывания. После первого срабатывания перехода, он работает только на извлечение фишек из входных позиций, без их добавления во входную.

При разделении объединённой фишки создается всегда одна управляющая фишка и столько семантических фишек, в скольких переходах по семантическим связям во входной функции находится данная позиция. Для конечных вычислительных операций, данные от которых больше нигде не используются, может быть не создано ни одной семантической фишки. Переходы по семантическим связям требуют, чтобы во всех входных позициях перехода была семантическая фишка, если иное не установлено в векторе условий срабатывания перехода Λ^S . Однако, существует дополнительное условие к семантической фишке. Она должна непременно быть результатом расщепления объединённой фишки, проверка чего осуществляется путем активации соответствующей дуги к переходу по семантическим связям только после срабатывания перехода по управлению, для которого эта же позиция является входной. Можно говорить о блокировке фишки, когда переход z_k^S не находится в возбуждённом состоянии, но на некоторых из его входных позиций $\exists a_j \in I_A(z_k^S)$ имеются фишки. Такой переход z_k^S называется блокирующим. Блокированная фишка не может участвовать во взаимодействии ни с какими другими переходами по семантическим связям кроме блокирующего. В единственную выходную позицию перехода после срабатывания помещается семантическая фишка, требующая объединения. Наличие дополнительных условий срабатывания перехода вызвано тем, что любая позиция не может принадлежать выходной функции более чем одного перехода по семантическим связям. Тогда при наличии циклических конструкций или ветвлений необходимо дополнительное условие «исключающего или» для ряда входных позиций.

Формирование и разделение объединённой фишки является больше абстрактным процессом. Каждый тип переходов изымает из вершины фишку исключительно своего типа и не пропускает «чужие» фишки.

Такая модель может эффективно использоваться как для моделирования, так и для верификации параллельных программ. В части верификации можно обнаружить следующие ошибки:

- некорректная последовательность действий, когда в позицию сперва попадает управляющая фишка;

- наличие гонок данных, то есть ситуаций, когда два или более параллельно работающих потока обращаются для записи к одной и той же переменной без использования синхронизации;
- наличие взаимных блокировок потоков, то есть ситуаций, когда потоки, получив эксклюзивные права на продолжение работы путем захвата блокировки, не могут её продолжить в связи с невозможностью получить следующую блокировку, захваченную другим потоком.

Анализ гонок данных в рамках РСПСС подразумевает проверку наличия семантических связей между параллельными участками сети при отсутствии синхронизации доступа между ними, который можно определить путем анализа связей по управлению. Определить это можно по наличию нескольких семантических зависимостей, где общая переменная является выходом (запись) или входом (чтение), а места её использования (входы и выходы переходов соответственно) относятся к параллельным участкам. При этом следует исключать из анализа переход z_i^S , где позиция a_i является входом, если его выход a_j является входом для перехода z_j^S , выходом которого является a_i , так как в данном случае переход z_i^S показывает не чтение, а использование памяти в операции записи. Наглядно это можно рассмотреть на примере позиции 0 рисунке 2, где семантические переходы 0 и 2 показывают использование памяти, с которой ассоциируется данная позиция, в операциях записи в неё, а семантические переходы 1 и 3 — зависимость памяти от операций записи в неё, и, так как позиции 3 и 5, являющиеся входами семантических переходов 1 и 3 соответственно, относятся к параллельным участкам, можно утверждать о наличии в модели гонки данных типа «запись — запись».

Для анализа взаимных блокировок, необходимо оценить достижимость соответствующей разметки сети, отображающей успешное выполнение этого параллельного участка, в случае наличия хотя бы одного пути на графе достижимости, который не приводит к желаемой разметке, определяется возможность возникновения взаимной блокировки (тупика). Более подробное рассмотрение этих задач не является целью данной статьи. Здесь хотелось бы обратить внимание, что в обоих случаях необходимым является анализ используемых механизмов синхронизации для обеспечения корректного доступа к переменным и вычислительным ресурсам системы. В связи с множеством различных используемых в разных технологиях распараллеливания механизмов синхронизации, необходимо определить их основные особенности и такие способы их корректного представления в РСПСС, которые позволят в дальнейшем автоматически строить их на основе кода параллельной программы и анализировать созданную сеть на предмет ошибок. В других работах, представляющих методы верификации параллельных программ на основе других вариантов сетей Петри, рассматривается лишь ограниченный набор примитивов синхронизации. Например, в работе [27] представлены только атомарные операции и мьютексы, в работе [30] — только условные переменные. Нашей целью является обобщение информации по всем основным способам синхронизации и представление их в виде нашей версии сетей Петри.

3. Основные типы операций синхронизации в параллельных программах

Для решения различных задач обеспечения взаимоисключающего доступа к общим данным в параллельных программах используют следующие основные методы синхронизации:

Атомарные операции — это такие операции, которые выполняются за один раз либо вообще не выполняются. Таким образом, если атомарная операция была запущена, она гарантирует, что дойдет до конца исполнения. В ряде языков программирования существуют средства задания и/или использования атомарных операций (например, класс `Interlocked` в .Net или директива `atomic` в OpenMP).

Барьер — это такой метод синхронизации, когда каждый поток из группы параллельных потоков, доходя до определённой точки программы (барьера), должен приостановить свою работу до тех пор, пока вся группа не достигнет заданной точки. Зачастую барьеры устанавливаются, когда необ-

ходимо получить промежуточные результаты вычислений, на основе которых будут выполняться следующие за барьером вычисления.

Критические секции представляют объекты синхронизации потоков, позволяющий предотвратить одновременное выполнение некоторого набора операций несколькими потоками. В один момент времени только один поток может исполнять код, принадлежащий соответствующей критической секции, все другие потоки, желающие попасть в критическую секцию, блокируются. Во многих технологиях параллельного программирования критические секции оформляются с использованием команд входа и выхода из критической секции (например, `EnterCriticalSection`, `LeaveCriticalSection` в WinAPI). Критические секции могут быть именованные, и в этом случае вхождение в одну критическую секцию, не оказывает влияние на доступ к другим.

Мьютекс — примитив синхронизации, обеспечивающий взаимное исключение критических участков кода. Несмотря на большое количество типов мьютексов в различных технологиях параллельного программирования (например, `recursive_mutex`, `timed_mutex` в `std::thread`, `spin_mutex`, `queuing_mutex` и т. д. в oneAPI TBB), каждый из которых обладает своими особенностями, принципиальный подход к их моделированию будет схож, но в некоторых случаях будет меняться порядок доступа к критическому коду потоков. Поэтому в данной статье будем рассматривать только классический мьютекс. Классический мьютекс представляется в виде переменной, которая может находиться в двух состояниях: в заблокированном и в незаблокированном. Когда нам нужно обеспечить взаимное исключение для некоторого кода потока, он вызывает блокирующую функцию мьютекса. Другие потоки, пытающиеся заблокировать тот же мьютекс, блокируются до тех пор, пока мьютекс не будет освобожден потоком, которому он принадлежит. После выполнения критического кода поток вызывает функцию для перевода мьютекса в разблокированное состояние и в то же время заблокированный этим мьютексом поток может продолжить свои вычисления. Если таких потоков много, планировщик выбирает один из них для возобновления выполнения.

Семафор является примитивом синхронизации, который позволяет ограничить количество потоков, которые имеют доступ к определённым ресурсам. Семафор представляет собой целую переменную, принимающую неотрицательные значения. Доступ к этой переменной осуществляется через две атомарные операции $P(S)$: пока $S \neq 0$ процесс блокируется; $S = S - 1$ и $V(S)$: $S = S + 1$. Во время операции P , если значение семафора больше нуля, то оно уменьшается на единицу, после чего можно получить доступ к критическому коду. Когда $S = 0$, поток блокируется до тех пор, пока какой-либо другой поток не увеличит значение S . Семафор может быть инициализирован начальным значением. Оно может быть равно нулю, что изначально запрещает доступ к любым данным, пока один из потоков не выполнит и не вызовет операцию V , или Любому другому неотрицательному значению N , что разрешает доступ к соответствующим данным несколько раз даже без выполнения операции V .

Условные переменные представляют примитив синхронизации, который обеспечивает блокировку одного или нескольких потоков до момента поступления сигнала от другого потока о выполнении некоторого условия или до истечения максимального промежутка времени ожидания. При этом, часто данные переменные дополнительно защищаются блокировками. При работе с условными переменными используются методы `wait` — для блокировки потока в ожидании условия, и метод `signal` (`notify` в `std::thread`) для информирования ожидающих потоков о срабатывании условия. В зависимости от реализации после операции `signal` может быть разблокирован один или несколько потоков.

Мониторы представляют собой высокоуровневый механизм взаимодействия и синхронизации процессов. Они представляют специальный тип данных, аналогичный классу или объекту в ООП, служащий для разграничения доступа к критическому коду программы. Монитор обладает собственными переменными, определяющими его состояния, а также функциями-методами, которые

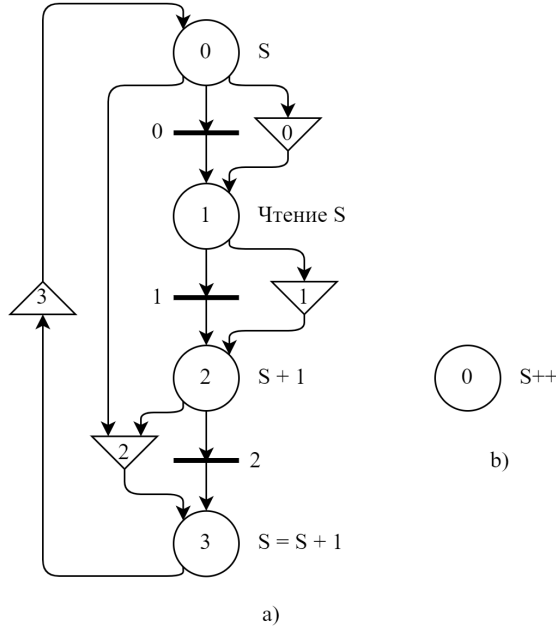


Fig. 3. Increment operation: a — non-atomic; b — atomic

Рис. 3. Операция инкремента: а — неатомарная; б — атомарная

могут изменять значения этих переменных и могут использовать только данные, находящиеся внутри монитора и свои параметры. Особенностью мониторов является то, что только один процесс может быть активен внутри данного монитора в любой момент времени. Фактически, с точки зрения задач моделирования мониторы представляют собой комбинацию уже рассмотренных выше примитивов синхронизации, а именно: мьютексов и условных переменных. Поэтому, в данной статье не будем останавливаться на них подробно.

4. Представление операций синхронизации в расширенных сетях Петри с семантическими связями

В этом разделе описывается представление в РСПСС вышеупомянутых примитивов синхронизации.

Атомарные операции. Простейшим примитивом является атомарная операция. Атомарная операция представляется как объединение нескольких операций в одной позиции РСПСС. Например, рассмотрим операцию увеличения семафорного счётчика $V(S) : S = S + 1$. Если бы она не была атомарной, то была бы представлена тремя позициями РСПСС (часть а на рисунке 3), но из-за атомарности это одна позиция (часть б на рисунке 3).

Для представления атомарной операции в первую очередь необходимо иметь РСПСС для программного кода без синхронизации, а затем реорганизовать его. Все переходы по семантическим связям, для которых изначальные позиции $A_{comb} = \{a_{comb_i}\}$, не объединённые еще в атомарную операцию, были выходными, объединяются в один переход. Множество входных позиций $I_A^S(z_{new}^S)$ нового перехода по семантическим связям z_{new}^S получается объединением множеств входных позиций объединяемых переходов $I_A^S(z_{old_i}^S)$ за исключением позиций, вошедших в новую атомарную, а выходная функция нового перехода $O_A^S(z_{new}^S)$ является позицией, представляющей атомарную операцию:

$$I_A^S(z_{new}^S) = \bigcup_i I_A^S(z_{old_i}^S) \setminus A_{comb}, O_A^S(z_{new}^S) = a_{atomic}.$$

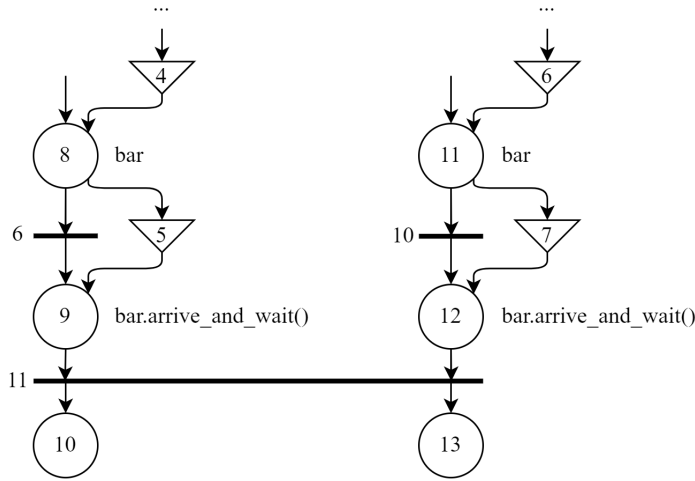


Fig. 4. Example of the barrier

Рис. 4. Пример барьерной синхронизации

Если какая-либо позиция a_i из множества A_{comb} являлась входной позицией любого иного перехода по семантическим связям, т. е. $I_A^S(z_j^S)$, то она должна быть заменена на новую атомарную позицию a_{atomic} .

$$I_{A_j}^S = \begin{cases} I_A^S(z_j^S), & \text{если } A_{comb} \cap I_A^S = \emptyset, \\ I_A^S(z_j^S) \cup \{a_{atomic}\} \setminus A_{comb}, & \text{иначе.} \end{cases}$$

Переходы по управлению между объединяемыми позициями должны быть удалены из сети. Мы сохраняем только переходы Z_{input}^C , которые вели в первую объединяемую позицию, и Z_{output}^C , которые имели последнюю объединяемую позицию в своей выходной функции. Больше ничего в РСПСС изменять не следует.

Барьеры в РСПСС представлены непримитивным переходом по управлению типа *synchro*. Этот тип переходов характеризуется тем, что он может иметь несколько входных и несколько выходных позиций. Для моделирования барьера необходимо определить точку в каждом потоке, после чего будет активирован режим ожидания всех потоков, которые соединены этим барьером. Эта точка представляет собой операцию вызова барьера на соответствующем языке программирования. Все такие позиции $a_{barrier_i}$ представляют входную функцию перехода *synchro* $I_{A_synchro}^C$. Выходная функция перехода *synchro* $O_{A_synchro}^C$ будет представлена позициями в каждом синхронизированном потоке, следующими за функцией барьера:

$$I_{A_synchro}^C = \bigcup_i \{a_{barrier_i}\}, O_{A_synchro}^C == \bigcup_i \{a_{barrier_{i_next}}\}.$$

Семантические отношения не меняются при барьерной синхронизации. Пример барьерной синхронизации представлен на рисунке 4.

Как видно, имеются два потока, имеющие доступ к одному и тому же барьеру *bar* (позиции 8 и 11 на рисунке 4 – чтение памяти, хранящей барьер). Они вызывают функцию *arrive_and_wait* (позиции 9 и 12 на рисунке 4), чтобы остановить поток, пока другой не достигнет этой точки, после чего потоки продолжат свои вычисления (позиции 10 и 13 на рисунке 4).

Мьютекс – это особая позиция в РСПСС, отражающая состояние счётчика мьютекса (позиция *lock* на рисунке 5). Мьютекс не содержит фишку в позиции в заблокированном состоянии и содержит одну в разблокированном состоянии. Позиция счётчика является входом для переходов

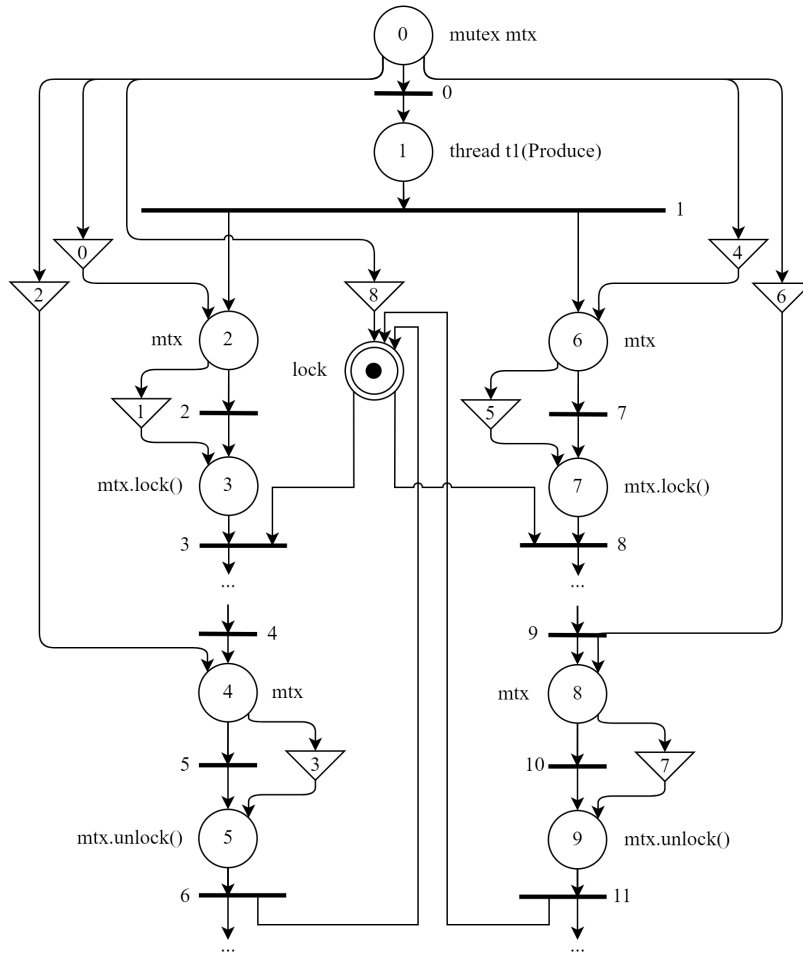


Fig. 5. Mutex representation in ExPNRS

Рис. 5. Представление мьютекса в РСПСС

по управлению, для которых другой позицией входа является операция попытки блокировки мьютекса (позиции 3 и 7 на рисунке 5) в соответствующем потоке. Поскольку в позиции счётчика мьютекса может быть только одна фишка, только один поток может продолжать свое выполнение. Переход по управлению после позиции освобождения мьютекса (позиции 5 и 9 на рисунке 5) имеет позицию счётчика в качестве одной из своих выходных позиций, что позволяет перевести его в разблокированное состояние и позволить другому потоку получить доступ к критичному коду.

Семантические связи внутри значимых операций не меняются. Позиция счётчика семантически зависит от позиции инициализации мьютекса (позиция 0, переход по семантическим связям 8 на рисунке 5). От одной и той же позиции инициализации зависят операции чтения состояния мьютекса (позиции 2, 4, 6 и 8 на рисунке 5 и переходы по семантическим связям 0, 2, 4, 6 соответственно) перед изменением его состояния.

Критические секции представляются аналогично мьютексам. Для каждой критической секции создается своя позиция, указывающая, находится ли какой-либо процесс в её критической секции или нет. Эта позиция идентична позиции счётчика мьютекса. Как и в случае с мьютексами, фишка в позиции означает, что поток может попасть в свою критическую секцию, а её отсутствие блокирует все остальные потоки. В этом случае позицией, управляющей изменением состояния критической секции, будет операция EnterCriticalSection как аналог операции блокировки мьютекса и LeaveCriticalSection как аналог операции разблокировки (освобождения) мьютекса. Операция

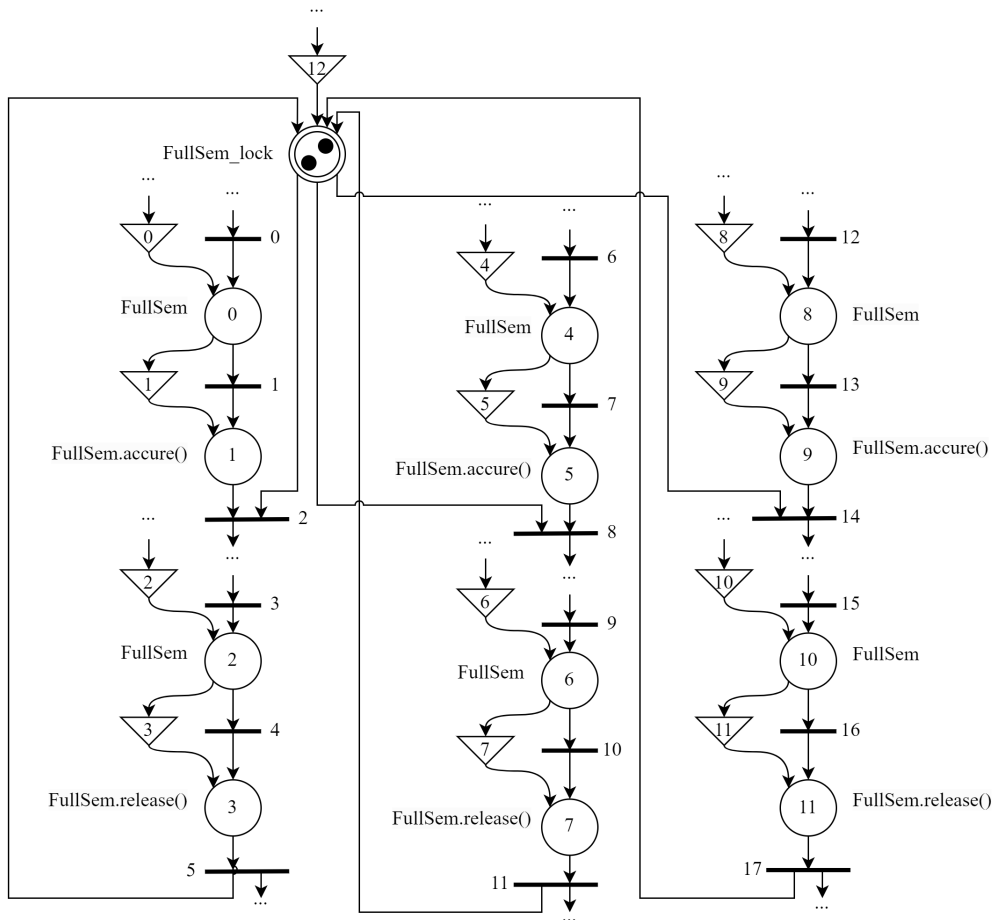


Fig. 6. Semaphore with several tokens in counter place in ExPNR

Рис. 6. Семафор с несколькими фишками в позиции счётчика в РСПСС

InitializeCriticalSection играет ту же роль, что и инициализация мьютекса, и представлена в РСПСС таким же образом. Из-за того, что представление критической секции и мьютекса отличается только названием операций, мы не будем приводить в этой статье какой-либо дополнительный рисунок в качестве примера критической секции.

Моделирование **семафоров** также аналогично представлению мьютексов. Отличие заключается только в количестве фишек, которые могут быть в позиции семафора. Семафор, таким образом, является единственным вариантом позиции в РСПСС, когда нарушается правило наличия не более чем одной фишки в позиции. Такое возможно для теории РСПСС, так как в данном случае позиция не представляет операцию, а только регулирует запуск параллельных потоков. Таким образом, может быть сформулировано дополнительное правило для проверки модели параллельной программы: если в позиции оказалось более одной фишки, и эта позиция не является позицией семафора, то при моделировании программы была допущена ошибка. Само значение семафорной переменной S будет представлено в сети фишками в позиции счётчика (позиция FullSem_lock на рисунке 6). При этом при формировании исходной разметки РСПСС на месте счётчика сохраняется количество меток, равное начальному значению семафора.

Кроме того, значение счётчика семафора изначально может быть равно 0, что соответствует отсутствию фишки в позиции счётчика (позиция EmptySem_lock на рисунке 7). Это возможно,

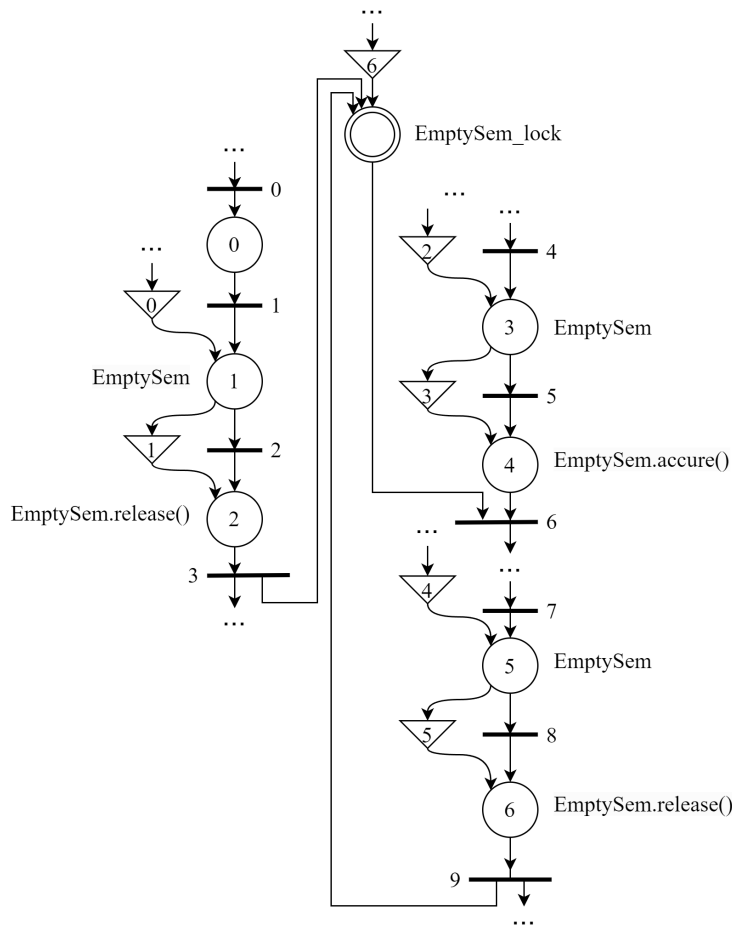


Fig. 7. Semaphore without tokens in counter place in ExPNSR

Рис. 7. Семафор без меток в позиции счётчика в PCPCC

потому что, в отличие от мьютекса, который должен быть разблокирован тем же потоком, который его заблокировал, операции с семафором могут выполняться независимо в разных потоках.

Для реализации некоторых сценариев работы программы может потребоваться, чтобы один из потоков дождался наступления определённых условий. Чтобы избежать постоянных проверок, поток можно приостановить до получения специального сигнала с помощью **условной переменной**. Один из потоков, использующий условную переменную, блокирует себя до тех пор, пока другой поток не освободит её (рисунок 8).

В зависимости от используемой технологии параллельного программирования могут быть небольшие особенности моделирования условных переменных. Мы рассмотрим реализацию, предоставляемую технологией `std::thread`. Обычно для целей блокировки используется `unique_lock` (позиции 0 и 6 на рисунке 8), который управляет блокировкой и разблокировкой мьютекса, связанного с условной переменной. Если какое-то условие (позиция 10 на рисунке 8) не выполняется, мьютекс разблокируется и вызывается метод ожидания условной переменной (позиция 13 на рисунке 8). Он блокирует текущий поток до тех пор, пока другой поток не вызовет метод `notify_one` (позиция 3 на рисунке 8), который отправит сигнал одному из ожидающих потоков (позиция 14 на рисунке 8). После этого текущий поток снова блокирует мьютекс и проверяет состояние.

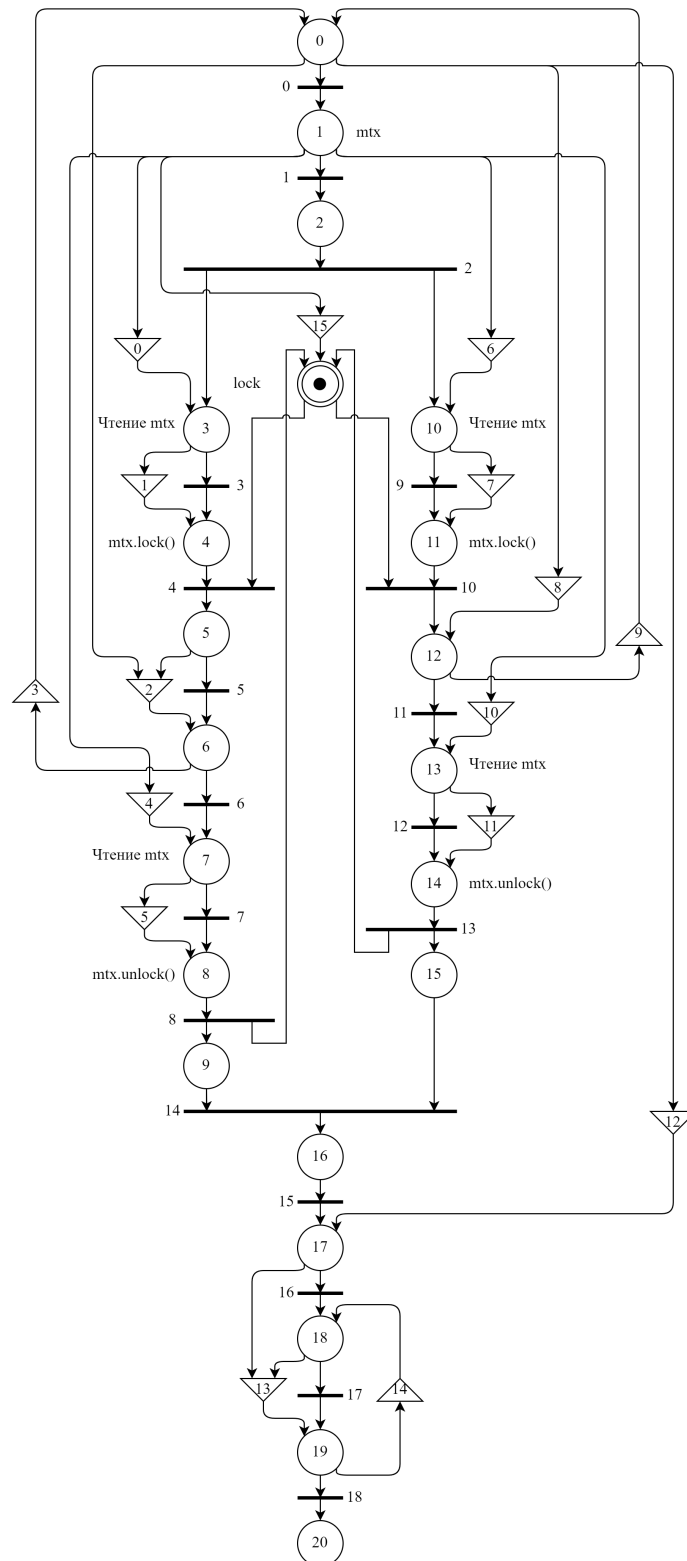


Fig. 9. ExpNSR of concurrent algorithm without data race

Рис. 9. РСПСС для параллельного алгоритма без гонки данных

Так как позиции 6 и 12, аналогичные позициям 3 и 5 рисунка 2 соответственно, находятся в критических секциях, образованных операциями блокировки мьютекса: позиции 4 и 11; соответствующие им операции будут выполняться последовательно, а значит гонка данных отсутствует.

Следует дополнительно отметить, что различные технологии распараллеливания могут иметь свои особенности реализации. Однако эти особенности не изменяют принципов их моделирования, так как относятся к исполнению, а не к фундаментальной логике, которая проверяется на наличие ошибок. Эти характеристики лишь в некоторых случаях могут привести к добавлению каких-то позиций в РСПСС, или добавлению логических условий срабатывания переходов (например, для некоторых типов мьютексов).

Следует также отметить появление новых схем параллельной обработки, внедряемых в существующие языки (например, JEP 436, 437 для Java 20–21, Concurrency и Coroutines в C++ 20 и 23). Они требуют проведения статической верификации на более низком, системном уровне, что означает более сложный процесс обнаружения ошибок, стоимость которых при этом гораздо выше. В рамках реализации статического детектора тупиков и гонок данных для конкретных языков программирования и технологий распараллеливания это будет учтено в нашей дальнейшей работе.

Как уже говорилось ранее, такое представление программного кода в виде модели РСПСС позволяет автоматически формировать сеть, а также в дальнейшем её анализировать. Рассмотрим кратко как это осуществляется для языка Go. Построение модели идёт на основе уже имеющегося более простого, предлагаемого языком представления кода в виде абстрактного синтаксического дерева. Оно представляет собой конечное помеченное дерево, вершины которого соответствуют операторам языка, а листья — операндам (идентификаторам или константам), а основными элементами являются функции. Каждое объявление функции приводит к формированию новой РСПСС. Для каждой функции неявно создается позиция терминатора и переход по управлению из позиции, представляющей последнюю операцию данной функции, в позицию терминатора. Далее обрабатываются аргументы функции, для которых создаются соответствующие позиции сети. После чего происходит обработка операторов данной функции с учётом ключевых слов, позволяющих определить выделение памяти под переменные, обработку выражений, запуск нового потока, использование примитивов синхронизации и т. д. Для каждой операции формируются соответствующие позиции сети и в зависимости от их порядка следования в программе добавляются переходы по управлению. В случае запуска/синхронизации/завершения потока могут добавляться непримитивные переходы по управлению. После добавления новых позиций производится анализ, от каких данных зависит данная операция, находятся соответствующие позиции, порождающие эти данные, находящиеся до текущей позиции или в одном с ней параллельном блоке, и добавляются переходы по семантическим связям. Основным примитивом синхронизации в Go является мьютекс. В связи с наличием нескольких вариантов мьютексов, а также нескольких вариантов их объявления в программе разбор несколько усложняется, однако все равно позволяет найти такие объявления и сформировать в сети особую позицию мьютекса с предварительно оставленной фишкой в ней. Эта позиция сети связывается с позициями, определяющими операции блокировки и разблокировки мьютекса соответствующими переходами по управлению. Более подробно этот подход рассмотрен в работе [31].

Заключение

В статье были рассмотрены подходы к верификации параллельных программ, в частности поиск гонок данных и взаимных блокировок. Отмечено, что полное представление о наличии таких ошибок в программах могут дать только статические методы верификации. Поэтому авторами было решено использовать метод проверки модели, для которого предложено использовать собственную модификацию сетей Петри — расширенную сеть Петри с семантическими связями. Проблемы анализа параллельных программ на основе моделей заключаются в сложности их построения. В ранних работах было показано, как моделируются с использованием РСПСС последовательные алгоритмы,

в том числе с линейными, циклическими структурами и ветвлениями, а также параллельные программы с простейшим примитивом синхронизации — барьером. В данной статье подробно рассмотрены особенности моделирования различных используемых в наиболее часто применяемых технологиях параллельного программирования примитивов синхронизации. Предложенные в статье методы их моделирования помогут в дальнейшем в автоматизации представления параллельных программ в виде РСПСС и их анализа на наличие ошибок организации доступа к общим ресурсам. Дальнейшая работа направлена на обнаружение взаимоблокировок и гонок данных на основе анализа графа достижимости и семантических отношений в РСПСС.

References

- [1] K. N. Zainidinov and Z. A. Karshiev, “Features of parallel execution of data mining algorithms”, *Automatics and Software Enginery*, vol. 31, no. 1, pp. 83–91, 2020.
- [2] C. Atilgan, B. T. Tezel, and E. Nasiboglu, “Efficient implementation and parallelization of fuzzy density based clustering”, *Information Sciences*, vol. 575, pp. 454–467, 2021. DOI: [10.1016/j.ins.2021.06.044](https://doi.org/10.1016/j.ins.2021.06.044).
- [3] A. Krizhevsky, *One weird trick for parallelizing convolutional neural networks*, 2014. arXiv: [1404.5997](https://arxiv.org/abs/1404.5997) [[cs.NE](#)].
- [4] B. Blum and G. Gibson, “Stateless model checking with data-race preemption points”, in *Proceedings of the ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Association for Computing Machinery, 2016, pp. 477–493. DOI: [10.1145/2983990.2984036](https://doi.org/10.1145/2983990.2984036).
- [5] A. N. Ivutin, A. G. Voloshko, and V. N. Izotov, “Approach to data race detection based on petri nets with additional semantic relations”, in *Proceedings of the 2020 ELEKTRO*, 2020, pp. 1–5. DOI: [10.1109/ELEKTRO49696.2020.9130252](https://doi.org/10.1109/ELEKTRO49696.2020.9130252).
- [6] A. N. Ivutin, A. G. Troshina, and D. Yesikov, “Parallelization of algorithms with use the semantic Petri-Markov nets”, *Vestnik of Ryazan State Radio Engineering University*, vol. 58, no. 4, pp. 49–56, 2016, in Russian. DOI: [10.21667/1995-4565-2016-58-4-49-56](https://doi.org/10.21667/1995-4565-2016-58-4-49-56).
- [7] A. V. Soloviev and D. A. Sedov, “Algorithms of detecting errors in parallel programs for distributed memory systems”, in *Proceedings of the Institute for Systems Analysis Russian Academy of Sciences (ISA RAS)*, in Russian, vol. 63, 2013, pp. 9–15.
- [8] S. Abbaspour Asadollah, D. Sundmark, S. Eldh, and H. Hansson, “Concurrency bugs in open source software: A case study”, *Journal of Internet Services and Applications*, vol. 8, no. 1, pp. 4–19, 2017. DOI: [10.1186/s13174-017-0055-2](https://doi.org/10.1186/s13174-017-0055-2).
- [9] I. Tamas, I. Salomie, and M. Antal, “Atomic invariants verification and deadlock detection at compile-time”, in *Proceedings of the IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2018, pp. 435–441.
- [10] Y. Cai, C. Ye, Q. Shi, and C. Zhang, “Peahen: Fast and precise static deadlock detection via context reduction”, in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 784–796. DOI: [10.1145/3540250.3549110](https://doi.org/10.1145/3540250.3549110).
- [11] M. Ronsse and K. De Bosschere, “RecPlay: A fully integrated practical record/replay system”, *ACM Transactions on Computer Systems*, vol. 17, no. 2, pp. 133–152, 1999. DOI: [10.1145/312203.312214](https://doi.org/10.1145/312203.312214).
- [12] C. Flanagan and S. N. Freund, “FastTrack: Efficient and precise dynamic race detection”, in *Proceedings of the 30th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2009, pp. 121–133. DOI: [10.1145/1542476.1542490](https://doi.org/10.1145/1542476.1542490).

- [13] E. Pozniansky and A. Schuster, “Efficient on-the-fly data race detection in multithreaded C++ programs”, in *Proceedings of the Ninth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2003, pp. 179–190. DOI: [10.1145/781498.781529](https://doi.org/10.1145/781498.781529).
- [14] K. Serebryany and T. Iskhodzhanov, “ThreadSanitizer: Data race detection in practice”, in *Proceedings of the Workshop on Binary Instrumentation and Applications*, 2009, pp. 62–71. DOI: [10.1145/1791194.1791203](https://doi.org/10.1145/1791194.1791203).
- [15] S. Savage, M. Burrows, G. Nelson, P. Sobalvarro, and T. Anderson, “Eraser: A dynamic data race detector for multithreaded programs”, *ACM Transactions on Computer Systems*, vol. 15, no. 4, pp. 391–411, 1997. DOI: [10.1145/265924.265927](https://doi.org/10.1145/265924.265927).
- [16] X. Xie and J. Xue, “Acculock: Accurate and efficient detection of data races”, in *Proceedings of the International Symposium on Code Generation and Optimization*, 2011, pp. 201–212. DOI: [10.1109/CGO.2011.5764688](https://doi.org/10.1109/CGO.2011.5764688).
- [17] M. Yu and D.-H. Bae, “SimpleLock+: Fast and accurate hybrid data race detection”, *The Computer Journal*, vol. 59, no. 6, pp. 793–809, 2016. DOI: [10.1093/comjnl/bxu119](https://doi.org/10.1093/comjnl/bxu119).
- [18] A. Legalov, V. Vasilyev, I. Matkovskii, and M. Ushakova, “Support tools for creation and transformation of functional-dataflow parallel programs”, *Proceedings of the Institute for System Programming of the RAS*, vol. 29, no. 5, pp. 165–184, 2017. DOI: [10.15514/ISPRAS-2017-29\(5\)-10](https://doi.org/10.15514/ISPRAS-2017-29(5)-10).
- [19] V. Tikhomirov, S. Timofeev, and E. Moshkova, “Modified model checking verification method”, *Bulletin of Russian Academy of Natural Sciences*, no. 3, pp. 118–121, 2018, in Russian.
- [20] E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem, *et al.*, *Handbook of Model Checking*. Springer Cham, 2018, vol. 10.
- [21] J. Barnat *et al.*, “Parallel model checking algorithms for linear-time temporal logic”, in *Handbook of Parallel Constraint Reasoning*. Springer, 2018, pp. 457–507. DOI: [10.1007/978-3-319-63516-3_12](https://doi.org/10.1007/978-3-319-63516-3_12).
- [22] H. Boucheneb, G. Gardey, and O. H. Roux, “TCTL model checking of time Petri nets”, *Journal of Logic and Computation*, vol. 19, no. 6, pp. 1509–1540, 2009. DOI: [10.1093/logcom/exp036](https://doi.org/10.1093/logcom/exp036).
- [23] K. M. Kavi, A. Moshtaghi, and D.-j. Chen, “Modeling multithreaded applications using Petri nets”, *International Journal of Parallel Programming*, vol. 30, pp. 353–371, 2002. DOI: [10.1023/A:1019917329895](https://doi.org/10.1023/A:1019917329895).
- [24] G. Liu, M. Zhou, and C. Jiang, “Petri net models and collaborativeness for parallel processes with resource sharing and message passing”, *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 4, pp. 1–20, 2017. DOI: [10.1145/2810001](https://doi.org/10.1145/2810001).
- [25] S. Bandyopadhyay, D. Sarkar, and C. Mandal, “Equivalence checking of Petri net models of programs using static and dynamic cut-points”, *Acta Informatica*, vol. 56, pp. 321–383, 2019. DOI: [10.1007/s00236-018-0320-2](https://doi.org/10.1007/s00236-018-0320-2).
- [26] A. Albaghajati and M. Ahmed, “CPN.Net: An automated colored Petri nets model extraction from .Net based source code”, in *Proceedings of the 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, 2021, pp. 245–250. DOI: [10.1109/CAIDA51941.2021.9425201](https://doi.org/10.1109/CAIDA51941.2021.9425201).
- [27] M. Westergaard, “Verifying parallel algorithms and programs using coloured petri nets”, in *Transactions on Petri Nets and Other Models of Concurrency VI*. Springer Berlin Heidelberg, 2012, pp. 146–168. DOI: [10.1007/978-3-642-35179-2_7](https://doi.org/10.1007/978-3-642-35179-2_7).
- [28] O. Kryukov, A. Voloshko, and A. Ivutin, “Automation of the formation of the maximum parallelism in the process”, *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskoye nauki*, vol. 10, pp. 230–237, 2022, in Russian.

- [29] I. Lomazova, *Nested Petri nets: Modeling and analysis of distributed systems with object structure*. Moscow: Scientific World, 2003. DOI: [10.13140/2.1.1843.9048](https://doi.org/10.13140/2.1.1843.9048).
- [30] P. d. C. Gomes, D. Gurov, M. Huisman, and C. Artho, “Specification and verification of synchronization with condition variables”, *Science of Computer Programming*, vol. 163, pp. 174–189, 2018. DOI: [10.1016/j.scico.2018.05.001](https://doi.org/10.1016/j.scico.2018.05.001).
- [31] O. Kryukov, A. Voloshko, and A. Ivutin, “Method for automatically building a parallel program model in Go language”, *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskiye nauki*, vol. 2, pp. 420–433, 2025, in Russian.

Detection of Square Wave Impulse Interference in Eddy Current Rail Defectograms

L. Y. Bystrov¹, A. N. Gladkov¹, E. V. Kuzmin¹

DOI: [10.18255/1818-1015-2025-2-172-205](https://doi.org/10.18255/1818-1015-2025-2-172-205)

¹P.G. Demidov Yaroslavl State University, Yaroslavl, Russia

MSC2020: 62P30

Research article

Full text in Russian

Received February 10, 2025

Revised May 23, 2025

Accepted May 28, 2025

Traffic safety in rail transport requires continuous monitoring of the rail condition for timely detection and elimination of defects. One of the methods of non-destructive testing of rails is eddy current flaw detection. The data obtained from eddy current flaw detectors (defectograms) are characterized by a significant volume, which makes it necessary to develop effective methods for their automatic processing and analysis. The analysis of defectograms is complicated by various interferences and noises present in the data. One of the most dangerous interferences that significantly distort the shape of useful signals is prolonged impulse interference. They are characterized by a pronounced square wave shape. Unlike instant impulse interference, prolonged noise cannot be eliminated by classical methods. There are no proven effective methods not only for suppressing square wave interference, but even for detecting it. This article attempts to eliminate this drawback and proposes an effective method for detecting square wave impulse interference on eddy current defectograms, which has good explanatory power. Square signals are explored from the point of view of their probability distribution. SW-characteristic was introduced, which allows to estimate the likelihood of data to the distribution of bipolar impulse signals. The smaller the value of SW-characteristic, the more similar the data distribution is to the distribution of bipolar impulse signals (upon condition that the data are normal). Square wave signals are particular example of bipolar impulse signals. The properties of SW-characteristic were examined. SW-characteristic were calculated for the normal distribution and the distribution of a homoscedastic mixture of two Gaussians. It was shown that the value of SW-characteristic for the normal distribution approximately separates the bimodal mixture of two Gaussians from the unimodal case. These and other properties of SW-characteristic allow using it to detect square wave signals in data by comparing with a threshold, which should satisfy a number of conditions. The application of the criterion based on SW-characteristic was demonstrated on the examples of eddy current defectograms; a comparison was made with criteria based on the EM-algorithm and multiscale disperse entropy. The criterion proposed in the article showed the best results. The use of SW-characteristic for detecting square wave noise has proven its effectiveness in the analysis of eddy current defectograms and can be adapted in the future to work with other types of data.

Keywords: impulse noise; square wave interference; sw-characteristic; eddy current testing; mixture of two gaussians

INFORMATION ABOUT THE AUTHORS

Bystrov, Leonid Y. (corresponding author)	ORCID iD: 0000-0002-0610-5466 . E-mail: l.bystrov@uniyar.ac.ru Assistant of the Department of Theoretical Computer Science
Gladkov, Artemy N.	ORCID iD: 0009-0007-0211-5660 . E-mail: a.gladkov@uniyar.ac.ru Assistant of the Department of Theoretical Computer Science
Kuzmin, Egor V.	ORCID iD: 0000-0003-0500-306X . E-mail: kuzmin@uniyar.ac.ru Head of the Department of Theoretical Computer Science, Dr. Sc.

Funding: Yaroslavl State University (project VIP-016).

For citation: L. Y. Bystrov, A. N. Gladkov, and E. V. Kuzmin, "Detection of square wave impulse interference in eddy current rail defectograms", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 172–205, 2025.

DOI: [10.18255/1818-1015-2025-2-172-205](https://doi.org/10.18255/1818-1015-2025-2-172-205).

Обнаружение прямоугольных импульсных помех на вихретоковых дефектограммах рельсов

Л. Ю. Быстров¹, А. Н. Гладков¹, Е. В. Кузьмин¹DOI: [10.18255/1818-1015-2025-2-172-205](https://doi.org/10.18255/1818-1015-2025-2-172-205)¹Ярославский государственный университет им.П.Г. Демидова, Ярославль, Россия

УДК 519.254

Научная статья

Полный текст на русском языке

Получена 10 февраля 2025 г.

После доработки 23 мая 2025 г.

Принята к публикации 28 мая 2025 г.

Обеспечение безопасности движения на железнодорожном транспорте требует постоянного мониторинга состояния рельсов для своевременного выявления и устранения дефектов. Одним из методов неразрушающего контроля рельсов является вихретоковая дефектоскопия. Данные (дефектограммы), получаемые от вихретоковых дефектоскопов, отличаются значительным объёмом, что делает необходимым разработку эффективных методов их автоматической обработки и анализа. Анализ дефектограмм может быть осложнён присутствием в данных различных помех и шумов. Одними из наиболее опасных помех, существенно искажающих форму полезных сигналов, являются продолжительные импульсные помехи. Они характеризуются выраженной прямоугольной формой. В отличие от мгновенных импульсных помех, продолжительные шумы классическими методами не устраняются. Не существует зарекомендовавших себя эффективных методов не только для подавления прямоугольных помех, но даже для их обнаружения. Данная статья пытается устранить этот недостаток и предлагает действенный метод для обнаружения таких помех на вихретоковых дефектограммах, обладающий хорошей объясняющей способностью. Прямоугольные сигналы исследуются с точки зрения их вероятностного распределения. Введена SW-характеристика, позволяющая оценить правдоподобие данных для распределения биполярных импульсных сигналов. Чем меньше значение SW-характеристики, тем более распределение данных похоже на распределение биполярных импульсных сигналов. Прямоугольные сигналы являются частным случаем биполярных импульсных сигналов. Исследованы свойства SW-характеристики. SW-характеристика вычислена для нормального распределения и распределения гомоскедастичной смеси двух гауссиан. Показано, что значение SW-характеристики нормального распределения примерно разграничивает бимодальную смесь двух гауссиан от унимодального случая. Эти и другие свойства SW-характеристики позволяют использовать её для обнаружения прямоугольных сигналов в данных. Применение критерия на основе SW-характеристики продемонстрировано на реальных примерах вихретоковых дефектограмм, проведено сравнение с критериями на основе ЕМ-алгоритма и многомасштабной дисперсной энтропии. Предложенный в данной статье критерий показал лучшие результаты. Использование SW-характеристики для обнаружения прямоугольного шума доказало свою эффективность на практике при анализе вихретоковых дефектограмм рельсов. Подход может быть адаптирован для работы с другими видами данных.

Ключевые слова: импульсный шум; прямоугольные помехи; sw-характеристика; вихретоковая дефектоскопия; смесь двух гауссиан

ИНФОРМАЦИЯ ОБ АВТОРАХ

Быстров, Леонид Юрьевич
(автор для корреспонденции)

ORCID iD: [0000-0002-0610-5466](https://orcid.org/0000-0002-0610-5466). E-mail: l.bystrov@uniyar.ac.ru
Ассистент, кафедра теоретической информатики

Гладков, Артемий Николаевич

ORCID iD: [0009-0007-0211-5660](https://orcid.org/0009-0007-0211-5660). E-mail: a.gladkov@uniyar.ac.ru
Ассистент, кафедра теоретической информатики

Кузьмин, Егор Владимирович

ORCID iD: [0000-0003-0500-306X](https://orcid.org/0000-0003-0500-306X). E-mail: kuzmin@uniyar.ac.ru
Заведующий кафедрой теоретической информатики, доктор физ.-мат. наук

Финансирование: ЯрГУ (проект VIP-016).

Для цитирования: L. Y. Bystrov, A. N. Gladkov, and E. V. Kuzmin, "Detection of square wave impulse interference in eddy current rail defectograms", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 172–205, 2025.

DOI: [10.18255/1818-1015-2025-2-172-205](https://doi.org/10.18255/1818-1015-2025-2-172-205).

© Быстров Л. Ю., Гладков А. Н., Кузьмин Е. В., 2025

Эта статья открытого доступа под лицензией CC BY license (<https://creativecommons.org/licenses/by/4.0/>).

Введение

Обеспечение безопасности движения на железнодорожном транспорте требует своевременного выявления дефектов рельсов. Для решения этой задачи регулярно проводится неразрушающий контроль рельсового полотна. Одним из методов обнаружения дефектов в объектах железнодорожной инфраструктуры является вихретоковая дефектоскопия [1].

Данная статья продолжает серию работ по автоматической расшифровке вихретоковых дефектограмм [2–5]. Объектом исследования выступают записи 12-разрядного 15-канального вихретокового дефектоскопа (см. рис. 1). Каждый канал соответствует отдельному физическому датчику, установленному на поверхности железнодорожного рельса перпендикулярно направлению движения дефектоскопа. Устройство регистрирует сигналы (по каждому каналу) с шагом в один миллиметр железнодорожного пути.

Данные (дефектограммы), полученные в ходе дефектоскопии, требуют тщательного анализа. Цель такого анализа — выделение участков, указывающих на наличие дефектов в рельсе на фоне рельсового шума и различных помех.

Среди шумов, затрудняющих анализ дефектограмм, особо выделяются импульсные шумы. Импульсный шум (impulse noise, burst noise) — это взрывное шумовое событие, обычно электромагнитного происхождения, с потенциально высокой амплитудой (мощностью). К возникновению импульсного шума приводят скачки напряжения, наводки, отказы сенсоров регистратора [6].

Имеются разные виды импульсных шумов. Для мгновенных импульсных шумов существуют эффективные алгоритмы фильтрации, в основном представляющие собой модификации медианного фильтра [7]. Подавления импульсного шума такого рода удаётся достигнуть с помощью алгоритма SD-ROM [8], адаптивно-когнитивного фильтра Калмана [9, 10], систем активного подавления шума [11], методов машинного обучения [12].

Совсем иначе обстоит дело с продолжительными импульсными шумами. Фильтры, основанные на ранговых статистиках (в том числе медианные фильтры), с продолжительными шумами не справляются, потому что шум теперь представляет собой не выбросы в отдельных точках данных, а существенную, несущую составляющую. В случае биполярного продолжительного импульсного шума смешанный с ним сигнал приобретает выраженную прямоугольную форму, которая классическими методами фильтрации не устраняется (см. рис. 2).

Фильтрация данных сопряжена с потерей полезных сигналов. В отношении прямоугольного импульсного шума особенно важно перед применением алгоритмов подавления шума проверить, действительно ли данные содержат этот вид шумов. Таким образом, помимо подавления шума не менее востребованным является решение задачи его обнаружения.

Данная работа посвящена обнаружению прямоугольных продолжительных импульсных шумов в вихретоковых дефектограммах. Задача решается для записей только одного канала дефектоскопа, а не всех 15 каналов сразу, то есть поиск прямоугольных шумов производится по каждому каналу отдельно. Подход к обнаружению основан на прямоугольной форме шума и его вероятностном распределении. Фактически решается задача оценки соответствия распределения данных распределению биполярных импульсных сигналов. В разделе 3 приводится формальное определение прямоугольного сигнала, рассматривается форма распределения биполярных импульсных шумов. Показано, что биполярный импульсный сигнал имеет закон распределения гомоскедастичной смеси двух гауссианов. Этот вид распределения достаточно хорошо изучен [13–16]. В разделе 4 делается вероятностное предположение о нормальности исходных данных, на котором основывается обсуждаемый далее критерий обнаружения биполярных импульсных сигналов. В разделе 5 установлена нижняя оценка функции правдоподобия распределения прямоугольных сигналов. В разделе 6 введена SW -характеристика, изучен ряд её свойств, в том числе показано, как характеристика связана с прямоугольными сигналами и как её оценивать по выборке. В разделе 7 SW -характеристика вычис-

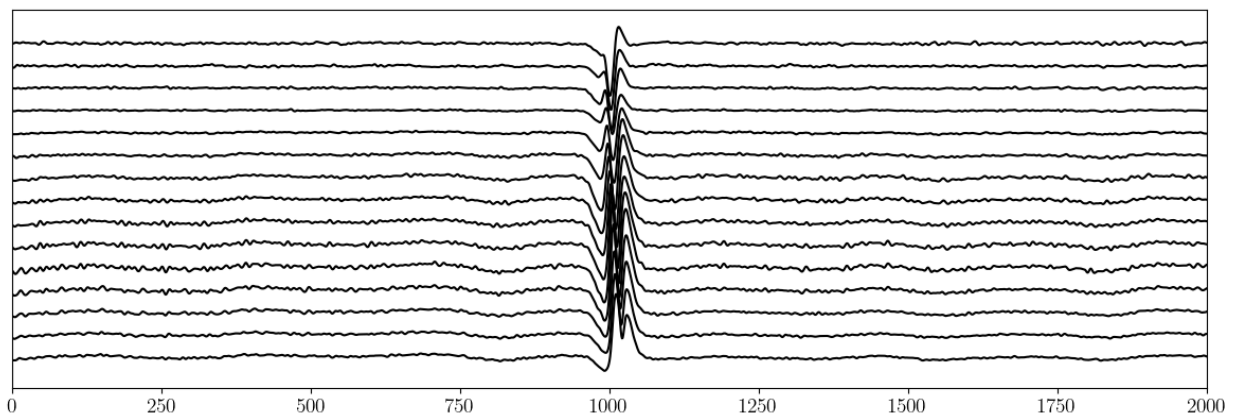


Fig. 1. The example of the flaw detector record of a bolted rail joint

Рис. 1. Пример записи дефектоскопом болтового стыка рельсов

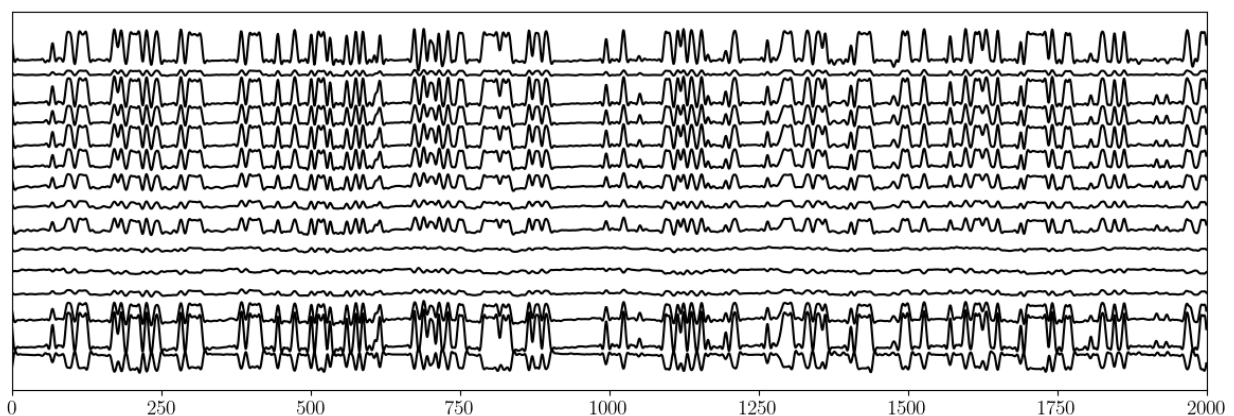


Fig. 2. The example of a square wave noise on an eddy current defectogram

Рис. 2. Пример прямоугольного шума на вихретоковой дефектограмме

лена для нормального распределения. В разделе 8 SW -характеристика считается для гомоскедастичной смеси двух гауссианов, показана связь SW -характеристики с бимодальностью распределения, выдвинут критерий обнаружения прямоугольных сигналов на основе SW -характеристики (SW -критерий). В разделе 9 приводятся экспериментальные результаты применения SW -критерия, EM -алгоритма и MDE -порога для обнаружения прямоугольных сигналов. Отмечено, что SW -критерий даёт лучшие результаты по сравнению с другими методами.

1. Обзор существующих решений

Традиционные методы, используемые для обнаружения импульсного шума, основаны на преобразовании Фурье. Методы эффективны только в том случае, если амплитуда импульсного шума имеет высокие значения, отчётливо выделяющиеся при идентификации, при этом методы оказываются трудоёмкими и требующими громоздких вычислений [17]. Другой подход к обнаружению продолжительных импульсных шумов основан на пороговой обработке вейвлет-коэффициентов. Авторы [18] выделили импульсный шум из гауссовского шума с помощью вейвлета Хаара. Также вейвлет-анализ для детекции импульсных шумов применялся в работе [17]. При этом ни те, ни другие исследователи не привели конкретных алгоритмов обнаружения импульсного шума. Их работы носят скорее методологический, а не конструктивный характер в отношении решаемой проблемы.

В геофизике нередко приходится работать с данными, представленными в виде временных рядов. Например, такой вид имеют результаты магнитотеллурического зондирования. В магнитотеллурических данных часто встречаются шумы культурного происхождения, к которым относятся и продолжительные прямоугольные шумы. Проанализируем несколько работ, вышедших в последние годы, которые посвящены подавлению и обнаружению прямоугольного шума в магнитотеллурических данных.

Исследователи из Цилинского университета в качестве показателя зашумлённости данных использовали многомасштабную дисперсную энтропию (Multiscale Dispersion Entropy, MDE) [19]. Если значение *MDE* ниже некоторого экспериментально полученного порога, то сигнал считался зашумлённым. Авторы статьи утверждают, что особенно хорошо *MDE* позволяет выявлять прямоугольный импульсный шум. Среди рассматриваемых ими шумов прямоугольный шум обладал наименьшим значением *MDE*. Помимо метода обнаружения прямоугольного шума исследователи предлагают также метод подавления шума на основе дискретного вейвлет-преобразования и сингулярного разложения. Они разбивают данные на несколько сегментов с помощью алгоритма восстановления фазового пространства. Далее, для каждого сегмента проводится вейвлет-фильтрация по заранее заданному количеству уровней вейвлет-коэффициентов. Отфильтрованные сегменты выстраиваются в матрицу, для которой ищется сингулярное разложение специального вида. Наконец, из матрицы сингулярных чисел выделяется подматрица, которая используется для восстановления исходного сигнала. Авторы утверждают, что их алгоритм подавления шумов успешно справляется с прямоугольным шумом, практически не искажая полезные элементы в данных. Свои утверждения они подтверждают тестами алгоритма на зашумлённых магнитотеллурических данных. При этом сам алгоритм подавления описывается авторами в общих словах, без предоставления конкретной реализации, что затрудняет проверку алгоритма на вихретоковых дефектограммах.

Авторы [20] для обнаружения прямоугольных импульсных помех предлагают использовать уточнённую составную многомасштабную дисперсную энтропию (RCMDE). Они вычисляют RCMDE с разными параметрами масштаба для получения вектора признаков, описывающего сигнал, с последующей кластеризацией с помощью алгоритма нечёткой кластеризации (Fuzzy C-means). В статье отмечаются преимущества RCMDE по сравнению с многомасштабной дисперсной энтропией (*MDE*) и многомасштабной энтропией (*ME*). Кластеризация на основе вектора RCMDE используется для обнаружения прямоугольных шумов, при этом, как интерпретировать результаты кластеризации в случае отсутствия прямоугольного шума, авторы не сообщают. Для подавления шумов предлагается использовать алгоритм OMP (Orthogonal matching pursuit) с обученным словарём на основе синусоидальных волн и вейвлетов семейства симлетов (sym) и Добеши (db). В заключении авторы подчёркивают, что при амплитуде прямоугольного шума, сопоставимой с амплитудой чистого сигнала, предложенный метод работать не будет.

В статье [21] обнаружение прямоугольных шумов производится с помощью свёрточных нейронных сетей с остаточными связями (ResNet). Для каждых 32 отчётов магнитотеллурических данных авторы сформировали грамианово угловое поле (GAF), которое в виде цветного изображения подавалось на вход сети. Каждый GAF был отмечен как содержащий (класс 1) или не содержащий шум (класс 0). Объём обучающей выборки составил 10 000 000 изображений. Доля правильных ответов (ассура) на валидационной выборке превысила 98 %. При этом авторы не пишут, как они сформировали обучающую выборку. Непонятно, использовали ли они ручную разметку или применяли какие-то алгоритмы для автоматического обнаружения шумов. Те же вопросы вызывает их метод подавления шумов, основанный на свёрточной сети с остаточными связями, но уже не с двумерными, а с одномерными свёртками. Авторы используют в обучающей выборке зашумлённые и очищенные от шума фрагменты данных, хотя о том, как данные были очищены от шума, в статье не сказано.

Ещё один подход для подавления прямоугольных шумов в магнитотеллурических данных на основе нейронных сетей представили авторы [22]. Исследователи использовали DRSN-сеть (Deep Residual Shrinkage Network). Эта нейронная сеть работает непосредственно с одномерными сигналами, а не с изображениями. Для формирования обучающей выборки авторы создают синтетические данные, искусственно прибавляя к чистым данным шумы разной формы. Обнаружение прямоугольных или иных шумов перед фильтрацией не производится. То же самое можно сказать и о работе [23]. В ней для подавления прямоугольного шума используется шумоподавляющая свёрточная нейронная сеть (Denoising Convolutional Neural Network) с управляющим рекуррентным блоком (Gated Recurrent Unit), интегрированная с обучением словаря по методу К-сингулярного разложения (K-Singular Value Decomposition). Обучение сети производилось на основе синтетических данных. Перед шумоподавлением процедура обнаружения шума отсутствует.

Итак, в последнее время появилось достаточно много новых методов решения задачи подавления прямоугольного шума. Эффективность и надёжность данных методов нуждается в апробации, которую иногда трудно провести ввиду расплывчатого и неполного описания методов в соответствующих статьях. Задаче обнаружения прямоугольного шума в литературе уделено гораздо меньше внимания. Существующие немногочисленные алгоритмы не гарантируют корректности своих ответов. Поэтому задача обнаружения прямоугольных шумов продолжает оставаться актуальной. В данной работе предлагается и исследуется одно из возможных её решений.

2. Основные понятия

Пусть X — непрерывная случайная величина, через $F_X(x)$ обозначим её функцию распределения, а через $f_X(t)$ — её функцию плотности:

$$F_X(x) = \int_{-\infty}^x f_X(t) dt.$$

$\mathbb{E}X$ — математическое ожидание величины X , $\mathbb{V}X$ — её дисперсия:

$$\mathbb{V}X = \mathbb{E}(X - \mathbb{E}X)^2 = \mathbb{E}X^2 - (\mathbb{E}X)^2. \quad (1)$$

Пусть $g(x)$ — функция действующая из $\mathbb{R}$ в $\mathbb{R}$. Тогда, если случайная величина $Y = g(X)$ имеет математическое ожидание, то оно может быть найдено по формуле

$$\mathbb{E}(g(X)) = \int_{-\infty}^{+\infty} g(t) f_X(t) dt.$$

Через $\mathbb{I}[Pred(X)]$ будем обозначать индикаторную случайную величину

$$\mathbb{I}[Pred(X)] = \begin{cases} 1, & Pred(X) = true, \\ 0, & Pred(X) = false. \end{cases}$$

Пусть $X_1, X_2, \dots, X_n$ — независимые одинаково распределённые с величиной X случайные величины. Последовательность чисел $x_1, x_2, \dots, x_n$, являющихся значениями соответствующих случайных величин $X_1, X_2, \dots, X_n$, называется *выборкой* объёма n , *порождённой случайной величиной* X .

Пусть X — непрерывная случайная величина, распределённая по некоторому закону с параметрами $\Theta = (\theta_1, \theta_2, \dots, \theta_k)$, последовательность $x_1, x_2, \dots, x_n$ — выборка, порождённая случайной величиной X , $f(\Theta, t)$ — функция плотности распределения X . *Функцией правдоподобия* $L(\Theta)$ называется функция

$$L(\Theta) = \prod_{i=1}^n f(\Theta, x_i).$$

Вероятность события A будем обозначать как $\mathbb{P}(A)$. Последовательность случайных величин $X_1, X_2, \dots, X_n$ сходится по вероятности к случайной величине X , то есть $X_n \xrightarrow{P} X$ [24], если

$$\forall \varepsilon > 0 : \lim_{n \rightarrow \infty} \mathbb{P}(|X_n - X| > \varepsilon) = 0.$$

Характеристическую функцию случайной величины X обозначим через $\varphi_X(t)$:

$$\varphi_X(t) = \mathbb{E}(e^{itX}), \text{ где } i = \sqrt{-1}.$$

Пусть $X_1, X_2, \dots, X_n$ — непрерывные случайные величины с функциями плотности распределения $f_{X_1}(t), f_{X_2}(t), \dots, f_{X_n}(t)$ соответственно, числа $p_1, p_2, \dots, p_n$ такие, что $p_i \in [0; 1], \forall i \in \overline{1, n}$ и $\sum_{i=1}^n p_i = 1$. Тогда смесью величин $X_1, X_2, \dots, X_n$ будет называться величина X , имеющая функцию плотности распределения

$$f_X(t) = \sum_{i=1}^n p_i f_{X_i}(t).$$

В частности, смесью нормальных распределений или смесью гауссиан будет называться смесь величин $X_1, X_2, \dots, X_n$, имеющих нормальный закон распределения $\mathbb{N}(\mu_i, \sigma_i^2), i \in \overline{1, n}$. При равенстве $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2$ смесь называется гомоскедастичной [14].

3. Прямоугольный сигнал

Под прямоугольным сигналом будем понимать периодическую с периодом T функцию $s(t)$, заданную в точках $t \in [0; n]$, вида

$$s(t) = \begin{cases} m_1, & t \in [kT; (p+k)T), \\ m_2, & t \in [(p+k)T; (k+1)T), \end{cases} \quad (2)$$

где $k \in \overline{0, \frac{n}{T} - 1}$. Число n считается кратным T .

Прямоугольный сигнал $s(t)$ является биполярным, то есть принимает в периоде одно из двух полярных значений (импульсов). Числа $m_1 \leq 0, m_2 > 0$ соответствуют значениям импульса. Длительность первого импульса m_1 на периоде T определяется как pT , где $p \in (0; 1)$, а длительность второго импульса m_2 соответственно будет равна $(1-p)T$. Графически прямоугольный сигнал представлен на рис. 3.

На практике идеальный прямоугольный сигнал, описываемый функцией $s(t)$, встречается редко. Более реалистичным сценарием является тот, когда на сигнал накладывается шум, искажающий его идеальную прямоугольную форму (рис. 4). Будем предполагать, что на прямоугольный сигнал

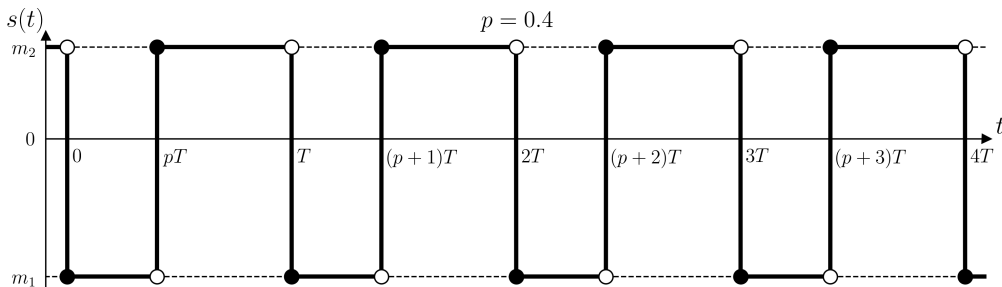


Fig. 3. A square wave signal

Рис. 3. Прямоугольный сигнал

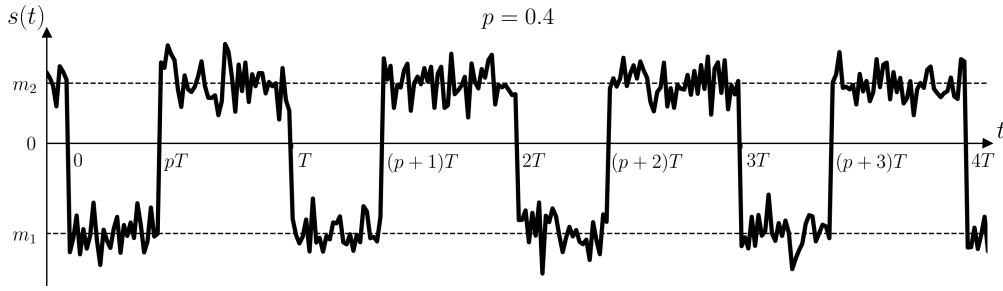


Fig. 4. The square wave signal with normal noise

Рис. 4. Прямоугольный сигнал с нормальным шумом

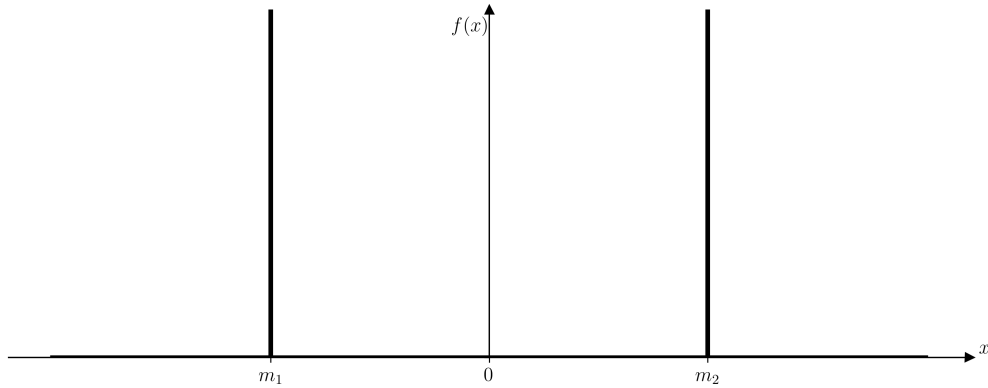


Fig. 5. The distribution density function of a square wave signal

Рис. 5. Функция плотности распределения прямоугольного сигнала

накладывается нормальный гомоскедастичный шум с дисперсией σ^2 . В этом случае можно говорить о случайных величинах Y_t вида

$$Y_t = s(t) + \varepsilon_t, \text{ где } \varepsilon_t \sim \mathbb{N}(0; \sigma^2). \quad (3)$$

Величины Y_t будут распределены по нормальному закону $\mathbb{N}(m_1; \sigma^2)$ при $t \in [kT; (p+k)T)$ и по нормальному закону $\mathbb{N}(m_2; \sigma^2)$ при $t \in [(p+k)T; (k+1)T)$, то есть в зависимости от параметра t величины Y_t могут иметь разные законы распределения.

Если бы импульсы в сигнале $s(t)$ не имели обязательной продолжительности и могли бы возникать произвольно в любой момент времени t , а параметр p регулировал бы соотношение количества отрицательных и положительных импульсов, то сигнал $s(t)$ можно было бы рассматривать как функцию от случайной величины, плотность распределения которой можно записать через смесь двух δ -функций Дирака (рис. 5):

$$f(x) = p\delta(x - m_1) + (1 - p)\delta(x - m_2), \text{ где } \delta(x) = 0 \text{ для } \forall x \neq 0 \text{ и } \int_{-\infty}^{+\infty} \delta(x)dx = 1.$$

Конечно, сигнал $s(t)$ с произвольными моментами импульсации уже не является прямоугольным. Но модель такого сигнала может быть полезна для исследования прямоугольных сигналов. В отличие от Y_t получаемые здесь случайные величины Z_t будут иметь один и тот же закон распределения независимо от времени t — гомоскедастичную смесь двух гауссиан $\mathbb{N}_2(m_1, m_2, \sigma^2, p)$, где m_1, m_2 — математические ожидания значений отрицательного и положительного импульсов,

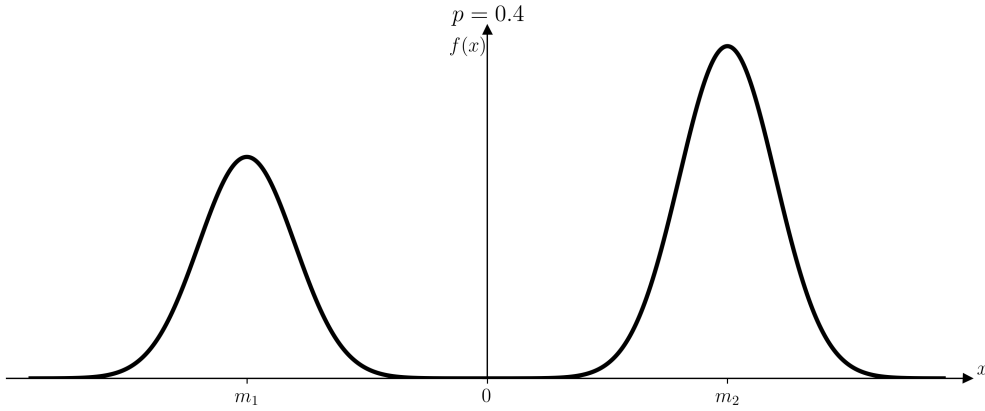


Fig. 6. The distribution density function of a square wave signal with normal noise

Рис. 6. Функция плотности распределения прямоугольного сигнала с нормальным шумом

σ^2 — дисперсия нормального шума, p — параметр смеси. Такой переход позволяет заменить последовательность величин Y_t , имеющих разный закон распределения в зависимости от t , на одну единственную величину Z :

$$Z_t = Z \sim \mathbb{N}_2(m_1, m_2, \sigma^2, p). \quad (4)$$

Первый гауссиан сосредоточен вокруг отрицательного импульса m_1 по нормальному закону $\mathbb{N}(m_1, \sigma^2)$. Соответственно, второй гауссиан сосредоточен вокруг положительного значения m_2 по закону $\mathbb{N}(m_2; \sigma^2)$. Соотношение гауссианов в смеси определяется параметром $p \in [0; 1]$. Это те самые гауссианы, которые описывали распределение величин Y_t (рис. 6). Функция плотности распределения величины Z имеет вид

$$f_Z(x) = \frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m_1)^2}{2\sigma^2}} + \frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m_2)^2}{2\sigma^2}}. \quad (5)$$

Величина Z соответствует импульсным сигналам с произвольными моментами импульсации. Такой объект для исследования кажется более простым по сравнению с последовательностью величин Y_t прямоугольного шума, поэтому пока остановимся на нём. В дальнейшем будет предложен способ связать произвольные импульсные шумы с прямоугольными через SW -характеристику.

Важным свойством прямоугольного шума является его выраженная полярность, или, иначе, бимодальность. Если m_1 и m_2 достаточно близки друг к другу, то смесь двух гауссиан вырождается в один единственный гауссиан. Необходимые и достаточные условия бимодальности гомоскедастичной смеси двух гауссиан были получены в [13].

Теорема о бимодальности гомоскедастичной смеси. Функция плотности (5) распределения является бимодальной тогда и только тогда, когда число $D = \frac{|m_2 - m_1|}{2\sigma}$ удовлетворяет всем перечисленным ниже условиям:

- (1) $D^2 > 1$,
- (2) $|\log \frac{p}{1-p}| < 2 \log(D - \sqrt{D^2 - 1}) + 2D\sqrt{D^2 - 1}$.

В нашем случае $m_2 > m_1$, поэтому модуль в числе D можно опустить. Рисунки 7 и 8 показывают, что происходит со смесью гауссиан $\mathbb{N}_2$, когда условия бимодальности (1) и (2) не соблюдаются. При нарушении условий биполярность функции плотности распределения теряется. Прямоугольные сигналы не могут иметь такого распределения. Таким образом, необходимо потребовать, чтобы функция плотности распределения величины Z была бимодальной смесью двух гауссиан.

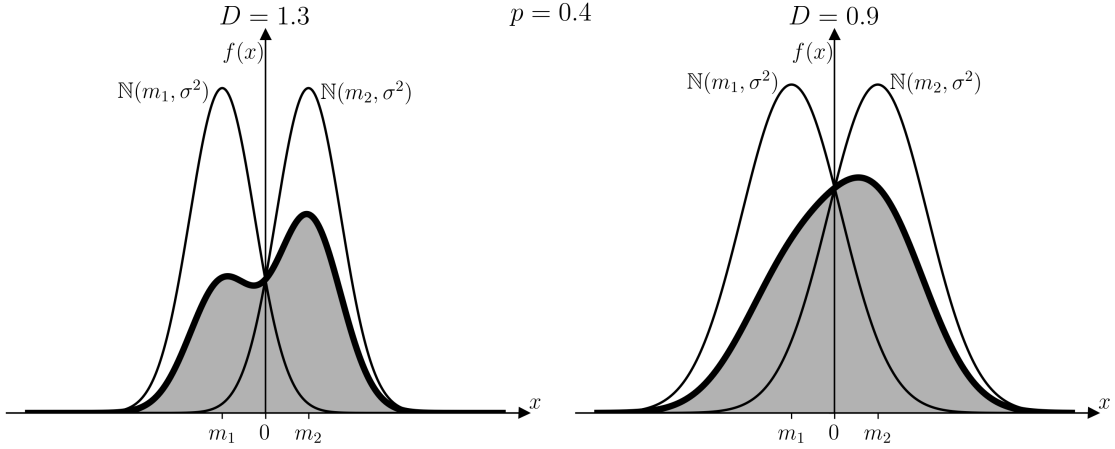


Fig. 7. The distribution density function of two Gaussian mixture when the condition (1) of the bimodality theorem is met (on the left) and when the condition is violated (on the right)

Рис. 7. Функция плотности распределения смеси двух гауссиан при выполнении условия (1) теоремы о бимодальности (слева) и нарушении этого условия (справа)

Число p отвечает за частоту встречаемости отрицательных импульсов среди импульсов обоих знаков. Изменение параметра p регулирует относительную высоту гауссиан в смеси. При значениях p , близких к 0 или к 1, смесь гауссиан в соответствии с [теоремой о бимодальности](#) теряет полярность, что, опять же, не характерно для прямоугольных сигналов.

4. Вероятностные предположения о данных

Будем считать, что данные одного канала вихретоковой дефектограммы $x_1, x_2, \dots, x_n$ порождены случайной величиной X с нормальным законом распределения $\mathbb{N}(\mu, s^2)$. К данным примешивается импульсный шум с распределением $\mathbb{N}_2$.

Лемма 1. Пусть $X \sim \mathbb{N}(\mu, s^2)$, $Z \sim \mathbb{N}_2(m_1, m_2, \sigma^2, p)$, X и Z независимы, тогда случайная величина $H = X + Z$ имеет распределение смеси гауссиан такое, что $H \sim \mathbb{N}_2(m_1 + \mu, m_2 + \mu, \sigma^2 + s^2, p)$.

Доказательство. Характеристическая функция нормально распределённой случайной величины K , $K \sim \mathbb{N}(m, \sigma^2)$, имеет вид [25]

$$\varphi_K(t) = e^{itm - \frac{\sigma^2 t^2}{2}}. \quad (6)$$

Рассмотрим вид характеристической функции смеси двух гауссиан.

$$\begin{aligned} \varphi_Z(t) = \mathbb{E}e^{itZ} &= \int_{-\infty}^{+\infty} e^{itl} f_Z(l) dl = \int_{-\infty}^{+\infty} e^{itl} \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(l-m_1)^2}{2\sigma^2}} + \frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(l-m_2)^2}{2\sigma^2}} \right) dl = \\ &= p \int_{-\infty}^{+\infty} e^{itl} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(l-m_1)^2}{2\sigma^2}} dl + (1-p) \int_{-\infty}^{+\infty} e^{itl} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(l-m_2)^2}{2\sigma^2}} dl. \end{aligned} \quad (7)$$

Каждый из интегралов в формуле (7) представляет собой характеристическую функцию нормального распределения. Заменим их по формуле (6). Получим

$$\varphi_Z(t) = pe^{itm_1 - \frac{\sigma^2 t^2}{2}} + (1-p)e^{itm_2 - \frac{\sigma^2 t^2}{2}}. \quad (8)$$

Итак, характеристическая функция смеси гауссиан имеет вид (8).

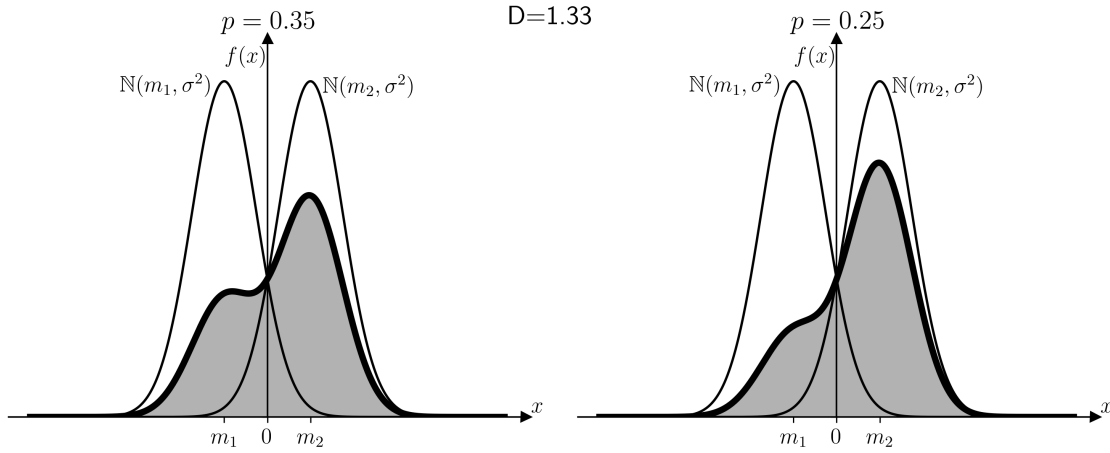


Fig. 8. The distribution density function of two Gaussian mixture when the condition (2) of the bimodality theorem is met (on the left) and when the condition is violated (on the right)

Рис. 8. Функция плотности распределения смеси двух гауссиан при выполнении условия (2) теоремы о бимодальности (слева) и нарушении этого условия (справа)

Известно, что если случайные величины $K_1, K_2, \dots, K_n$ независимы, то характеристическая функция их совместной величины $(K_1, K_2, \dots, K_n)$ может быть найдена как [26]

$$\varphi(t) = \prod_{i=1}^n \varphi_{K_i}(t).$$

Величины X и Z независимы. Это означает, что характеристическую функцию их суммы $X + Z$ можно вычислить следующим образом:

$$\begin{aligned} \varphi_H(t) = \varphi_{X+Z}(t) &= \varphi_X(t) \cdot \varphi_Z(t) = \left(p e^{itm_1 - \frac{\sigma^2 t^2}{2}} + (1-p) e^{itm_2 - \frac{\sigma^2 t^2}{2}} \right) \cdot e^{it\mu - \frac{s^2 t^2}{2}} = \\ &= p e^{it(m_1 + \mu) - \frac{(\sigma^2 + s^2)t^2}{2}} + (1-p) e^{it(m_2 + \mu) - \frac{(\sigma^2 + s^2)t^2}{2}}. \end{aligned} \quad (9)$$

Заметим, что формулы (8) и (9) отличаются только значениями параметров распределения $\mathbb{N}_2$. Таким образом, величина H имеет распределение $\mathbb{N}_2(m_1 + \mu, m_2 + \mu, \sigma^2 + s^2, p)$. $\square$

Лемма 1 позволяет утверждать, что если данные имеют нормальное распределение и к ним аддитивным образом примешивается биполярный импульсный шум с распределением $\mathbb{N}_2$, то новое распределение будет также удовлетворять закону $\mathbb{N}_2$.

5. Правдоподобие импульсного шума

В связи с тем, что нормально распределённые данные сохраняют форму распределения произвольного импульсного шума $\mathbb{N}_2$, возникает идея обнаружения этого шума на основе оценки соответствия распределения данных распределению $\mathbb{N}_2$.

Наиболее распространённым методом оценки параметров смеси распределений является *ЕМ*-алгоритм (Expectation Maximization Algorithm) [27]. Алгоритм является итеративным и решает задачу, максимизируя функцию правдоподобия данных. Хотя *ЕМ*-алгоритм находит широкое применение, он обладает существенным недостатком, так как обеспечивает сходимость только к локальному минимуму функции правдоподобия. В связи с этим разработан ряд альтернативных методов оценки параметров смеси [15].

В *ЕМ*-алгоритме и в его аналогах количество распределений в смеси является гиперпараметром. Это означает, что для определения количества гауссиан в смеси необходимо использовать какие-то

другие методы. Распространённым подходом является оценка числа распределений в смеси с помощью информационного критерия. Модель смеси с наименьшим значением информационного критерия считается наиболее подходящей для имеющихся данных. Таким образом, с помощью информационного критерия можно оценить, наличие какого количества гауссиан в смеси наиболее оправдано. Популярными информационными критериями являются байесовский критерий (*BIC*), критерии Такеучи (*TIC*) и Акаике (*AIC*) [28].

Применение *ЕМ*-алгоритма для обнаружения прямоугольных сигналов в вихретоковых дефектограммах осложнено требованием алгоритма для своей сходимости большого количества данных, тогда как на практике прямоугольные сигналы чаще всего обладают малой протяжённостью. Поэтому востребованными остаются эффективные методы обнаружения прямоугольных сигналов, работающие на относительно небольших выборках.

Пусть $\bar{x} = (x_1, x_2, \dots, x_n)$ — выборка из распределения $\mathbb{N}_2(m_1, m_2, \sigma^2, p)$, порождённая случайной величиной X . Рассмотрим функцию правдоподобия

$$L(\bar{x}, m_1, m_2, \sigma, p) = \prod_{i=1}^n \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_1)^2}{2\sigma^2}} + \frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_2)^2}{2\sigma^2}} \right). \quad (10)$$

Разделим произведение (10) на 2 части: в первом произведении будут фигурировать числа $x_i \leq \mathbb{E}X$, то есть данные, не превосходящие математическое ожидание величины X , а во втором — все оставшиеся числа. Через K обозначим количество величин $X_1, X_2, \dots, X_n$, не превосходящих $\mathbb{E}X$: $K = \sum_{i=1}^n \mathbb{I}[X_i \leq \mathbb{E}X]$. Число k — реализация величины K на выборке $\bar{x}$.

$$L(\bar{x}, m_1, m_2, \sigma, p) = \prod_{x_i \leq \mathbb{E}X} \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_1)^2}{2\sigma^2}} + \frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_2)^2}{2\sigma^2}} \right) \cdot \prod_{x_i > \mathbb{E}X} \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_1)^2}{2\sigma^2}} + \frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_2)^2}{2\sigma^2}} \right) = L_1 \cdot L_2. \quad (11)$$

Оценим L снизу с помощью формулы

$$L(\bar{x}, m_1, m_2, \sigma, p) \geq \prod_{x_i \leq \mathbb{E}X} \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_1)^2}{2\sigma^2}} \right) \cdot \prod_{x_i > \mathbb{E}X} \left(\frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - m_2)^2}{2\sigma^2}} \right) = L'_1 \cdot L'_2. \quad (12)$$

Если математическое ожидание $\mathbb{E}X$ равно 0, то числа m_1 и m_2 соответствуют средним значениям отрицательных и положительных импульсов в сигнале. В общем случае, m_1 — это среднее значение импульсов, не превосходящих $\mathbb{E}X$, а m_2 — среднее значение импульсов, больших $\mathbb{E}X$: $m_1 \leq \mathbb{E}X < m_2$. Числа L'_1 и L'_2 в формуле (12) можно интерпретировать как правдоподобия «левого» и «правого» гауссиан в смеси. Та часть функции L , которой нижняя оценка пренебрегает, соответствует правдоподобию данных в «хвостах» гауссиан. В случае ярко выраженных биполярных импульсов погрешность оценки становится несущественной.

Рассмотрим логарифм правдоподобия нижней оценки L из формулы (12)

$$\ln(L'_1 \cdot L'_2) = k \ln \left(\frac{p}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum_{x_i \leq \mathbb{E}X} (x_i - m_1)^2 + (n - k) \ln \left(\frac{(1-p)}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum_{x_i > \mathbb{E}X} (x_i - m_2)^2. \quad (13)$$

Рост логарифма правдоподобия в формуле (13) обеспечивается в первую очередь за счёт минимизации среднеквадратического отклонения данных, не превосходящих $\mathbb{E}X$, от среднего значения отрицательных импульсов m_1 и среднеквадратического отклонения данных, больших $\mathbb{E}X$, от среднего значения положительных импульсов m_2 . Величина этих отклонений определяет правдоподобие

двух компонент сигнала, наличие двух заметных импульсов. Чем меньше значение среднеквадратических отклонений, тем более распределение данных похоже на смесь двух гауссиан, а значит и на распределение импульсного шума. Характеристика, основанная на этих величинах, может позволить отличать биполярные импульсные шумы от прочих шумов, имеющих распределение $\mathbb{N}_2$. Подробнее эта идея будет рассмотрена в разделе 6.

Приравниваем формулы (13) к 0 можно получить оценки параметров распределения:

$$p' = \frac{K}{n}, \quad (14)$$

$$m'_1 = \frac{1}{K} \sum_{i=1}^n X_i \cdot \mathbb{I}[X_i \leq \mathbb{E}X], \quad (15)$$

$$m'_2 = \frac{1}{n-K} \sum_{i=1}^n X_i \cdot \mathbb{I}[X_i > \mathbb{E}X]. \quad (16)$$

6. SW-характеристика

Пусть X — непрерывная случайная величина с функцией плотности распределения $f_X(t)$, $\mathbb{E}X$ — её математическое ожидание. Определим на основе X две случайные величины X_1 и X_2 . Функция плотности $f_{X_1}(t)$ имеет тот же вид, что и функция $f_X(t)$ на промежутке $(-\infty; \mathbb{E}X]$ и равна 0 на прочей области. Число J_1 выступает в роли нормирующей константы для распределения.

$$f_{X_1}(t) = \begin{cases} \frac{f_X(t)}{J_1}, & t \in (-\infty; \mathbb{E}X], J_1 = \int_{-\infty}^{\mathbb{E}X} f_X(l) dl, \\ 0, & t \in (\mathbb{E}X; +\infty). \end{cases} \quad (17)$$

Аналогично функция плотности $f_{X_2}(t)$ величины X_2 имеет вид

$$f_{X_2}(t) = \begin{cases} \frac{f_X(t)}{J_2}, & t \in (\mathbb{E}X; +\infty), J_2 = \int_{\mathbb{E}X}^{+\infty} f_X(l) dl, \\ 0, & t \in (-\infty; \mathbb{E}X]. \end{cases} \quad (18)$$

Определим характеристику, которую в дальнейшем будем использовать для оценки соответствия распределения данных распределению прямоугольных сигналов. Эта характеристика ранее в литературе не упоминалась, в данной работе вводится впервые.

Определение 1. Пусть функция $g(X)$ от случайной величины X имеет вид

$$g(X) = \begin{cases} \left(\frac{X - \mathbb{E}X_1}{\mathbb{E}X_1 - \mathbb{E}X_2} \right)^2, & X \in (-\infty; \mathbb{E}X], \\ \left(\frac{X - \mathbb{E}X_2}{\mathbb{E}X_1 - \mathbb{E}X_2} \right)^2, & X \in (\mathbb{E}X; +\infty). \end{cases} \quad (19)$$

SW-характеристикой (Square Waves Characteristic) случайной величины X назовём число

$$\begin{aligned} SW(X) = \mathbb{E}_X(g(X)) &= \int_{-\infty}^{+\infty} g(t) f_X(t) dt = \int_{-\infty}^{\mathbb{E}X} \left(\frac{t - \mathbb{E}X_1}{\mathbb{E}X_1 - \mathbb{E}X_2} \right)^2 f_X(t) dt + \\ &+ \int_{\mathbb{E}X}^{+\infty} \left(\frac{t - \mathbb{E}X_2}{\mathbb{E}X_1 - \mathbb{E}X_2} \right)^2 f_X(t) dt = \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt + \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt \right). \end{aligned} \quad (20)$$

SW-характеристика показывает, чему в среднем равна сумма квадратов отклонений случайной величины X на интервале $(-\infty; \mathbb{E}X)$ от среднего значения величины на этом интервале $\mathbb{E}X_1$

и квадратов отклонений этой величины на полуинтервале $[\mathbb{E}X; +\infty)$ от среднего значения на этом полуинтервале $\mathbb{E}X_2$. Это те самые квадратичные отклонения, которые фигурировали в (13). Вместо чисел m_1, m_2 выступают их оценки $\mathbb{E}X_1, \mathbb{E}X_2$ в соответствии с оценками правдоподобия (15), (16). Таким образом, минимизация SW -характеристики ведёт к максимизации правдоподобия смеси гауссиан $\mathbb{N}_2$.

Число $\frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2}$ призвано усилить значение «полярности» в SW -характеристике. Разность $|m_2 - m_1|$ тесно связана с количеством мод в распределении. В соответствии с теоремой о бимодальности гомоскедастичная смесь двух гауссиан является бимодальной, только если квадрат разности $|m_2 - m_1|$ по крайней мере в 4 раза превосходит дисперсию σ^2 . Соответственно, при малых значениях $|m_2 - m_1|$ бимодальность в смеси становится всё менее выраженной и даже пропадает вовсе. Функция плотности распределения биполярных импульсных шумов должна обладать выраженной бимодальностью, поэтому разность $|m_2 - m_1|$ и оценка разности $|\mathbb{E}X_2 - \mathbb{E}X_1|$ в таких сигналах должны быть достаточно большими величинами. Это в свою очередь означает, что SW -характеристика для них должна быть «небольшой». Формальная оценка величины SW -характеристики для биполярных импульсных сигналов будет проведена в следующих разделах статьи.

Отметим, что кусочно-заданная формулой (19) функция $g(X)$ в точке $\mathbb{E}X$ терпит разрыв 1-го рода. Но, так как рассматриваемые интегралы являются лебеговскими, а не римановскими, данное обстоятельство никак не вредит интегрируемости функции на промежутке $(\mathbb{E}X; +\infty)$ [26].

Лемма 2 позволяет оценивать SW -характеристику по выборке.

Лемма 2. Пусть $X_1, X_2, \dots, X_n$ — независимые и одинаково распределённые с X величины, $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$, K — количество $X_i \leq \bar{X}$, то есть $K = \sum_{i=1}^n \mathbb{I}[X_i \leq \bar{X}]$. Также определены оценки $\bar{X}_1$ и $\bar{X}_2$ как $\bar{X}_1 = \frac{1}{K} \sum_{i=1}^n X_i \cdot \mathbb{I}[X_i \leq \bar{X}]$ и $\bar{X}_2 = \frac{1}{n-K} \sum_{i=1}^n X_i \cdot \mathbb{I}[X_i > \bar{X}]$. Тогда оценка $\widehat{SW}(X_1, X_2, \dots, X_n)$

$$\widehat{SW}(X_1, X_2, \dots, X_n) = \frac{1}{n} \sum_{i=1}^n \left(\frac{(X_i - \bar{X}_1) \cdot \mathbb{I}[X_i \leq \bar{X}]}{\bar{X}_1 - \bar{X}_2} \right)^2 + \frac{1}{n} \sum_{i=1}^n \left(\frac{(X_i - \bar{X}_2) \cdot \mathbb{I}[X_i > \bar{X}]}{\bar{X}_1 - \bar{X}_2} \right)^2 \quad (21)$$

сходится по вероятности к $SW(X)$ при $n \rightarrow \infty$.

Доказательство. Лемма целиком основывается на следствии из закона больших чисел [24], которое гласит, что выборочное среднее сходится по вероятности к математическому ожиданию

$$\bar{X} \xrightarrow{P} \mathbb{E}X \text{ при } n \rightarrow \infty.$$

Обозначим через S_1 сумму

$$S_1 = \sum_{i=1}^n X_i \cdot \mathbb{I}[X_i \leq \bar{X}].$$

Необходимо показать, что

$$\bar{X}_1 = \frac{S_1}{K} \xrightarrow{P} \mathbb{E}X_1 = \mathbb{E}(X|X \leq \mathbb{E}X) \text{ при } n \rightarrow \infty.$$

Величина K равна количеству величин X_i , не превосходящих $\bar{X}$. Это означает, что K имеет биномиальное распределение $Binomial(n, q)$, где $q = \mathbb{P}(X \leq \mathbb{E}X)$. Рассмотрим число $\frac{K}{n}$. Сумму индикаторов, образующую K , разобьём на 2 части:

$$\frac{K}{n} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[X_i \leq \bar{X}] = \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbb{I}[X_i \leq \mathbb{E}X]}_{A_n} + \underbrace{\frac{1}{n} \sum_{i=1}^n (\mathbb{I}[X_i \leq \bar{X}] - \mathbb{I}[X_i \leq \mathbb{E}X])}_{B_n}. \quad (22)$$

По закону больших чисел

$$A_n \xrightarrow{P} q = \mathbb{P}(X \leq \mathbb{E}X) \text{ при } n \rightarrow \infty.$$

Покажем, что B_n сходится по вероятности к 0. Рассмотрим $|B_n|$:

$$|B_n| = \frac{1}{n} \sum_{i=1}^n \left| \mathbb{I}[X_i \leq \bar{X}] - \mathbb{I}[X_i \leq \mathbb{E}X] \right|.$$

Заметим, что ненулевой вклад в $|B_n|$ дают индикаторы событий, когда X_i попадает между $\bar{X}$ и $\mathbb{E}X$, то есть $\mathbb{I} \left[\min\{\bar{X}, \mathbb{E}X\} \leq X_i \leq \max\{\bar{X}, \mathbb{E}X\} \right]$. Вычислим вероятность индикаторного события:

$$\mathbb{P}(\min\{\bar{X}, \mathbb{E}X\} \leq X_i \leq \max\{\bar{X}, \mathbb{E}X\}) = \int_{\min\{\bar{X}, \mathbb{E}X\}}^{\max\{\bar{X}, \mathbb{E}X\}} f_X(t) dt = f_X(c) \cdot |\bar{X} - \mathbb{E}X|, c \in [\min\{\bar{X}, \mathbb{E}X\}; \max\{\bar{X}, \mathbb{E}X\}].$$

По закону больших чисел $|\bar{X} - \mathbb{E}X|$ сходится по вероятности к 0, поэтому

$$\mathbb{P}(\min\{\bar{X}, \mathbb{E}X\} \leq X_i \leq \max\{\bar{X}, \mathbb{E}X\}) \xrightarrow{P} 0 \text{ при } n \rightarrow \infty.$$

Величину $n \cdot |B_n|$ можно интерпретировать как количество исходов, когда величина X_i оказалась заключена между $\bar{X}$ и $\mathbb{E}X$, то есть можно считать, что $n \cdot |B_n|$ распределена по биномиальному закону $Binomial(n, \mathbb{P}(\min\{\bar{X}, \mathbb{E}X\} \leq X_i \leq \max\{\bar{X}, \mathbb{E}X\}))$. Тогда по закону больших чисел

$$|B_n| \xrightarrow{P} \mathbb{P}(\min\{\bar{X}, \mathbb{E}X\} \leq X_i \leq \max\{\bar{X}, \mathbb{E}X\}) \xrightarrow{P} 0 \text{ при } n \rightarrow \infty.$$

Итак, $|B_n|$ сходится по вероятности к 0. Тогда из (22) следует, что

$$\frac{K}{n} \xrightarrow{P} q = \mathbb{P}(X \leq \mathbb{E}X) \text{ при } n \rightarrow \infty. \quad (23)$$

Величины $X_1, X_2, \dots, X_n$ независимы, поэтому величины $X_i \cdot \mathbb{I}[X_i \leq \bar{X}]$, являющиеся функциями от исходных величин X_i , будут также независимы [24]. По тому же закону больших чисел величина $\frac{S_1}{n}$ будет сходиться по вероятности к математическому ожиданию величины $X \cdot \mathbb{I}[X \leq \mathbb{E}X]$:

$$\frac{S_1}{n} \xrightarrow{P} \mathbb{E}(X \cdot \mathbb{I}[X \leq \mathbb{E}X]) \text{ при } n \rightarrow \infty. \quad (24)$$

Величина $X \cdot \mathbb{I}[X \leq \mathbb{E}X]$ равна $X|X \leq \mathbb{E}X$ с вероятностью $\mathbb{P}(X \leq \mathbb{E}X) = q$ и равна 0 с вероятностью $1 - q$. Поэтому ожидание $\mathbb{E}(X \cdot \mathbb{I}[X \leq \mathbb{E}X])$ можно переписать как

$$\mathbb{E}(X \cdot \mathbb{I}[X \leq \mathbb{E}X]) = \mathbb{E}(q \cdot X|X \leq \mathbb{E}X) = q \cdot \mathbb{E}(X|X \leq \mathbb{E}X) = q \cdot \mathbb{E}X_1.$$

Из (23) и (24) следует, что $\bar{X}_1$ будет сходиться по вероятности к $\mathbb{E}X_1$ при $K \neq 0$:

$$\bar{X}_1 = \frac{S_1}{K} = \frac{\frac{S_1}{n}}{\frac{K}{n}} \xrightarrow{P} \frac{q \cdot \mathbb{E}X_1}{q} = \mathbb{E}X_1 \text{ при } n \rightarrow \infty.$$

Аналогично можно показать, что $\bar{X}_2$ сходится по вероятности к $\mathbb{E}X_2$.

Функцию $g(X)$ из (19) можно представить через выборочные оценки в виде

$$g'(X) = \left(\frac{(X - \bar{X}_1) \cdot \mathbb{I}[X \leq \bar{X}] + (X - \bar{X}_2) \cdot \mathbb{I}[X > \bar{X}]}{\bar{X}_1 - \bar{X}_2} \right)^2.$$

Тогда по закону больших чисел

$$\frac{1}{n} \sum_{i=1}^n g'(X_i) \xrightarrow{P} \mathbb{E}_X(g(X)) = SW(X) \text{ при } n \rightarrow \infty.$$

□

Формулу для SW -характеристики можно переписать через моменты величин X , X_1 и X_2 .

Теорема 1. Для случайной величины X её SW -характеристика может быть найдена по формуле:

$$SW(X) = \frac{\mathbb{E}X^2 - J_1\mathbb{E}^2X_1 - J_2\mathbb{E}^2X_2}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2}.$$

Доказательство. По определению (20) SW -характеристики:

$$SW(X) = \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt + \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt \right) = \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} (G_1 + G_2).$$

Раскроем квадрат разности в G_1 :

$$\begin{aligned} G_1 &= \int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt = \int_{-\infty}^{\mathbb{E}X} (t^2 - 2t\mathbb{E}X_1 + \mathbb{E}^2X_1) f_X(t) dt = \\ &= \int_{-\infty}^{\mathbb{E}X} t^2 f_X(t) dt - 2\mathbb{E}X_1 \int_{-\infty}^{\mathbb{E}X} t f_X(t) dt + \mathbb{E}^2X_1 \int_{-\infty}^{\mathbb{E}X} f_X(t) dt. \end{aligned}$$

Интеграл определён на промежутке $(-\infty; \mathbb{E}X]$. Поэтому $f_X(t)$ можно заменить на $J_1 f_{X_1}(t)$ в соответствии с (17):

$$G_1 = \int_{-\infty}^{\mathbb{E}X} t^2 f_X(t) dt - 2J_1\mathbb{E}X_1 \int_{-\infty}^{+\infty} t f_{X_1}(t) dt + J_1\mathbb{E}^2X_1 \int_{-\infty}^{+\infty} f_{X_1}(t) dt = \int_{-\infty}^{\mathbb{E}X} t^2 f_X(t) dt - J_1\mathbb{E}^2X_1.$$

Выполним аналогичные преобразования в G_2 :

$$G_2 = \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt = \int_{\mathbb{E}X}^{+\infty} (t^2 - 2t\mathbb{E}X_2 + \mathbb{E}^2X_2) f_X(t) dt = \int_{\mathbb{E}X}^{+\infty} t^2 f_X(t) dt - J_2\mathbb{E}^2X_2.$$

Подставим G_1 и G_2 в формулу SW -характеристики и в результате получим:

$$\begin{aligned} SW(X) &= \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} (G_1 + G_2) = \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^{\mathbb{E}X} t^2 f_X(t) dt + \int_{\mathbb{E}X}^{+\infty} t^2 f_X(t) dt - J_1\mathbb{E}^2X_1 - J_2\mathbb{E}^2X_2 \right) = \\ &= \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^{+\infty} t^2 f_X(t) dt - J_1\mathbb{E}^2X_1 - J_2\mathbb{E}^2X_2 \right) = \frac{\mathbb{E}X^2 - J_1\mathbb{E}^2X_1 - J_2\mathbb{E}^2X_2}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2}. \end{aligned}$$

□

Вычислим второй момент гомоскедастичной смеси двух гауссиан $\mathbb{N}_2$. Для этого воспользуемся формулой (1):

$$\begin{aligned} \mathbb{E}X^2 &= \int_{-\infty}^{+\infty} t^2 f_X(t) dt = \int_{-\infty}^{+\infty} t^2 \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} + \frac{1-p}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = p \int_{-\infty}^{+\infty} t^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt + \\ &+ (1-p) \int_{-\infty}^{+\infty} t^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt = p(\sigma^2 + m_1^2) + (1-p)(\sigma^2 + m_2^2) = \sigma^2 + pm_1^2 + (1-p)m_2^2. \end{aligned} \quad (25)$$

Если использовать оценки параметров распределения (14), (15), (16), то можно заметить сходство между вторым моментом распределения $\mathbb{N}_2$ и величиной $J_1 \mathbb{E}^2 X_1 + J_2 \mathbb{E}^2 X_2$ в теореме 1. Можно сказать, что SW -характеристика некоторым образом оценивает величину $\frac{\sigma^2}{(m_1 - m_2)^2}$, что является обратным значением для числа $\frac{D^2}{4}$ из теоремы о бимодальности. Это ещё раз подчёркивает связь SW -характеристики с бимодальностью распределения. Чем $SW(X)$ меньше, тем более выраженную бимодальность имеет распределение величины X .

Лемма 3. SW -характеристика неотрицательна для любой случайной величины X и равна 0 тогда и только тогда, когда функцию плотности распределения $f_X(x)$ можно считать смесью двух δ -функций

$$f_X(x) = p\delta(x - \mathbb{E}X_1) + (1-p)\delta(x - \mathbb{E}X_2), \text{ где } \delta(x) = 0 \text{ для } \forall x \neq 0 \text{ и } \int_{-\infty}^{+\infty} \delta(x) dx = 1.$$

Доказательство. Функция $g(X)$, задаваемая формулой (19), неотрицательна на всей прямой $(-\infty; +\infty)$. Следовательно, SW -характеристика, равная по определению $\mathbb{E}_X(g(X))$, также неотрицательна.

Пусть $SW(X) = 0$. По формуле (20) это означает, что

$$\begin{aligned} \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt + \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt \right) &= 0; \Leftrightarrow \int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt + \\ &+ \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt = 0; \Leftrightarrow \int_{-\infty}^{\mathbb{E}X} (t - \mathbb{E}X_1)^2 f_X(t) dt = \int_{\mathbb{E}X}^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt = 0. \end{aligned} \quad (26)$$

Теперь, если в выражении (26) от величины X перейти к X_1 и X_2 в соответствии с формулами (17) и (18), то получим

$$J_1 \int_{-\infty}^{+\infty} (t - \mathbb{E}X_1)^2 f_{X_1}(t) dt = J_2 \int_{-\infty}^{+\infty} (t - \mathbb{E}X_2)^2 f_{X_2}(t) dt = 0; \Leftrightarrow \forall X_1 = \forall X_2 = 0.$$

Итак, в случае нулевой характеристики $SW(X)$ дисперсии величин X_1 и X_2 равны 0. Отсюда следует [24], что величины X_1 и X_2 являются константами и равны своим ожиданиям $\mathbb{E}X_1$ и $\mathbb{E}X_2$. Саму величину X тогда можно записать в виде:

$$X = \begin{cases} \mathbb{E}X_1, & X \leq \mathbb{E}X, \\ \mathbb{E}X_2, & X > \mathbb{E}X. \end{cases}$$

Функцию плотности распределения величины X можно представить через смесь δ -функций с параметром p , равным $\mathbb{P}(X \leq \mathbb{E}X)$:

$$f_X(x) = p\delta(x - \mathbb{E}X_1) + (1-p)\delta(x - \mathbb{E}X_2).$$

□

Лемма 3 показывает, как SW -характеристика связана с импульсными сигналами. Она равна 0 только для величин, имеющих распределение такое же, как у идеального биполярного импульсного сигнала (рис. 5). Кроме того, SW -характеристика неотрицательна. Если считать, что при возрастании SW -характеристика соответствует сигналам всё менее и менее похожим на биполярный импульсный шум, то SW -характеристику можно использовать для обнаружения таких шумов. Детектировать биполярные импульсные шумы в данных можно было бы по малому значению SW -характеристики. Насколько эта идея оправдана, будет обсуждено в последующих разделах статьи. Но эту же идею можно взять за основу для критерия обнаружения прямоугольных сигналов, являющихся частным случаем биполярных импульсных шумов. Для этого необходимо определить, как считать SW -характеристику для прямоугольного шума. Следующая теорема даёт ответ на данный вопрос.

Теорема 2. Пусть $Y_1, Y_2, \dots, Y_n$ — независимые случайные величины с распределением $Y_t = s(t) + \varepsilon_t$, где $\varepsilon_t \sim \mathbb{N}(0; \sigma^2)$, Z — величина с законом распределения $\mathbb{N}_2(m_1, m_2, \sigma^2, p)$. Тогда $\widehat{SW}(Y_1, Y_2, \dots, Y_n)$ сходится по вероятности к $SW(Z)$ при $n \rightarrow \infty$.

Доказательство. Формула (21) оценки SW -характеристики представляет собой сумму случайных величин. Это означает, что порядок величин $Y_1, Y_2, \dots, Y_n$ при суммировании становится не важен. Данный факт позволяет при вычислении $\widehat{SW}(Y_1, Y_2, \dots, Y_n)$ рассматривать $Y_1, Y_2, \dots, Y_n$ как набор независимых величин, одинаково распределённых с Z . Покажем, как происходит такой переход при вычислении $\bar{Y}$.

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i = \frac{1}{n} \sum_{i=1}^n Y_i \cdot \mathbb{I}[s(i) = m_1] + \frac{1}{n} \sum_{i=1}^n Y_i \cdot \mathbb{I}[s(i) = m_2]. \quad (27)$$

Рассмотрим величину $Y_i \cdot \mathbb{I}[s(i) = m_1]$. Она равна Y_i с вероятностью $\mathbb{P}(s(i) = m_1)$ и равна 0 с вероятностью $1 - \mathbb{P}(s(i) = m_1)$. При суммировании порядок величин становится не важен, поэтому вероятность $\mathbb{P}(s(i) = m_1)$ можно вычислить как отношение количества точек i , в которых $s(i)$ равна m_1 , к общему числу точек n . Если количество точек n кратно периоду T , то трудностей при вычислении вероятности $\mathbb{P}(s(i) = m_1)$ не возникает. В случае n , не кратного T , в последнем периоде точки будут иначе распределены, чем во всех предыдущих, но в силу того, что $n \rightarrow \infty$, это обстоятельство нивелируется и вероятность всё равно можно считать как в случае n , кратного периоду. Количество точек, в которых $s(i) = m_1$, равно числу периодов, умноженному на количество точек в периоде и на число p точек в периоде, где $s(i) = m_1$. Тогда вероятность $\mathbb{P}(s(i) = m_1)$ может быть найдена как

$$\mathbb{P}(s(i) = m_1) = \frac{\frac{n}{T} \cdot T \cdot p}{n} = p.$$

Величины $Y_i \cdot \mathbb{I}[s(i) = m_1]$ и $Y_i \cdot \mathbb{I}[s(i) = m_2]$ в (27) можно заменить на величины Z_i :

$$Z_i \sim \begin{cases} \mathbb{N}(m_1; \sigma^2), & \text{с вероятностью } p, \\ \mathbb{N}(m_2; \sigma^2), & \text{с вероятностью } 1 - p. \end{cases}$$

По закону больших чисел

$$\bar{Y} = \bar{Z} \xrightarrow{P} \mathbb{E}Z \text{ при } n \rightarrow \infty.$$

Аналогичное преобразование можно выполнить во всех суммах, фигурирующих в (21). $\square$

Теорема 2 связывает гомоскедастичную смесь двух гауссиан с прямоугольным шумом с помощью SW -характеристики и все проведённые в данном разделе рассуждения позволяет перенести

на прямоугольные сигналы. При игнорировании порядка отчётов $Y_1, Y_2, \dots, Y_n$ сигнал перестаёт обязательно содержать продолжительные фрагменты, но такое допущение позволяет рассматривать прямоугольный сигнал как частный случай биполярного импульсного сигнала. SW -характеристика для прямоугольного шума будет оценивать «выраженность» прямоугольной структуры в виде двух продолжительных импульсов, а нулевое значение характеристики будет соответствовать идеальному прямоугольному сигналу.

При $SW(X) = 0$ можно говорить об идеальном биполярном импульсном сигнале и, в частности, идеальном прямоугольном сигнале. Попробуем найти критические значения SW -характеристики, которые сигнализируют об окончательной утрате сигналом биполярности, а значит и прямоугольности. Для этого исследуем, чему равна SW -характеристика в нормальном распределении и в распределении смеси гауссиан.

7. SW -характеристика нормального распределения

В данном разделе будем считать, что величина X распределена нормально по закону $\mathbb{N}(0; \sigma^2)$.

Теорема 3. Пусть $X \sim \mathbb{N}(0; \sigma^2)$. SW -характеристика для величины X равна

$$SW(X) = \frac{\pi}{8} - \frac{1}{4}.$$

Доказательство. Математическое ожидание $\mathbb{E}X$ равно 0. Вычислим $\mathbb{E}X_1$ и $\mathbb{E}X_2$, используя формулы (17) и (18).

Если $X \sim \mathbb{N}(0, 1)$, то функция распределения $F_X(x)$ случайной величины X может быть вычислена как $F_X(x) = \frac{1}{2} + \int_0^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{1}{2} + \Phi(x)$, где функция $\Phi(X)$ называется *функцией Лапласа* [24]. Функция Лапласа обладает рядом полезных свойств. Для нахождения констант J_1 и J_2 будет полезно следующее свойство функции Лапласа:

$$\lim_{x \rightarrow +\infty} \Phi(x) = - \lim_{x \rightarrow -\infty} \Phi(x) = \frac{1}{2}.$$

Пользуясь этим свойством, найдём J_1 :

$$J_1 = \int_{-\infty}^{\mathbb{E}X} f_X(t) dt = \int_{-\infty}^0 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} dt = \left| \begin{matrix} l = \frac{t}{\sigma} \\ dl = \frac{1}{\sigma} dt \end{matrix} \right| = \int_{-\infty}^0 \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl = - \int_0^{-\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl = -\Phi(-\infty) = \frac{1}{2}.$$

Аналогично константа J_2 получается также равной $1/2$.

Числа J_1 и J_2 равны, следовательно, функции $f_{X_1}(t)$ и $f_{X_2}(t)$ будут иметь один и тот же вид

$$f_{X_1}(t) = \begin{cases} \frac{2}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}, & t \in (-\infty; 0], \\ 0, & t \in (0; +\infty). \end{cases} \quad \text{и} \quad f_{X_2}(t) = \begin{cases} \frac{2}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}, & t \in (0; +\infty), \\ 0, & t \in (-\infty; 0]. \end{cases}$$

Найдём числа $\mathbb{E}X_1$ и $\mathbb{E}X_2$.

$$\begin{aligned} \mathbb{E}X_1 &= \int_{-\infty}^{+\infty} t f_{X_1}(t) dt = \int_{-\infty}^0 \frac{2t}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} dt = \left| \begin{matrix} l = t^2 \\ dl = 2t dt \end{matrix} \right| = \int_{+\infty}^0 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{l}{2\sigma^2}} dl = \\ &= \left| \begin{matrix} s = \frac{l}{2\sigma^2} \\ ds = \frac{1}{2\sigma^2} dl \end{matrix} \right| = \int_{+\infty}^0 \frac{2\sigma}{\sqrt{2\pi}} e^{-s} ds = - \frac{2\sigma}{\sqrt{2\pi}} e^{-s} \Big|_{+\infty}^0 = - \frac{2\sigma}{\sqrt{2\pi}} (e^0 - e^{-\infty}) = - \frac{2\sigma}{\sqrt{2\pi}}. \end{aligned}$$

Вычисление $\mathbb{E}X_2$ производится аналогично $\mathbb{E}X_1$

$$\mathbb{E}X_2 = \int_{-\infty}^{+\infty} t f_{X_2}(t) dt = \int_0^{+\infty} \frac{2t}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} dt = \left| \begin{array}{l} l = t^2 \\ dl = 2t dt \end{array} \right| = \int_0^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{l}{2\sigma^2}} dl = -\frac{2\sigma}{\sqrt{2\pi}} e^{-s} \Big|_0^{+\infty} = \frac{2\sigma}{\sqrt{2\pi}}.$$

Посчитаем SW -характеристику для X по определению 1:

$$\begin{aligned} SW(X) &= \int_{-\infty}^{+\infty} g(t) f_X(t) dt = \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^0 (t - \mathbb{E}X_1)^2 f_X(t) dt + \int_0^{+\infty} (t - \mathbb{E}X_2)^2 f_X(t) dt \right) = \\ &= \frac{1}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(J_1 \int_{-\infty}^0 (t^2 - 2t\mathbb{E}X_1 + \mathbb{E}^2X_1) f_{X_1}(t) dt + J_2 \int_0^{+\infty} (t^2 - 2t\mathbb{E}X_2 + \mathbb{E}^2X_2) f_{X_2}(t) dt \right) = \\ &= \frac{1}{2(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^0 t^2 f_{X_1}(t) dt - 2\mathbb{E}X_1 \cdot \mathbb{E}X_1 + \mathbb{E}^2X_1 + \int_0^{+\infty} t^2 f_{X_2}(t) dt - 2\mathbb{E}X_2 \cdot \mathbb{E}X_2 + \mathbb{E}^2X_2 \right) = \\ &= \frac{1}{2(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(\int_{-\infty}^0 t^2 f_{X_1}(t) dt - \mathbb{E}^2X_1 + \int_0^{+\infty} t^2 f_{X_2}(t) dt - \mathbb{E}^2X_2 \right) = \\ &= \frac{1}{2(\mathbb{E}X_1 - \mathbb{E}X_2)^2} \left(2 \int_{-\infty}^{+\infty} t^2 f_X(t) dt - \mathbb{E}^2X_1 - \mathbb{E}^2X_2 \right) = \frac{2\sigma^2 - \frac{4\sigma^2}{\pi}}{\frac{16\sigma^2}{\pi}} = \frac{\pi}{8} - \frac{1}{4}. \end{aligned}$$

□

SW -характеристика нормального распределения равна числу $\frac{\pi}{8} - \frac{1}{4}$. Этот результат позволяет с помощью SW -характеристики сравнивать, насколько распределение смеси двух гауссиан соответствует распределению одного единственного гауссиана. Фактически это и есть то самое «критическое значение», соответствующее моменту, когда два выраженных биполярных импульса окончательно слились в один импульс.

8. SW -характеристика распределения $\mathbb{N}_2$ и SW -критерий обнаружения прямоугольных сигналов

Теперь вычислим SW -характеристику для смеси гауссиан $\mathbb{N}_2(m_1, m_2, \sigma^2, p)$.

Теорема 4. Пусть $X \sim \mathbb{N}_2(m_1, m_2, \sigma^2, p)$, $m_2 \geq m_1$, $r = \frac{m_2 - m_1}{\sigma}$, тогда SW -характеристика для величины X равна

$$\begin{aligned} SW(X) &= J_2^2 \frac{1 + r^2(1-p)p}{(\Theta_1 - r(1-p))^2} - J_1 J_2, \quad \text{где} \\ \Theta_1 &= \frac{1}{J_1} \left(-\frac{1}{\sqrt{2\pi}} \left(p e^{-\frac{(p-1)^2 r^2}{2}} + (1-p) e^{-\frac{p^2 r^2}{2}} \right) + r p \int_{-\infty}^{-(p-1)r} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl \right), \\ J_1 &= p \int_{-\infty}^{-(p-1)r} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl + (1-p) \int_{-\infty}^{-pr} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl, \quad J_2 = 1 - J_1. \end{aligned} \tag{28}$$

Доказательство. См. приложение А.

□

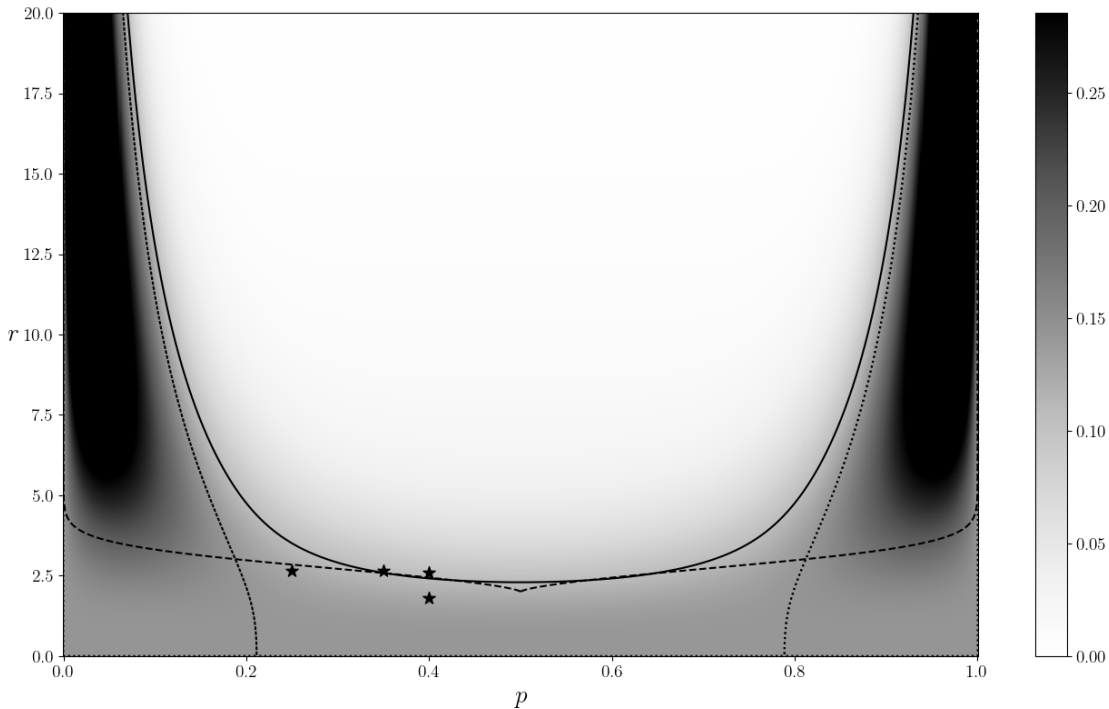


Fig. 9. The heat map of the SW'' -characteristic for the $\mathbb{N}_2$ distribution. The curve specified by the condition (2) of bimodality theorem is shown as a dash, a dotted line corresponds to the number 0,1427, a solid line — to the number 0,1077.

Рис. 9. Тепловая карта SW -характеристики распределения $\mathbb{N}_2$. Пунктиром обозначена кривая, задаваемая условием (2) из теоремы о бимодальности, линия из точек соответствует числу 0,1427, сплошная линия — числу 0,1077.

Из теоремы 4 следует, что для распределения $\mathbb{N}_2$ формулу SW -характеристики можно разрешить относительно всего 2 переменных r и p . Это позволяет исследовать SW -характеристику для смеси двух гауссиан графически.

Переменная r — это практически тот же параметр, что фигурировал в [теореме о бимодальности](#), но умноженный на 2, то есть $r = 2D$. Таким образом, SW -характеристика смеси двух гауссиан зависит от параметров, определяющих бимодальность распределения. Изучим поведение SW -характеристики для распределения $\mathbb{N}_2$ в зависимости от значений r и p .

Рисунок 9 представляет собой тепловую карту SW -характеристики для распределения $\mathbb{N}_2$. По оси Ox отложены значения параметра $p \in [0; 1]$, по оси Oy — значения параметра $r \geq 0$. Белому цвету соответствуют значения SW -характеристики близкие к 0, чёрному — значительно превышающие 0.

В соответствии с теоремой 3 SW -характеристика для нормального распределения равна $\frac{\pi}{8} - \frac{1}{4} \approx 0,1427$. Нормальное распределение унимодально. Если SW -характеристика смеси гауссиан $\mathbb{N}_2$ близка к данному числу, то можно считать, что распределение уже однозначно перестало быть бимодальным. Значения SW -характеристики, равные 0,1427, соединены на рис. 9 линией из точек. Тривиальные расположения этой линии при $p = 0$, $p = 1$ и $r = 0$ соответствует случаю идеального нормального распределения, когда в смеси присутствует только один гауссиан. Большой интерес представляют линии из точек, отделяющие белую область от областей с насыщенным чёрным цветом. Такие случаи смеси двух гауссиан [теорема о бимодальности](#) считает бимодальными (расположены выше пунктирной линии). При этом эти виды распределений никак нельзя назвать соответствующими распределению прямоугольных сигналов. В таких случаях распределение выглядит как один единственный гауссиан с левым или правым «тяжёлым хвостом» (см., например,

мгновенный импульсный шум на рис. 15). Таким образом, значение 0,1427 SW -характеристики позволяет обнаруживать не только случай одного гауссиана в смеси, но и более сложный случай, когда гауссиан всё ещё один, но имеет «тяжёлый хвост», артефакт второго гауссиана. Поведение SW -характеристики распределения $\mathbb{N}_2$ в «чёрных областях» заслуживает отдельного исследования. В данной работе ограничимся уже сделанными замечаниями.

Пунктирной линией на рис. 9 изображена кривая, разграничивающая бимодальный и унимодальный случай в распределении. Она задана с помощью уравнения из свойства (2) **теоремы о бимодальности**. Звёздочками на рис. 9 отмечены случаи, соответствующие рисункам 7 и 8 ($D = \frac{\tau}{2}$). Некоторые из этих случаев унимодальны, хотя и не имеют форму идеального гауссиана, то есть SW -характеристика нормального распределения 0,1427 их не отлавливает. Но если взять значение SW -характеристики как некоторое $\omega < 0,1427$, такое, чтобы все значения, меньшие ω лежали на рис. 9 выше пунктирной линии, то случаи, когда SW -характеристика попадает в промежуток $(0; \omega)$, будут считаться бимодальными. При этом в чёрные области такие случаи попадать не будут. На рис. 9 сплошной линией изображены значения SW -характеристики, равные 0,1077. Эта линия соприкасается снизу с пунктирной линией. Таким образом, по одну сторону от этой линии располагаются только бимодальные случаи смеси двух гауссиан. На рис. 9 также видно, что сплошная линия разделяет значения SW -характеристики, меньшие 0,1077, от значений, больших 0,1077. Это значение характеристики можно воспринимать как критическое, при достижении которого говорить о наличии прямоугольных сигналов в данных уже нельзя.

Итак, выбрав некоторое критическое значение $\omega \in (0; 0,1427)$, мы можем сравнить с этим значением SW -характеристику и, если она окажется меньше ω , делать вывод, что в данных могут присутствовать прямоугольные сигналы. С помощью такого простого правила определяется **SW -критерий для обнаружения прямоугольных сигналов**. Критическое значение ω для SW -характеристики подбирается эмпирически в зависимости от анализируемых данных.

Напомним, что в соответствии с леммой 1 распределение биполярных импульсов будет сохраняться при их суммировании с нормально распределённой величиной. Это позволяет применять SW -критерий в любых данных, о нормальном распределении которых можно говорить достаточно уверенно. В частности, большая часть данных вихретоковых дефектограмм представлена рельсовым нормально распределённым шумом, поэтому применение SW -критерия более чем оправдано. Распределение сигналов конструктивных элементов в дефектограммах не является нормальным, но ввиду того, что они возникают достаточно редко (менее 10 % сигналов [2]) и часто имеют малую протяжённость, форма нормального распределения искажается незначительно.

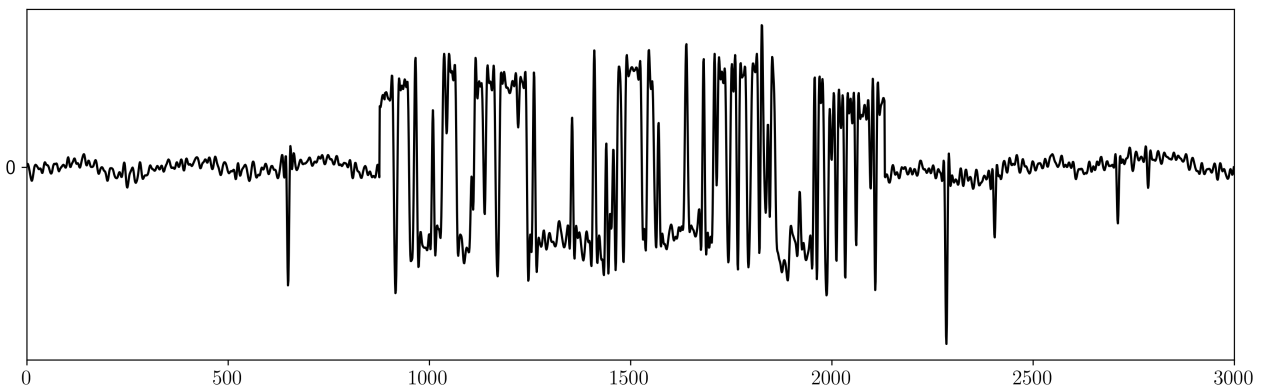


Fig. 10. Local additions of square wave signals on the eddy current defectogram

Рис. 10. Локальные примеси прямоугольных сигналов на вихретоковой дефектограмме

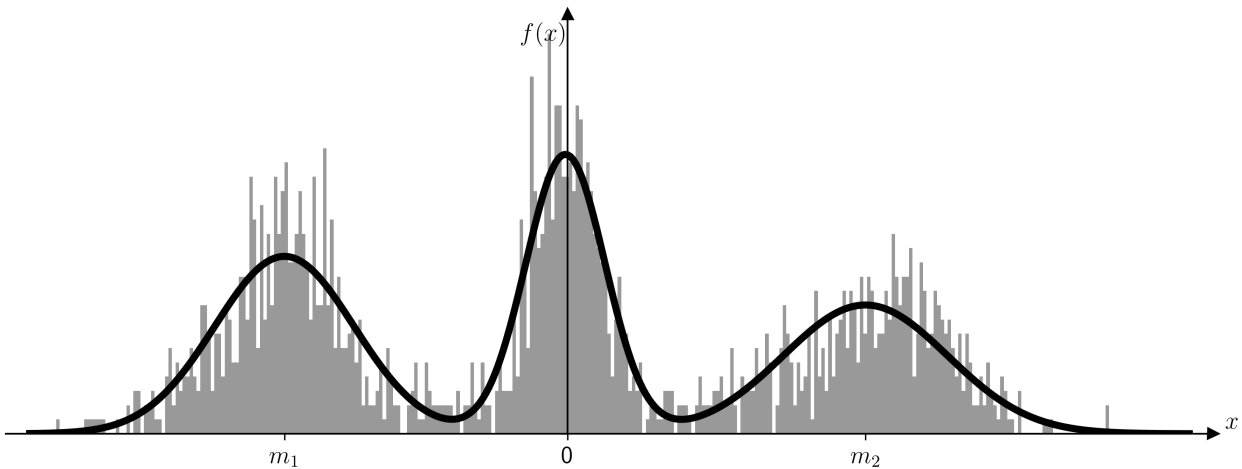


Fig. 11. The distribution density function of local additions of square wave signals on the eddy current defectogram

Рис. 11. Функция плотности распределения локальной примеси прямоугольных сигналов на вихретоковой дефектограмме

Table 1. The comparative table of statistical characteristics used for detection of square wave signals in eddy current defectogram data

Таблица 1. Сравнительная таблица статистических характеристик, используемых для детекции прямоугольных сигналов в данных вихретоковых дефектограмм

Figure	SW	BIC for 1 Gaussian	BIC for 2 Gaussians	Optimal amount	Optimal BIC	MDE
12	0,0220	2,8425	1,5863	9	0,8113	3.7840
13	0,0201	2,8413	1,4877	8	1,1393	3.6726
14	0,0239	2,7968	1,7124	7	1,4284	3.7242
15	0,2425	2,8389	1,9133	9	1,2228	3.6111
16	0,1488	2,8392	2,8467	1	2,8392	5.4550

Протяжённые сигналы, распределение которых не является нормальным, вносят существенные изменения в распределение, и гомоскедастичная смесь двух гауссиан перестаёт сохраняться. Этим определяется одно из ограничений SW -критерия, основанного на свойствах SW -характеристики распределения N_2 .

Ещё одним важным ограничением на применение SW -критерия является *локальность* аддитивности прямоугольных сигналов. Если прямоугольные сигналы возникают локально, только в некоторой части данных, распределённых нормально, то распределение всей совокупности данных с локальными примесями прямоугольных сигналов будет представлять собой смесь трёх, а не двух гауссиан. Новый гауссиан будет возникать как артефакт исходного нормального распределения данных (см. рис. 10 и 11). Локальность SW -критерия может быть преодолена использованием скользящего окна. Конкретные реализации оконного SW -критерия заслуживают отдельного исследования. Здесь они не обсуждаются.

9. Примеры применения SW -критерия

В завершение будут рассмотрены несколько примеров поиска прямоугольных сигналов в реальных данных вихретоковых дефектограмм. Все рассматриваемые сигналы нормированы по выборочной дисперсии.

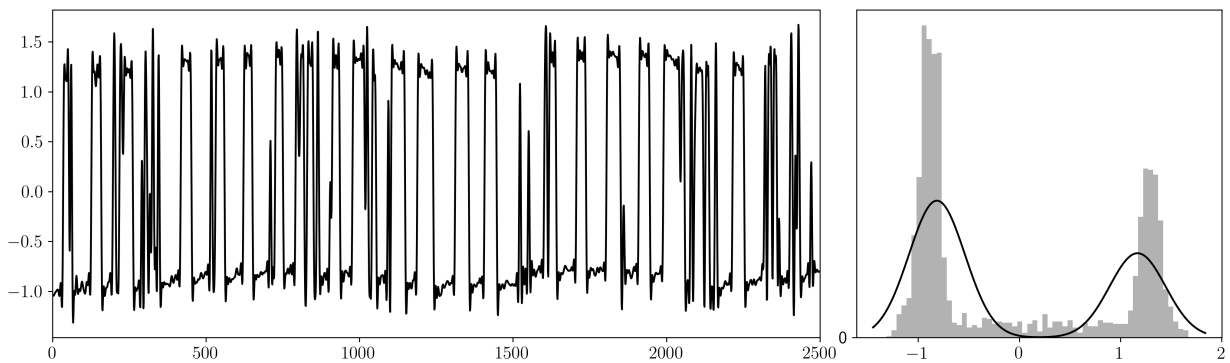


Fig. 12. The common square wave signal on the eddy current defectogram

Рис. 12. Стандартный прямоугольный сигнал на вихретоковой дефектограмме

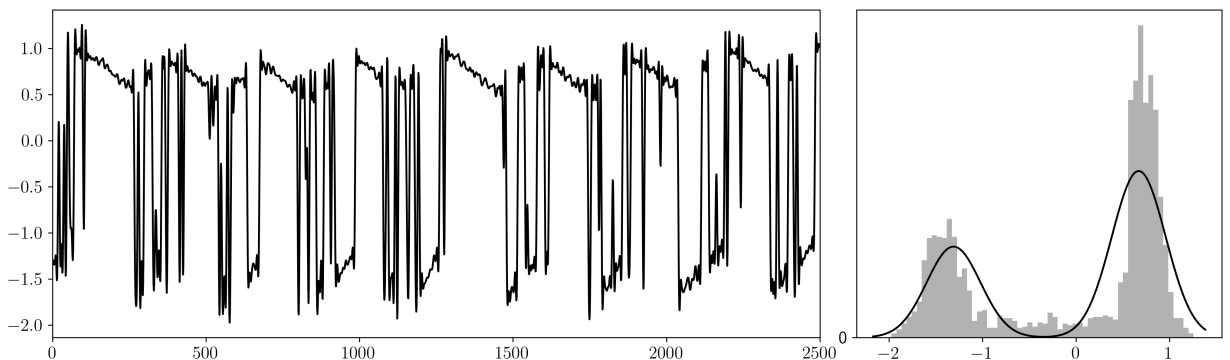


Fig. 13. The inclined square wave signal on the eddy current defectogram

Рис. 13. Наклонный прямоугольный сигнал на вихретоковой дефектограмме

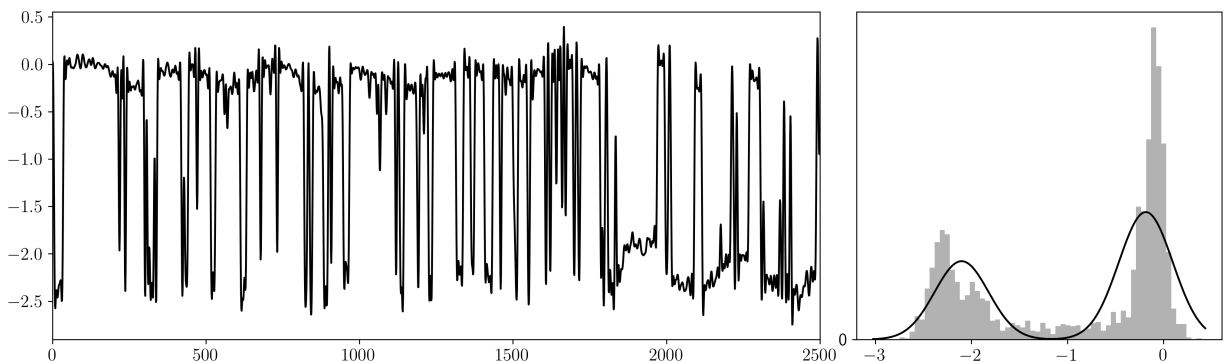


Fig. 14. The square wave signal with variable pulsation on the eddy current defectogram

Рис. 14. Прямоугольный сигнал с переменной пульсацией на вихретоковой дефектограмме

В таблице 1 приведены результаты эксперимента для 5 видов данных вихретоковых дефектограмм. Помимо вычисления SW -характеристики к данным применяется EM -алгоритм и считается сумма многомасштабных дисперсных энтропий (MDE) по трём масштабам, которая использовалась для обнаружения прямоугольного шума в [19]. В таблице приведены значения байесовского информационного критерия (BIC) для моделей на основе EM -алгоритма с разным количеством гауссиан. Кроме одного и двух гауссиан приводится модель с количеством гауссиан, в которой значение BIC минимально (оптимальная модель).

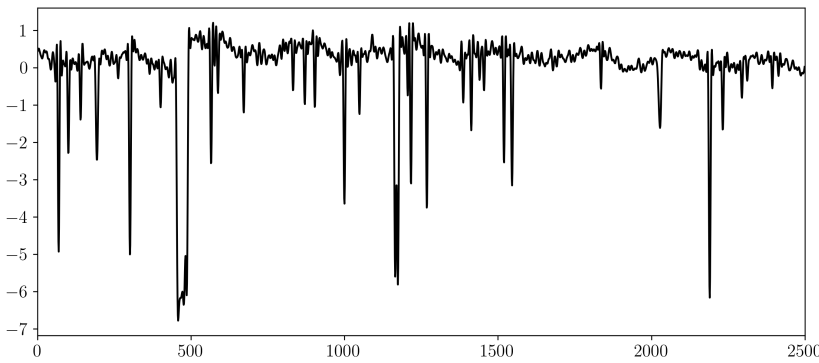


Fig. 15. The impulse instant noise on the eddy current defectogram

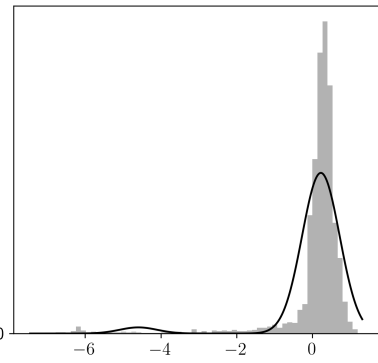


Рис. 15. Импульсный мгновенный шум на вихретоковой дефектограмме

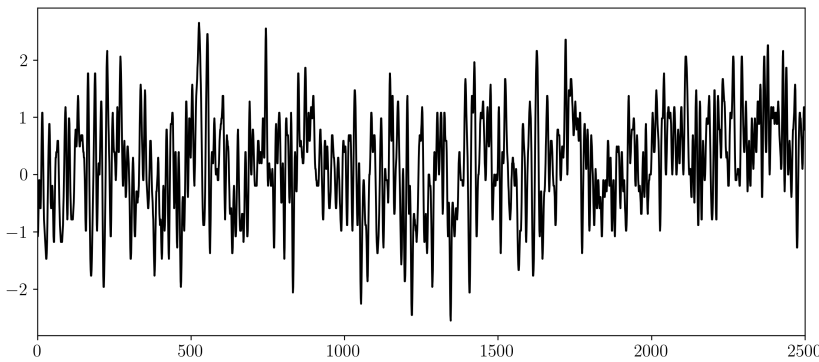


Fig. 16. The rail noise on the eddy current defectogram

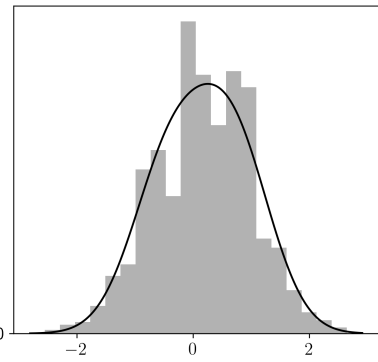


Рис. 16. Рельсовый шум на вихретоковой дефектограмме

В 3 из 5 рассмотренных случаев, где действительно присутствуют прямоугольные сигналы, SW -характеристика принимает значения в 7,5 раз меньше, чем число 0,1427, которое, как было показано в разделе 7, характерно для одного гауссиана. Это можно расценивать как свидетельство о том, что в данных присутствуют прямоугольные сигналы. Значение информационного критерия во всех примерах с прямоугольными сигналами в модели для двух гауссиан меньше, чем в модели с одним гауссианом. При этом для случая импульсного мгновенного шума (рис. 15) BIC для одного гауссиана больше, чем для двух, тогда как в распределении имеется только один ярко выраженный гауссиан. Значение MDE в зашумлённых данных меньше, чем в чистых, но наименьшим значением обладают не прямоугольные, а импульсные мгновенные шумы. Таким образом, утверждение, выдвинутое в [19], о том, что прямоугольный шум обладает наименьшим значением MDE , не подтверждается.

В примерах, где прямоугольные сигналы отсутствуют, SW -характеристика получилась большой: $0,1488 > 0,1427$ — для рельсового шума, и $0,2425$ — для импульсного мгновенного шума.

Заключение

Прямоугольные сигналы являются примером биполярных импульсных сигналов, которые, в свою очередь, обладают специфическим видом вероятностного распределения. Этот вид распределения хорошо описывается гомоскедастичной смесью двух гауссиан. Ввиду того, что прямоугольный сигнал имеет выраженную полярность импульсации, смесь гауссиан должна быть бимодальной. Таким образом, задачу обнаружения прямоугольных сигналов в данных можно рассматривать как составную задачу проверки соответствия распределения данных гомоскедастичной бимодальной смеси

двух гауссиан. Лемма 1 показывает, что при условии нормальности данных примешанные к ним биполярные импульсные сигналы будут сохранять свою специфическую форму распределения.

Естественным методом проверки соответствия распределения данных заданному распределению является оценка функции правдоподобия. Таковая оценка была произведена в разделе 5 и в дальнейшем использована для определения *SW*-характеристики. *SW*-характеристика вводится как показатель «прямоугольности» данных. Доказано, что она равна 0 только для величин, имеющих распределение идеального биполярного импульсного сигнала (Лемма 3). Лемма 2 определяет формулу оценки *SW*-характеристики по выборке. Теорема 1 предоставляет ещё одну формулу для вычисления *SW*-характеристики через второй момент. Наконец, теорема 2 связывает распределение биполярных импульсных шумов с прямоугольными. *SW*-характеристика позволяет рассматривать прямоугольный шум как частный случай биполярного импульсного шума, для которого продолжительная импульсация не имеет обязательного характера.

SW-характеристика была подсчитана для нормального распределения (теорема 3) и для гомоскедастичной смеси двух гауссиан (теорема 4). Рисунок 9 демонстрирует, что *SW*-характеристика позволяет различать бимодальную и одномодальную смеси гауссиан. Когда *SW*-характеристика смеси гауссиан становится близка к *SW*-характеристике нормального распределения, то бимодальность смеси совершенно исчезает. Такое простое верхнее ограничение *SW*-характеристики, кроме всего прочего, позволяет отделять продолжительный прямоугольный шум от мгновенного импульсного шума в некоторых ситуациях. Таким образом, возникает естественное определение критерия обнаружения прямоугольных сигналов на основе *SW*-характеристики (*SW*-критерий) — сравнить значение характеристики с некоторым верхним порогом, не превышающим значение *SW*-характеристики для нормального распределения и позволяющим отделить только бимодальные случаи.

SW-критерий оказывается достаточно эффективным в обнаружении прямоугольных сигналов в данных вихретоковых дефектограмм, которые преимущественно представлены нормальным рельсовым шумом. В завершение приведены результаты эксперимента применения разных критериев обнаружения прямоугольных сигналов в вихретоковых дефектограммах. Во всех примерах *SW*-критерий дал верный ответ, из чего можно сделать вывод, что применение *SW*-критерия может быть достаточно эффективным при решении задачи выявления прямоугольных шумов в данных вихретоковых дефектограмм. Кроме того, по сравнению с другими методами критерий обладает высокой объясняющей способностью, так как значения *SW*-характеристики имеют прямую связь с бимодальностью, правдоподобием распределения импульсных сигналов, а также идеальным биполярным импульсным сигналом и, соответственно, идеальным прямоугольным сигналом. Таким образом, обнаружение прямоугольного шума с помощью *SW*-характеристики может быть оправданным не только для вихретоковых дефектограмм, но и для других видов данных.

References

- [1] K. V. Vlasov and A. L. Bobrov, “Influence of object physical properties instability on edge current method sensitivity”, *Vestnik IzhGTU imeni M. T. Kalashnikova*, vol. 27, no. 1, pp. 55–62, 2024, in Russian. doi: [10.22213/2413-1172-2024-1-55-62](https://doi.org/10.22213/2413-1172-2024-1-55-62).
- [2] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, and V. A. Tyukin, “Finding the level of useful signals on interpretation of magnetic and eddy-current defectograms”, *Automatic Control and Computer Sciences*, vol. 52, pp. 658–666, 2018. doi: [10.3103/S0146411618070179](https://doi.org/10.3103/S0146411618070179).
- [3] E. V. Kuzmin *et al.*, “Application of convolutional neural networks for recognizing long structural elements of rails in eddy current defectograms”, *Automatic Control and Computer Sciences*, vol. 55, pp. 712–722, 2021. doi: [10.3103/S0146411621070099](https://doi.org/10.3103/S0146411621070099).

- [4] E. V. Kuzmin *et al.*, “Assessing flaw severity on interpretation of eddy-current defectograms”, *Automatic Control and Computer Sciences*, vol. 56, pp. 723–734, 2023. DOI: [10.3103/S0146411622070124](https://doi.org/10.3103/S0146411622070124).
- [5] L. Y. Bystrov, A. N. Gladkov, and E. V. Kuzmin, “Suppression of additive periodic low-frequency interference on eddy current defectograms”, *Modeling and Analysis of Information Systems*, vol. 31, no. 2, pp. 164–181, 2024, in Russian. DOI: [10.18255/1818-1015-2024-2-164-181](https://doi.org/10.18255/1818-1015-2024-2-164-181).
- [6] P. A. Lyakhov and A. R. Orazhev, “New method for detecting and removing random-valued impulse noise from images”, *Computer Optics*, vol. 47, pp. 262–271, 2023, in Russian. DOI: [10.18287/2412-6179-CO-1145](https://doi.org/10.18287/2412-6179-CO-1145).
- [7] R. Kunsoth and M. Biswas, “Modified decision based median filter for impulse noise removal”, in *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1316–1319. DOI: [10.1109/WiSPNET.2016.7566350](https://doi.org/10.1109/WiSPNET.2016.7566350).
- [8] G. Manmadha Rao *et al.*, “Reduction of impulsive noise from speech and audio signals by using Sd-Rom algorithm”, *International Journal of Recent Technology and Engineering*, vol. 10, no. 1, pp. 265–268, 2021. DOI: [10.35940/ijrte.A5943.0510121](https://doi.org/10.35940/ijrte.A5943.0510121).
- [9] S. Pham and A. Dinh, “Adaptive-cognitive Kalman filter and neural network for an upgraded nondispersive thermopile device to detect and analyze fusarium spores”, *Sensors*, vol. 19, no. 22, p. 4900, 2021. DOI: [10.3390/s19224900](https://doi.org/10.3390/s19224900).
- [10] Y. Huo, K. Yang, Y. Qi, and T. Xu, “Robust maximum correlation entropy Kalman filtering algorithm based on S-functions under impulse noise”, *Signal, Image and Video Processing*, vol. 18, pp. 1–15, 2024. DOI: [10.1007/s11760-024-03135-y](https://doi.org/10.1007/s11760-024-03135-y).
- [11] Y. Cheng, C. Li, S. Chen, and Z. Zhou, “An enhanced impulse noise control algorithm using a novel nonlinear function”, *Circuits, Systems, and Signal Processing*, vol. 42, pp. 1–20, 2023. DOI: [10.1007/s00034-023-02421-3](https://doi.org/10.1007/s00034-023-02421-3).
- [12] C. Xing, Y. Ran, G. Tan, Q. Meng, and M. Lu, “Impulse noise mitigation and channel estimation method in OFDM systems based on TMSBL”, *IEEE Access*, vol. 12, pp. 123 376–123 387, 2024. DOI: [10.1109/ACCESS.2024.3454316](https://doi.org/10.1109/ACCESS.2024.3454316).
- [13] J. Behboodian, “On the modes of a mixture of two normal distributions”, *Technometrics*, vol. 12, no. 1, pp. 131–139, 1970. DOI: [10.1080/00401706.1970.10488640](https://doi.org/10.1080/00401706.1970.10488640).
- [14] M. Á. Carreira-Perpiñán and C. K. I. Williams, “On the number of modes of a Gaussian mixture”, *Scale Space Methods in Computer Vision*, pp. 625–640, 2003. DOI: [10.1007/3-540-44935-3_44](https://doi.org/10.1007/3-540-44935-3_44).
- [15] Y. A. Dubnov and A. V. Bulychiev, “Bayesian Identification of a Gaussian Mixture Model”, *Journal of Information Technologies and Computing Systems*, no. 1, pp. 101–111, 2017.
- [16] S. Jammalamadaka and Q. Jin, “A Bayesian test for the number of modes in a Gaussian mixture”, *Asian Journal of Earth Sciences*, vol. 1, pp. 9–22, 2021.
- [17] L. Dan, W. Xue, W. Guiqin, and Q. Zhihong, “A methodological approach for detecting burst noise in the time domain”, *World Academy of Science, Engineering and Technology*, vol. 58, pp. 974–978, Oct. 2009.
- [18] X. Chen, Y. Han, and J. Wu, “Burst noise measuring on the basis of wavelet and fourier transform”, in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, 2010, pp. 769–771. DOI: [10.1109/ICMTMA.2010.366](https://doi.org/10.1109/ICMTMA.2010.366).
- [19] R. Zhou, J. Han, Z. Guo, and T. Li, “De-noising of magnetotelluric signals by discrete wavelet transform and SVD decomposition”, *Remote Sensing*, vol. 13, no. 23, pp. 1–19, 2021. DOI: [10.3390/rs13234932](https://doi.org/10.3390/rs13234932).

- [20] X. Zhang *et al.*, “Separation of magnetotelluric signals based on refined composite multiscale dispersion entropy and orthogonal matching pursuit”, *Earth, Planets and Space*, vol. 73, pp. 1–18, 2021. DOI: [10.1186/s40623-021-01399-z](https://doi.org/10.1186/s40623-021-01399-z).
- [21] L. Zhang *et al.*, “Identification and suppression of magnetotelluric noise via a deep residual network”, *Minerals*, vol. 12, p. 766, 2022. DOI: [10.3390/min12060766](https://doi.org/10.3390/min12060766).
- [22] G. Zuo *et al.*, “Magnetotelluric noise attenuation using a deep residual shrinkage network”, *Minerals*, vol. 12, no. 9, p. 1086, 2022. DOI: [10.3390/min12091086](https://doi.org/10.3390/min12091086).
- [23] G. Li *et al.*, “Low-frequency magnetotelluric data denoising using improved denoising convolutional neural network and gated recurrent unit”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024. DOI: [10.1109/TGRS.2024.3374950](https://doi.org/10.1109/TGRS.2024.3374950).
- [24] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. John Wiley & Sons, 1970, vol. 1, p. 509.
- [25] V. Boss, *Lekcii po matematike: Veroyatnost', informaciya, statistika*. KomKniga, 2005, vol. 4, 216 pp., in Russian.
- [26] A. N. Shiryaev, *Veroyatnost'*, 3rd ed. MCNMO, 2004, vol. 1, p. 520, in Russian.
- [27] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977. DOI: [10.1111/j.2517-6161.1977.tb01600.x](https://doi.org/10.1111/j.2517-6161.1977.tb01600.x).
- [28] K. P. Burham and D. R. Anderson, *Model Selection and Multimodel Inference*, 2nd ed. Springer New York, 2002, 515 pp. DOI: [10.1007/b97636](https://doi.org/10.1007/b97636).
- [29] L. D. Kudryavcev, *Kurs matematicheskogo analiza*. Vysshaya shkola, 1981, vol. 1, 687 pp., in Russian.

A. Вывод формулы SW-характеристики

Для вывода формул используется ряд классических методов интегральных преобразований. Для ясности обозначений приведём здесь некоторые из них.

Пусть $f(x)$ непрерывна на интервале $(a; b)$, функция $\varphi(t)$ определена и непрерывна вместе со своей производной $\varphi'(t)$ на отрезке $(\alpha; \beta)$, причём для всех $t \in (\alpha; \beta)$ выполнено неравенство $a < \varphi(t) < b$. Если $a = \varphi(\alpha)$ и $b = \varphi(\beta)$, тогда справедливо равенство:

$$\int_{\alpha}^{\beta} f(\varphi(t))\varphi'(t)dt = \int_a^b f(x)dx.$$

Если функции $u = u(x)$ и $v = v(x)$ непрерывно-дифференцируемы на отрезке $(a; b)$, то [29]

$$\int_a^b u dv = uv|_a^b - \int_a^b v du.$$

Приложение посвящено доказательству следующей теоремы:

Теорема 4. Пусть $X \sim \mathbb{N}_2(m_1, m_2, \sigma^2, p)$, $m_2 \geq m_1$, $r = \frac{m_2 - m_1}{\sigma}$, тогда SW-характеристика для величины X равна

$$SW(X) = J_2^2 \frac{1 + r^2(1 - p)p}{(\Theta_1 - r(1 - p))^2} - J_1 J_2, \text{ где}$$

$$\Theta_1 = \frac{1}{J_1} \left(-\frac{1}{\sqrt{2\pi}} \left(p e^{-\frac{(p-1)^2 r^2}{2}} + (1-p) e^{-\frac{p^2 r^2}{2}} \right) + r p \int_{-\infty}^{-(p-1)r} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl \right),$$

$$J_1 = p \int_{-\infty}^{-(p-1)r} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl + (1-p) \int_{-\infty}^{-pr} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl, \quad J_2 = 1 - J_1.$$

Доказательство. Доказательство разбивается на несколько частей. В первой части производится прямое вычисление SW -характеристики для распределения $\mathbb{N}_2$. Во второй и третьей частях формула преобразуется так, чтобы она зависела только от 2 переменных. Наконец, в заключительной четвёртой части полученная формула SW -характеристики приводится к упрощённому виду, в котором она представлена в формулировке доказываемой теоремы.

Пусть $X \sim \mathbb{N}_2(m_1, m_2, \sigma^2, p)$. Вычислим $\mathbb{E}X$

$$\begin{aligned} \mathbb{E}X &= \int_{-\infty}^{+\infty} \frac{t}{\sigma\sqrt{2\pi}} \left(p e^{-\frac{(t-m_1)^2}{2\sigma^2}} + (1-p) e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = p \int_{-\infty}^{+\infty} \frac{t}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt + \\ &+ (1-p) \int_{-\infty}^{+\infty} \frac{t}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt = p m_1 + (1-p) m_2. \end{aligned}$$

Выпишем нормировочные константы функции плотности $f_{X_1}(t)$ и $f_{X_2}(t)$ с учётом найденного значения $\mathbb{E}X$

$$J_1 = \int_{-\infty}^{pm_1+(1-p)m_2} \frac{1}{\sigma\sqrt{2\pi}} \left(p e^{-\frac{(t-m_1)^2}{2\sigma^2}} \right) dt + \int_{-\infty}^{pm_1+(1-p)m_2} \frac{1}{\sigma\sqrt{2\pi}} \left((1-p) e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = J_{11} + J_{12}, \quad (29)$$

$$J_2 = \int_{pm_1+(1-p)m_2}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} \left(p e^{-\frac{(t-m_1)^2}{2\sigma^2}} \right) dt + \int_{pm_1+(1-p)m_2}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} \left((1-p) e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = J_{21} + J_{22}. \quad (30)$$

Числа $J_{11}, J_{12}, J_{21}, J_{22}$ оказываются связаны с параметром распределения p . Связующие их соотношения будут полезны в последующих преобразованиях.

$$J_{11} + J_{21} = p, \quad (31)$$

$$J_{12} + J_{22} = 1 - p. \quad (32)$$

Вычисление $\mathbb{E}X_2$ разобьём на 2 части:

$$\mathbb{E}X_2 = \frac{1}{J_2} \int_{pm_1+(1-p)m_2}^{+\infty} t \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} \right) dt + \frac{1}{J_2} \int_{pm_1+(1-p)m_2}^{+\infty} t \left(\frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = \frac{1}{J_2} (I_1 + I_2). \quad (33)$$

Интеграл I_1 в свою очередь также разобьём на 2 части и в одной из частей получим выражение с J_{21} из (30):

$$\begin{aligned} I_1 &= p \int_{pm_1+(1-p)m_2}^{+\infty} \frac{t}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = p \int_{pm_1+(1-p)m_2}^{+\infty} \frac{t - m_1 + m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = \\ &= p \int_{pm_1+(1-p)m_2}^{+\infty} \frac{t - m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt + m_1 J_{21} = I_{11} + m_1 J_{21}. \end{aligned}$$

Выполним замену $s = \frac{(t-m_1)^2}{2\sigma^2}$ в интеграле I_{11} :

$$\begin{aligned}
 I_{11} &= p \int_{pm_1+(1-p)m_2}^{+\infty} \frac{t-m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = \left| \begin{array}{lll} s = \frac{(t-m_1)^2}{2\sigma^2} & \alpha = pm_1 + (1-p)m_2 & \beta = +\infty \\ ds = \frac{t-m_1}{\sigma^2} dt & a = \frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2} & b = +\infty \end{array} \right| = \\
 &= p \int_{\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}}^{+\infty} \frac{\sigma}{\sqrt{2\pi}} e^{-s} ds = -\frac{p\sigma}{\sqrt{2\pi}} e^{-s} \Big|_{\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}}^{+\infty} = -\frac{p\sigma}{\sqrt{2\pi}} \left(e^{-\infty} - e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} \right) = \\
 &= \frac{p\sigma}{\sqrt{2\pi}} e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}}.
 \end{aligned}$$

Вычисление I_2 во многом повторяет рассуждения, применяемые при вычислении I_1 . Сначала разобьём интеграл I_2 на 2 части и в одной из частей обнаружим J_{22} из (30):

$$\begin{aligned}
 I_2 &= \int_{pm_1+(1-p)m_2}^{+\infty} \frac{(1-p)t}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt = \int_{pm_1+(1-p)m_2}^{+\infty} \frac{(1-p)(t-m_2+m_2)}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt = \\
 &= \int_{pm_1+(1-p)m_2}^{+\infty} \frac{(1-p)(t-m_2)}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt + m_2 J_{22} = I_{21} + m_2 J_{22}.
 \end{aligned}$$

Выполним замену $s = \frac{(t-m_2)^2}{2\sigma^2}$ в интеграле I_{21} . Она производится аналогично замене в интеграле I_{11} , поэтому, чтобы не повторяться, несколько сократим вычисления:

$$I_{21} = (1-p) \int_{\frac{p^2(m_1-m_2)^2}{2\sigma^2}}^{+\infty} \frac{\sigma}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds = \frac{(1-p)\sigma}{\sqrt{2\pi}} e^{-\frac{p^2(m_1-m_2)^2}{2\sigma^2}}.$$

Полученные формулы для I_1 и I_2 подставим в (33):

$$\mathbb{E}X_2 = \frac{1}{J_2} \left(\frac{\sigma}{\sqrt{2\pi}} \left(p e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} + (1-p) e^{-\frac{p^2(m_1-m_2)^2}{2\sigma^2}} \right) + m_1 J_{21} + m_2 J_{22} \right).$$

Теперь все те же рассуждения проведём для вычисления $\mathbb{E}X_1$:

$$\mathbb{E}X_1 = \frac{1}{J_1} \int_{-\infty}^{pm_1+(1-p)m_2} t \left(\frac{p}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} \right) dt + \frac{1}{J_1} \int_{-\infty}^{pm_1+(1-p)m_2} t \left(\frac{(1-p)}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} \right) dt = \frac{1}{J_1} (I_3 + I_4). \quad (34)$$

Интеграл I_3 разобьём на 2 части и в одной из частей произведём замену интеграла на J_{11} из (29):

$$\begin{aligned}
 I_3 &= p \int_{-\infty}^{pm_1+(1-p)m_2} \frac{t}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = p \int_{-\infty}^{pm_1+(1-p)m_2} \frac{t-m_1+m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = \\
 &= p \int_{-\infty}^{pm_1+(1-p)m_2} \frac{t-m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt + m_1 J_{11} = I_{31} + m_1 J_{11}.
 \end{aligned}$$

Выполним замену $s = \frac{(t-m_1)^2}{2\sigma^2}$ в интеграле I_{31} :

$$\begin{aligned} I_{31} &= p \int_{-\infty}^{pm_1+(1-p)m_2} \frac{t-m_1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = \left| \begin{array}{lll} s = \frac{(t-m_1)^2}{2\sigma^2} & \alpha = -\infty & \beta = pm_1 + (1-p)m_2 \\ ds = \frac{t-m_1}{\sigma^2} dt & a = +\infty & b = \frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2} \end{array} \right| = \\ &= p \int_{+\infty}^{\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} \frac{\sigma}{\sqrt{2\pi}} e^{-s} ds = -\frac{p\sigma}{\sqrt{2\pi}} e^{-s} \Big|_{+\infty}^{\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} = -\frac{p\sigma}{\sqrt{2\pi}} \left(e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} - e^{-\infty} \right) = \\ &= -\frac{p\sigma}{\sqrt{2\pi}} e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}}. \end{aligned}$$

Повторяя все те же рассуждения, получим формулу для I_4 :

$$I_4 = -\frac{(1-p)\sigma}{\sqrt{2\pi}} e^{-\frac{p^2(m_1-m_2)^2}{2\sigma^2}} + m_2 J_{12}.$$

Подставим I_3 и I_4 в (34):

$$\mathbb{E}X_1 = \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} \left(p e^{-\frac{(p-1)^2(m_1-m_2)^2}{2\sigma^2}} + (1-p) e^{-\frac{p^2(m_1-m_2)^2}{2\sigma^2}} \right) + m_1 J_{11} + m_2 J_{12} \right). \quad (35)$$

Нам известны $\mathbb{E}X$, $\mathbb{E}X_1$ и $\mathbb{E}X_2$. Для вычисления $SW(X)$ воспользуемся теоремой 1. Единственной неизвестной в формуле величиной является второй момент $\mathbb{E}X^2$. Его возьмём из (25).

Теперь покажем, что для описания SW -характеристики смеси двух гауссиан достаточно трёх переменных, то есть $SW(m_1, m_2, \sigma^2, p) = SW(d, \sigma^2, p)$, где $d = m_2 - m_1$. Это означает, что SW -характеристика зависит не от математических ожиданий в смеси, а только от расстояния между ними.

Для того, чтобы J_1 и J_2 зависели только от разности m_2 и m_1 , достаточно показать, что J_{11} , J_{12} , J_{21} и J_{22} зависят только от $m_2 - m_1$. Напомним, что $\mathbb{E}X = pm_1 + (1-p)m_2$. Рассмотрим интеграл J_{11} .

$$J_{11} = p \int_{-\infty}^{\mathbb{E}X} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_1)^2}{2\sigma^2}} dt = \left| \begin{array}{lll} s = t - m_1 & \alpha = -\infty & \beta = pm_1 + (1-p)m_2 \\ ds = dt & a = -\infty & b = (p-1)(m_1 - m_2) \end{array} \right| = p \int_{-\infty}^{-(p-1)d} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds.$$

В интеграле J_{12} выполним замену $t - m_2$:

$$J_{12} = \int_{-\infty}^{\mathbb{E}X} \frac{1-p}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m_2)^2}{2\sigma^2}} dt = \left| \begin{array}{lll} s = t - m_2 & \alpha = -\infty & \beta = pm_1 + (1-p)m_2 \\ ds = dt & a = -\infty & b = p(m_1 - m_2) \end{array} \right| = \int_{-\infty}^{-pd} \frac{1-p}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds.$$

То же самое сделаем для J_{21} и J_{22} . Получим

$$J_{21} = p \int_{-(p-1)d}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds, \quad J_{22} = \int_{-pd}^{+\infty} \frac{1-p}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds.$$

Выполним замену $m_2 = m_1 + d$ в $\mathbb{E}X$:

$$\mathbb{E}X = pm_1 + (1-p)m_2 = pm_1 + (1-p)(m_1 + d) = m_1 + (1-p)d.$$

Далее для сокращения размера формул будем использовать функцию

$$\xi(t) = te^{-\frac{(t-1)^2 d^2}{2\sigma^2}}. \quad (36)$$

Функция $\xi(t)$ будет фигурировать только с двумя аргументами:

$$\xi(p) = pe^{-\frac{(p-1)^2 d^2}{2\sigma^2}}, \quad (37)$$

$$\xi(1-p) = (1-p)e^{-\frac{p^2 d^2}{2\sigma^2}}. \quad (38)$$

Перепишем формулу (35) с учётом замены $m_2 = m_1 + d$ и (37), (38):

$$\begin{aligned} \mathbb{E}X_1 &= \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} \left(pe^{-\frac{(p-1)^2 d^2}{2\sigma^2}} + (1-p)e^{-\frac{p^2 d^2}{2\sigma^2}} \right) + m_1 J_{11} + m_2 J_{12} \right) = \\ &= \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + m_1 J_{11} + m_2 J_{12} \right) = \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + m_1 J_{11} + (m_1 + d) J_{12} \right) = \\ &= m_1 + \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{12} \right). \quad (39) \end{aligned}$$

То же самое проделаем с $\mathbb{E}X_2$:

$$\mathbb{E}X_2 = m_1 + \frac{1}{J_2} \left(\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{22} \right). \quad (40)$$

Отклонения $\mathbb{E}X_1$ и $\mathbb{E}X_2$ от m_1 обозначим θ_1 и θ_2 соответственно:

$$\theta_1 = \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{12} \right), \quad (41)$$

$$\theta_2 = \frac{1}{J_2} \left(\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{22} \right). \quad (42)$$

Рассмотрим выражение $\mathbb{E}X_1 - \mathbb{E}X_2$:

$$\begin{aligned} \mathbb{E}X_1 - \mathbb{E}X_2 &= m_1 + \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{12} \right) - m_1 - \frac{1}{J_2} \left(\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{22} \right) = \\ &= \frac{1}{J_1} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{12} \right) - \frac{1}{J_2} \left(\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{22} \right) = \\ &= \frac{1}{J_1 J_2} \left(-\frac{(J_1 + J_2)\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_2 J_{12} - d J_1 J_{22} \right) = \\ &= \frac{1}{J_1 J_2} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_2 J_{12} - d J_1 ((1-p) - J_{12}) \right) = \\ &= \frac{1}{J_1 J_2} \left(-\frac{\sigma}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + d J_{12} - d J_1 (1-p) \right) = \frac{1}{J_1 J_2} (J_1 \theta_1 - d J_1 (1-p)) = \frac{1}{J_2} (\theta_1 - d(1-p)). \quad (43) \end{aligned}$$

Из формул (41), (42) заметим, что

$$J_1 \theta_1 + J_2 \theta_2 = d(J_{12} + J_{22}) = d(1-p).$$

Напомним формулу SW-характеристики:

$$SW(X) = \frac{\mathbb{E}X^2 - J_1 \mathbb{E}^2 X_1 - J_2 \mathbb{E}^2 X_2}{(\mathbb{E}X_1 - \mathbb{E}X_2)^2}.$$

Формула (43) позволяет переписать знаменатель данной формулы через параметр d . Запишем числитель, воспользовавшись формулой (25). Ожидания $\mathbb{E}X_1$ и $\mathbb{E}X_2$ перепишем с использованием чисел θ_1, θ_2 из (41), (42):

$$\begin{aligned} \mathbb{E}X^2 - J_1\mathbb{E}^2X_1 - J_2\mathbb{E}^2X_2 &= \sigma^2 + pm_1^2 + (1-p)m_2^2 - J_1(m_1 + \theta_1)^2 - J_2(m_1 + \theta_2)^2 = \sigma^2 + pm_1^2 + (1-p)m_2^2 - \\ &- J_1(m_1^2 - 2m_1\theta_1 + \theta_1^2) - J_2(m_1^2 - 2m_1\theta_2 + \theta_2^2) = \sigma^2 + (p-1)m_1^2 + (1-p)(m_1 + d)^2 + J_1(2m_1\theta_1 - \theta_1^2) + \\ &+ J_2(2m_1\theta_2 - \theta_2^2) = \sigma^2 + (1-p)(-2m_1d + d^2) - J_1\theta_1^2 - J_2\theta_2^2 + 2m_1d(1-p) = \\ &= \sigma^2 + (1-p)d^2 - J_1\theta_1^2 - J_2\theta_2^2. \end{aligned} \quad (44)$$

Как можно видеть, и числитель (44), и знаменатель (43) SW-характеристики могут быть переписаны через параметр $d = m_2 - m_1$.

Оказывается, SW-характеристика смеси двух гауссиан может быть разрешена относительно всего лишь двух переменных. Вместо расстояния d между математическими ожиданиями гауссиан можно рассматривать величину $r = \frac{d}{\sigma}$.

$$SW(X) = SW(d, \sigma^2, p) = SW(r, p),$$

Замену переменной начнём выполнять с наиболее простых функций. Функция $\xi(t)$ из (36) переписывается следующим образом:

$$\xi(t) = te^{-\frac{(t-1)^2 r^2}{2}}. \quad (45)$$

Рассмотрим J_{11} .

$$J_{11} = p \int_{-\infty}^{-(p-1)d} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds = \left| \begin{array}{lll} l = \frac{s}{\sigma} & \alpha = -\infty & \beta = -(p-1)d \\ dl = \frac{ds}{\sigma} & a = -\infty & b = -\frac{(p-1)d}{\sigma} \end{array} \right| = p \int_{-\infty}^{-(p-1)r} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl.$$

Ту же замену произведём в интеграле J_{12} :

$$J_{12} = (1-p) \int_{-\infty}^{-pd} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} ds = \left| \begin{array}{lll} l = \frac{s}{\sigma} & \alpha = -\infty & \beta = -pd \\ dl = \frac{ds}{\sigma} & a = -\infty & b = -\frac{pd}{\sigma} \end{array} \right| = (1-p) \int_{-\infty}^{-pr} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl.$$

Опустим аналогичные преобразования для J_{21} и J_{22} .

$$J_{21} = p \int_{-(p-1)r}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl, \quad J_{22} = (1-p) \int_{-pr}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{l^2}{2}} dl.$$

Числитель SW-характеристики (44) перепишем следующим образом:

$$\sigma^2 + (1-p)d^2 - J_1\theta_1^2 - J_2\theta_2^2 = \sigma^2(1 + (1-p)r^2 - J_1\Theta_1^2 - J_2\Theta_2^2).$$

За Θ_1, Θ_2 обозначены величины:

$$\Theta_1 = \frac{\theta_1}{\sigma} = \frac{1}{J_1} \left(-\frac{1}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + rJ_{12} \right),$$

$$\Theta_2 = \frac{\theta_2}{\sigma} = \frac{1}{J_2} \left(\frac{1}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + rJ_{22} \right).$$

Преобразуем знаменатель SW -характеристики (43):

$$(\mathbb{E}X_1 - \mathbb{E}X_2)^2 = \frac{1}{J_2^2} (\theta_1 - d(1-p))^2 = \frac{\sigma^2}{J_2^2} (\Theta_1 - r(1-p))^2.$$

Число σ^2 присутствует и в знаменателе, и в числителе SW -характеристики. Следовательно, на него можно сократить. Итоговая формула имеет вид:

$$SW(X) = \frac{J_2^2(1 + (1-p)r^2 - J_1\Theta_1^2 - J_2\Theta_2^2)}{(\Theta_1 - r(1-p))^2}. \quad (46)$$

Формулу (46) можно ещё несколько упростить, чтобы она приобрела заявленный в теореме 4 вид. Перепишем Θ_2 , используя свойство (32):

$$\begin{aligned} \Theta_2 = \frac{\theta_2}{\sigma} &= \frac{1}{J_2} \left(\frac{1}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + rJ_{22} \right) = \frac{1}{J_2} \left(\frac{1}{\sqrt{2\pi}} (\xi(p) + \xi(1-p)) + r((1-p) - J_{12}) \right) = \\ &= \frac{1}{J_2} (-J_1\Theta_1 + r(1-p)). \end{aligned}$$

Числитель формулы (46) будет иметь вид

$$J_2^2(1 + (1-p)r^2 - J_1\Theta_1^2 - J_2\Theta_2^2) = J_2^2 \left(1 + (1-p)r^2 - J_1\Theta_1^2 - \frac{1}{J_2} (J_1^2\Theta_1^2 - 2J_1\Theta_1r(1-p) + r^2(1-p)^2) \right).$$

Воспользуемся следующим преобразованием

$$-J_1 - \frac{J_1^2}{J_2} = -\frac{J_1J_2 + J_1^2}{J_2} = -\frac{J_1(J_2 + J_1)}{J_2} = -\frac{J_1}{J_2},$$

Числитель SW -характеристики примет вид:

$$\begin{aligned} J_2^2 \left(1 + (1-p)r^2 - \frac{J_1}{J_2} \Theta_1^2 + 2\frac{J_1}{J_2} \Theta_1 r(1-p) - \frac{1}{J_2} r^2(1-p)^2 \right) &= \\ &= J_2^2 \left(1 + (1-p)r^2 - \frac{J_1}{J_2} (\Theta_1 - r(1-p))^2 + \frac{J_1}{J_2} r^2(1-p)^2 - \frac{1}{J_2} r^2(1-p)^2 \right). \end{aligned}$$

Теперь, пользуясь фактом, что $J_1 + J_2 = 1$, используем ещё одно полезное преобразование:

$$\frac{J_1}{J_2} - \frac{1}{J_2} = \frac{J_1 - (J_1 + J_2)}{J_2} = -1,$$

В результате числитель SW -характеристики перепишется:

$$\begin{aligned} J_2^2 \left(1 + (1-p)r^2 - \frac{J_1}{J_2} (\Theta_1 - r(1-p))^2 - r^2(1-p)^2 \right) &= \\ = J_2^2 \left(1 - \frac{J_1}{J_2} (\Theta_1 - r(1-p))^2 + r^2(-p^2 + 2p - 1 + 1 - p) \right) &= J_2^2 \left(1 - \frac{J_1}{J_2} (\Theta_1 - r(1-p))^2 + r^2(1-p)p \right). \quad (47) \end{aligned}$$

Теперь, подставив (47) в SW -характеристику, формулу удастся заметно сократить:

$$SW(X) = J_2^2 \frac{1 + r^2(1-p)p}{(\Theta_1 - r(1-p))^2} - J_1J_2. \quad (48)$$

□

Piecewise Constant Regimes of Fully Coupled Networks and Their Limit Integro-differential Systems

S. D. Glyzin¹, S. A. Kashchenko¹, D. S. Kosterin¹DOI: [10.18255/1818-1015-2025-2-206-224](https://doi.org/10.18255/1818-1015-2025-2-206-224)¹P.G. Demidov Yaroslavl State University, Yaroslavl, Russia

MSC2020: 45J05, 34K18, 39A11

Research article

Full text in Russian

Received April 23, 2025

Revised May 12, 2025

Accepted May 14, 2025

Fully connected networks of oscillators and their limit systems of integro-differential equations with periodic boundary conditions are considered. It is assumed that the connection is weak, i. e. the coefficient at the integral term is small. In the problem of stability of the zero equilibrium state, the simplest critical cases of loss of stability are distinguished. In these situations, quasi-normal forms are constructed, which are integro-differential equations for which several continuous families of piecewise constant two-step solutions are analytically determined, and their stability is studied. The existence of piecewise constant solutions with more than one discontinuity point is shown. A numerical experiment illustrating the analytical constructions is performed.

Keywords: integro-differential systems; piecewise constant solutions; stability; cluster synchronization

INFORMATION ABOUT THE AUTHORS

Glyzin, Sergey D.	ORCID iD: 0000-0002-6403-4061 . E-mail: glyzin@uniyar.ac.ru Head of the Department of Computer Networks, Dr. Sc., Professor
Kashchenko, Sergey A.	ORCID iD: 0000-0002-4846-6040 . E-mail: kasch@uniyar.ac.ru Dr. Sc., Professor
Kosterin, Dmitry S. (corresponding author)	ORCID iD: 0009-0003-6385-8063 . E-mail: kosterin.dim@mail.ru Senior Lecturer

Funding: This work was carried out within the framework of a development programme for the Regional Scientific and Educational Mathematical Center of the Yaroslavl State University with financial support from the Ministry of Science and Higher Education of the Russian Federation (Agreement on provision of subsidy from the federal budget No. 075-02-2025-1636).

For citation: S. D. Glyzin, S. A. Kashchenko, and D. S. Kosterin, "Piecewise constant modes of operation of fully coupled networks and their limit integro-differential systems", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 206–224, 2025. DOI: [10.18255/1818-1015-2025-2-206-224](https://doi.org/10.18255/1818-1015-2025-2-206-224).

Кусочно-постоянные режимы работы полносвязных сетей и их предельных интегро-дифференциальных систем

С. Д. Глызин¹, С. А. Кащенко¹, Д. С. Костерин¹

DOI: [10.18255/1818-1015-2025-2-206-224](https://doi.org/10.18255/1818-1015-2025-2-206-224)

¹Ярославский государственный университет им.П.Г. Демидова, Ярославль, Россия

УДК 517.937

Научная статья

Полный текст на русском языке

Получена 23 апреля 2025 г.

После доработки 12 мая 2025 г.

Принята к публикации 14 мая 2025 г.

Рассматриваются полносвязные сети осцилляторов и их предельные системы интегро-дифференциальных уравнений с периодическими краевыми условиями. Предполагается, что связь слабая, то есть мал коэффициент при интегральном члене. В задаче об устойчивости нулевого состояния равновесия выделяются простейшие критические случаи потери устойчивости. В этих ситуациях строятся квазинормальные формы, представляющие собой интегро-дифференциальные уравнения, для которых аналитически определяются несколько континуальных семейств кусочно-постоянных двухступенчатых решений. Исследуется устойчивость этих решений. Показано существование кусочно-постоянных решений, имеющих более одной точки разрыва. Выполнен численный эксперимент, иллюстрирующий аналитические построения.

Ключевые слова: интегро-дифференциальные системы; кусочно-постоянные решения; устойчивость; кластерная синхронизация

ИНФОРМАЦИЯ ОБ АВТОРАХ

Глызин, Сергей Дмитриевич	ORCID iD: 0000-0002-6403-4061 . E-mail: glyzin@uniyar.ac.ru Заведующий кафедрой компьютерных сетей, доктор физ.-мат. наук, профессор
Кащенко, Сергей Александрович	ORCID iD: 0000-0002-4846-6040 . E-mail: kasch@uniyar.ac.ru Доктор физ.-мат. наук, профессор
Костерин, Дмитрий Сергеевич (автор для корреспонденции)	ORCID iD: 0009-0003-6385-8063 . E-mail: kosterin.dim@mail.ru Старший преподаватель

Финансирование: Работа выполнена в рамках программы развития РНОМЦ ЯрГУ им. П. Г. Демидова при финансовой поддержке Министерства науки и высшего образования РФ (Соглашение о предоставлении субсидии из федерального бюджета № 075-02-2025-1636).

Для цитирования: S. D. Glyzin, S. A. Kashchenko, and D. S. Kosterin, "Piecewise constant modes of operation of fully coupled networks and their limit integro-differential systems", *Modeling and Analysis of Information Systems*, vol. 32, no. 2, pp. 206–224, 2025. DOI: [10.18255/1818-1015-2025-2-206-224](https://doi.org/10.18255/1818-1015-2025-2-206-224).

1. Постановка задачи

В работе рассматривается система интегро-дифференциальных уравнений

$$\frac{\partial u}{\partial t} = Au + F_2(u, u) + F_3(u, u, u) + D \frac{1}{2\pi} \int_0^{2\pi} f(s)u(t, x + s)ds \quad (1)$$

с периодическим краевым условием

$$u(t, x + 2\pi) \equiv u(t, x). \quad (2)$$

Здесь $u(t, x) \in \mathbb{R}^n$ при каждом $t > 0$, $x \in [0, 2\pi)$, A_0 , A_1 и D_0 , D_1 — $n \times n$ -матрицы. Вектор-функции $F_2(\cdot, \cdot)$, $F_3(\cdot, \cdot, \cdot)$ линейны по каждому аргументу.

Задача (1), (2) может быть получена в качестве предельной для полносвязной системы нелинейных осцилляторов, которая в общем случае (см. [1, 2]) может быть записана в следующей форме:

$$\dot{u}_j = Au_j + F_2(u_j, u_j) + F_3(u_j, u_j, u_j) + \frac{1}{N}D \sum_{k=1}^N c_{j-k}u_k, \quad (3)$$

где $j = 1, \dots, N+1$, $u_{N+1} \equiv u_1$, $u_j \in \mathbb{R}^n$, матрицы A_0 , A_1 и D_0 , D_1 — те же, что и в (1), c_j — весовые коэффициенты.

Значения $u_j(t)$ удобно ассоциировать со значениями функции двух переменных $u(t, x)$, где x_j — равномерно распределенные на некоторой окружности точки такие, что $x_j \in (-\infty, \infty)$ и выполнено условие периодичности $u(t, x_j + 1) \equiv u(t, x_j)$.

Предположим, что количество N элементов цепочки является достаточно большим: $N \gg 1$. В этом случае разность между значениями угловых переменных $x_{j+1} - x_j$ мала и равна

$$v = \frac{2\pi}{N} \ll 1.$$

Это дает основание переменную x считать непрерывной и от системы (3) перейти (см., [1–5]) к пространственно-распределенной системе уравнений (1) относительно $u(t, x)$ с периодическим краевым условием (2). Отметим, что весовые коэффициенты c_j предполагаются равными единице.

Система интегро-дифференциальных уравнений (1), (2), дополненная начальными условиями, является однозначно разрешимой (см. [6–8]). Следует отметить, что системы вида (1), (2) используются в качестве модельных для ряда задач нелинейной оптики [9, 10] и популяционной динамики [11–13].

Будем считать, что матрица A зависит от малого положительного параметра $0 < \varepsilon \ll 1$, причем

$$A = A_0 + \varepsilon A_1.$$

Заметим, что нулевое состояние равновесия получившейся краевой задачи является асимптотически устойчивым, если все собственные значения матриц

$$A + \frac{1}{2\pi}D \int_0^{2\pi} f(s) \exp(iks)ds \quad (k = 0, \pm 1, \pm 2, \dots) \quad (4)$$

имеют отрицательную вещественную часть.

В настоящей работе исследуется поведение в простейших критических случаях решений краевой задачи (1), (2) из некоторой достаточно малой и не зависящей от ε окрестности нулевого состояния равновесия.

В работе будет предполагаться, что связь, определяемая матрицей D , является слабой, т. е. эта матрица представима в виде

$$D = \varepsilon D_0.$$

Рассмотрим ситуацию, когда все весовые коэффициенты в системе (3) равны единице. В этом случае $f(x) \equiv 1$. Тогда при $k \neq 0$ все матрицы вида (4) равны A . При $k = 0$ имеем матрицу $A + D$, то есть матрицу $A_0 + \varepsilon A_1 + \varepsilon D_0$. Это означает, что устойчивость нулевого решения краевой задачи (1), (2) определяется в главном собственными числами матрицы A_0 .

Будем рассматривать следующие критические случаи:

1. Первый случай. Матрица A_0 имеет простое нулевое собственное значение, а все остальные ее собственные значения имеют отрицательные вещественные части.
2. Второй случай. Матрица A_0 имеет пару чисто мнимых собственных значений $\pm i\omega$ ($\omega > 0$), а все остальные ее собственные значения имеют отрицательные вещественные части.

Совокупность матриц (4) имеет в этих случаях бесконечно много собственных значений на мнимой оси. Тем самым реализуется критический случай в задаче об устойчивости решения $u \equiv 0$ бесконечной размерности. Ниже будет показано, что это приводит к новым динамическим эффектам.

Методика исследования рассматриваемой задачи схожа с исследованием систем обыкновенных дифференциальных уравнений вида

$$\dot{u} = (A_0 + \varepsilon A_1)u + F_2(u, u) + F_3(u, u, u). \quad (5)$$

Приведем некоторые известные результаты, связанные с устойчивостью нулевого решения этой системы. Если матрица A_0 имеет только собственные значения с отрицательной вещественной частью, то нулевое решение системы является асимптотически устойчивым. Если для матрицы A_0 выполнено условие 1 или 2, то реализуется критический случай размерности один или размерности два в задаче об устойчивости нулевого решения.

1. Первый случай. Учитывая, что матрица A_0 имеет простое нулевое собственное значение, предположим, что $A_0 a = 0$, $A_0^* b = 0$, где A_0^* — сопряженная матрица. Удобно, кроме того, считать, что $(a, b) = 1$.

Пусть выполнено неравенство

$$\delta = (F_2(a, a), b) \neq 0,$$

тогда в некоторой достаточно малой и не зависящей от ε окрестности нулевого состояния равновесия системы (5) существует локальное устойчивое одномерное инвариантное многообразие, на котором система (5) принимает с точностью до $O(\varepsilon)$ вид

$$\frac{d\xi}{d\tau} = \lambda \xi + \delta \xi^2, \quad (6)$$

где $\tau = \varepsilon t$, $\lambda = (A_1 a, b)$, $\delta = (F_2(a, a), b)$. Таким образом, при $\lambda, \delta \neq 0$ уравнение (6) имеет два состояния равновесия, одно из которых устойчиво, а второе — неустойчиво.

Решения (6) и (5) связаны асимптотическим равенством

$$u(t, \varepsilon) = \varepsilon a \xi(\tau) + \varepsilon^2 u_2 \xi^2(\tau) + \dots, \quad (7)$$

где u_2 — решение системы

$$\frac{\partial u_2}{\partial t} + \frac{d\xi}{d\tau} a = A_0 u_2 + \xi A_1 a + \xi^2 F_2(a, a).$$

Нормальная форма (6) выводится из условий разрешимости вышеуказанного уравнения в классе ограниченных функций.

Отметим, что при $\delta = 0$ последнее слагаемое в правой части (6) заменяется на $\delta_1 \xi^3$, где

$$\delta_1 = (F_3(a, a, a) + F_2(u_2, a) + F_2(a, u_2), b),$$

а разложение в аналогичный (7) ряд идет по степеням $\varepsilon^{1/2}$.

2. Второй случай. Здесь выполняется условие известной бифуркации Андронова–Хопфа. Предположим, что $A_0 a = i\omega a$, $A_0^* b = -i\omega b$, причем удобно считать, что $(a, b) = 1$, $(a, \bar{b}) = 0$.

Положим

$$\sigma = (F_3(a, a, \bar{a}) + F_3(a, \bar{a}, a) + F_3(a, a, \bar{a}) + F_2(u_{20}, a) + F_2(a, u_{20}) + F_2(u_{21}, \bar{a}) + F_2(\bar{a}, u_{21}), b),$$

пусть

$$\operatorname{Re} \sigma \neq 0,$$

тогда в достаточно малой и независимой от ε окрестности нулевого состояния равновесия (5) имеется двумерное устойчивое локальное инвариантное многообразие, на котором систему (5) с точностью до $O(\varepsilon)$ можно записать в виде

$$\frac{d\xi}{d\tau} = \lambda \xi + \sigma \xi |\xi|^2, \quad (8)$$

где $\tau = \varepsilon t$, $\lambda = (A_1 a, b)$.

Решения (8) и (5) связаны асимптотическим равенством

$$u = \varepsilon^{1/2} (a \xi \exp(i\omega t) + \bar{a} \bar{\xi} \exp(-i\omega t)) + \varepsilon (u_{20} |\xi|^2 + u_{21} \xi^2 \exp(2i\omega t) + \bar{u}_{21} \bar{\xi}^2 \exp(-2i\omega t)) + \dots,$$

где

$$u_{20} = -A_0^{-1} (F_2(a, \bar{a}) + F_2(\bar{a}, a)), \quad u_{21} = (2i\omega I - A_0)^{-1} F_2(a, a).$$

Отметим, что уравнение (8) интегрируется. Например, при условии $\operatorname{Re} \gamma > 0$ и $\operatorname{Re} \sigma < 0$ в (8) имеется орбитально устойчивый цикл $\xi = \xi_0 \exp(i\varphi_0 t)$, где

$$\xi_0(\tau) = ((\operatorname{Re} \lambda)(-\operatorname{Re} \sigma)^{-1})^{1/2} \exp(i\varphi_0 \tau), \quad \varphi_0 = \operatorname{Im} \gamma - \operatorname{Im} \sigma (\operatorname{Re} \lambda)(\operatorname{Re} \sigma)^{-1}.$$

Этому циклу соответствует устойчивый цикл в (5).

Уравнения (6) и (8) называют нормальными формами. Алгоритмы метода нормальных форм хорошо описаны в классических книгах [14–16] для конечномерного фазового пространства и в книгах [17–20] для бесконечномерного случая.

В настоящей работе для изучения локальной динамики краевой задачи (1), (2) в соответствующих критических случаях применен метод бесконечномерной нормализации, развитый в исследованиях [9, 10, 21, 22], построены аналоги нормальных форм, называемые квазинормальными формами (КНФ). В качестве основных результатов построены КНФ и приведены некоторые утверждения о решениях этих КНФ. В частности, показано, что они могут иметь семейства нетривиальных кусочно-непрерывных периодических решений.

2. Случай простого нулевого собственного значения. Уравнение с квадратичной нелинейностью

2.1. Построение квазинормальной формы

Здесь предполагаем, что выполнены условия случая 1. Способ построения КНФ довольно простой. Решения краевой задачи (1), (2) ищем в виде асимптотического ряда

$$u(t, x, \varepsilon) = \varepsilon \xi(\tau, x) a + \varepsilon^2 u_2 \xi^2(\tau, x) + \dots \quad (9)$$

Подставим (9) в (1). В получившемся формальном тождестве будем приравнивать коэффициенты при одинаковых степенях ε . На первом шаге, собирая коэффициенты при ε^1 , получаем верное равенство $A_0 a = 0$. Собирая коэффициенты при ε^2 , приходим к краевой задаче для определения неизвестной функции $\xi(\tau, x)$:

$$\frac{\partial \xi}{\partial \tau} = \lambda \xi + \delta \xi^2 + \gamma_0 M(\xi), \quad \xi(\tau, x + 2\pi) \equiv \xi(\tau, x). \quad (10)$$

Здесь коэффициенты λ и δ те же, что и в (6), а для γ_0 имеем равенство $\gamma_0 = (Da, b)$. $M(\xi)$ обозначает среднее функции ξ на отрезке $[0, 2\pi]$, т. е.

$$M(\xi) = \frac{1}{2\pi} \int_0^{2\pi} \xi(\tau, x) dx.$$

Отличие распределенной КНФ (10) от (6) состоит лишь в том, что в (10) присутствует линейное по ξ слагаемое $\gamma_0 M(\xi)$. Достаточно ограничиться случаем, когда $\lambda > 0$, $\delta < 0$. С помощью нормировок времени и функции ξ можно считать, что $\lambda = 1$, $\delta = -1$.

Таким образом, приходим к краевой задаче

$$\frac{\partial \xi}{\partial \tau} = \xi - \xi^2 + \gamma M(\xi), \quad \xi(\tau, x + 2\pi) \equiv \xi(\tau, x), \quad (11)$$

где γ — действительное число, пропорциональное γ_0 .

Далее рассмотрим примеры решений этой краевой задачи.

2.2. Устойчивость однородных состояний равновесия

В начале рассмотрим вопрос о существовании и устойчивости однородных состояний равновесия краевой задачи (11). Справедлив следующий результат.

Теорема 1. *Краевая задача (11) имеет однородные состояния равновесия $\xi_0 = 0$ и $\xi_1 = 1 + \gamma$. Состояние равновесия ξ_0 неустойчиво. Состояние равновесия ξ_1 является асимптотически устойчивым, если $\gamma > -1/2$ и неустойчивым, если $\gamma < -1/2$.*

Доказательство. Существование таких состояний равновесия проверяется непосредственной подстановкой их в уравнение (11). Неустойчивость состояния равновесия ξ_0 очевидна и легко получается применением теоремы Ляпунова об устойчивости по первому приближению.

Рассмотрим вопрос об устойчивости решения ξ_1 . Пусть

$$\xi(\tau, x) = 1 + \gamma + \eta(\tau, x).$$

Подставим $\xi(\tau, x)$ в уравнение (11), отбросив слагаемые более высокого порядка малости. Получим уравнение

$$\frac{\partial \eta}{\partial \tau} = (-1 - 2\gamma)\eta + \frac{\gamma}{2\pi} \int_0^{2\pi} \eta(\tau, x) dx.$$

Представим функцию $\eta(\tau, x)$ в виде суммы двух функций

$$\eta(\tau, x) = \eta_0(\tau, x) + \eta_1(\tau, x),$$

причем $\int_0^{2\pi} \eta_0(\tau, x) dx = 0$. Тогда получим систему из двух уравнений

$$\begin{aligned}\frac{\partial \eta_0}{\partial \tau} &= (-1 - 2\gamma)\eta_0, \\ \frac{\partial \eta_1}{\partial \tau} &= (-1 - 2\gamma)\eta_1 + \frac{\gamma}{2\pi} \int_0^{2\pi} \eta_1(\tau, x) dx.\end{aligned}$$

Проинтегрировав второе уравнение системы по x от 0 до 2π , получим линейную систему обыкновенных дифференциальных уравнений

$$\begin{aligned}\frac{\partial \eta_0}{\partial \tau} &= (-1 - 2\gamma)\eta_0, \\ \int_0^{2\pi} \frac{\partial \eta_1}{\partial \tau} dx &= (-1 - \gamma) \int_0^{2\pi} \eta_1(\tau, x) dx.\end{aligned}$$

Решения первого уравнения построенной системы стремятся к нулю при $\gamma > -1/2$, а решения второго уравнения — при $\gamma > -1$. Тем самым, решение $\xi_1 = 1 + \gamma$ краевой задачи (11) является асимптотически устойчивым при $\gamma > -1/2$ и неустойчивым при $\gamma < -1/2$. $\square$

2.3. Кусочно-постоянные решения

Рассмотрим вопрос о существовании и устойчивости неоднородных периодических по x состояний равновесия краевой задачи (11).

Пусть $\alpha_0 = 0 < \alpha_1 < \alpha_2 < \dots < \alpha_{k+1} = 2\pi$ и

$$\xi_*(x) = \{\rho_j \text{ при } x \in [\alpha_j, \alpha_{j+1}), j = 0, \dots, k\}$$

является кусочно-постоянным решением краевой задачи (11).

Определим понятие α -устойчивого кусочно-постоянного решения. Рассмотрим решения краевой задачи (11) с начальными условиями, «близкими» к $\xi_*(x)$. Для этого фиксируем достаточно малое значение $r > 0$ и рассмотрим множество начальных условий $\xi(\tau_0, x)$ вида

$$\{\xi(\tau_0, x) = \rho_j + \eta_j(x), \text{ при } x \in [\alpha_j, \alpha_{j+1}), j = 0, \dots, k, \max_x \sum_{j=1}^k |\eta_j(x)| \leq r\}.$$

Назовем решение $\xi_*(x)$ краевой задачи (11) α -устойчивым по Ляпунову, если для каждого $\varepsilon > 0$ найдётся такое $\delta > 0$, что из условия $\max_x \sum_{j=1}^k |\eta_j(x)| < \delta$ следует неравенство $\max_x |\xi(\tau, x) - \xi_*(x)| < \varepsilon$ при всех $\tau \geq \tau_0$.

Если при этом $\lim_{\tau \rightarrow \infty} \max_x |\xi(\tau, x) - \xi_*(x)| = 0$, то решение $\xi_*(x)$ будем называть асимптотически α -устойчивым.

Из указанных определений вытекает, что вопрос об α -устойчивости решения на отрезке $[0, 2\pi]$ можно условно разбить на изучение асимптотической устойчивости отдельно на каждом из промежутков $[\alpha_j, \alpha_{j+1})$. Кроме того, справедлива теорема А. М. Ляпунова об устойчивости по первому приближению.

Аналогично можно ввести понятие α -неустойчивости. Отметим, что из α -неустойчивости следует неустойчивость решения.

Обсудим теперь вопрос о существовании и устойчивости двухступенчатых решений краевой задачи. Справедлив следующий результат.

Теорема 2. Краевая задача (11) имеет однопараметрическое семейство кусочно-постоянных решений вида

$$\xi(\tau, x) = \begin{cases} \rho, & x \in [0, \alpha), \\ 1 - \rho, & x \in [\alpha, 2\pi), \end{cases} \quad (12)$$

где ρ — решение квадратного уравнения

$$\rho^2 - \left(1 + \frac{\gamma(\alpha - \pi)}{\pi}\right)\rho - \frac{\gamma(2\pi - \alpha)}{2\pi} = 0. \quad (13)$$

Решения этого семейства являются неустойчивыми при любых α и γ .

Доказательство. Наличие решений вида (12) проверяется непосредственной подстановкой функции $\xi(\tau, x)$ в уравнение (11).

Рассмотрим вопрос об устойчивости решений вида (12). Представим решение краевой задачи в виде

$$\xi(\tau, x) = \begin{cases} \rho + \eta_1(\tau, x), & x \in [0, \alpha), \\ 1 - \rho + \eta_2(\tau, x), & x \in [\alpha, 2\pi). \end{cases}$$

Подставив его в уравнение (11) и отбросив слагаемые более высокого порядка малости, получаем систему уравнений

$$\begin{aligned} \frac{\partial \eta_1}{\partial \tau} &= (1 - 2\rho)\eta_1 + \frac{\gamma}{2\pi} \left(\int_0^\alpha \eta_1(\tau, x) dx + \int_\alpha^{2\pi} \eta_2(\tau, x) dx \right), \\ \frac{\partial \eta_2}{\partial \tau} &= (-1 + 2\rho)\eta_2 + \frac{\gamma}{2\pi} \left(\int_0^\alpha \eta_1(\tau, x) dx + \int_\alpha^{2\pi} \eta_2(\tau, x) dx \right). \end{aligned}$$

Представим функции $\eta_1(\tau, x)$ и $\eta_2(\tau, x)$ в виде суммы двух функций

$$\eta_j(\tau, x) = \eta_{j0}(\tau, x) + \eta_{j1}(\tau, x), \quad j = 1, 2,$$

где $\int_0^\alpha \eta_{10}(\tau, x) dx = 0, \int_\alpha^{2\pi} \eta_{20}(\tau, x) dx = 0.$

Тогда получим систему четырех уравнений

$$\begin{aligned} \frac{\partial \eta_{10}}{\partial \tau} &= (1 - 2\rho)\eta_{10}, \quad \frac{\partial \eta_{20}}{\partial \tau} = (-1 + 2\rho)\eta_{20}, \\ \frac{\partial \eta_{11}}{\partial \tau} &= (1 - 2\rho)\eta_{11} + \frac{\gamma}{2\pi} \left(\int_0^\alpha \eta_{11}(\tau, x) dx + \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx \right), \\ \frac{\partial \eta_{21}}{\partial \tau} &= (-1 + 2\rho)\eta_{21} + \frac{\gamma}{2\pi} \left(\int_0^\alpha \eta_{11}(\tau, x) dx + \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx \right). \end{aligned}$$

Проинтегрируем третье уравнение по x от 0 до α , а четвертое уравнение — от α до 2π . Получим линейную систему обыкновенных дифференциальных уравнений

$$\begin{aligned}\frac{\partial \eta_{10}}{\partial \tau} &= (1 - 2\rho)\eta_{10}, \quad \frac{\partial \eta_{20}}{\partial \tau} = (-1 + 2\rho)\eta_{20}, \\ \int_0^\alpha \frac{\partial \eta_{11}}{\partial \tau} dx &= \left(1 - 2\rho + \frac{\gamma\alpha}{2\pi}\right) \int_0^\alpha \eta_{11}(\tau, x) dx + \frac{\gamma\alpha}{2\pi} \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx, \\ \int_0^\alpha \frac{\partial \eta_{21}}{\partial \tau} dx &= \frac{\gamma(2\pi - \alpha)}{2\pi} \int_0^\alpha \eta_{11}(\tau, x) dx + \left(\frac{-1 + 2\rho + \gamma(2\pi - \alpha)}{2\pi}\right) \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx.\end{aligned}$$

Если $1 - 2\rho < 0$, то $-1 + 2\rho > 0$ и наоборот. Значит, решения этой системы не стремятся к нулю. Таким образом, решения вида (12) являются неустойчивыми. $\square$

3. Случай простого нулевого собственного значения. Уравнение с кубической нелинейностью

3.1. Построение квазинормальной формы

Более интересная ситуация может возникать в случае дополнительного вырождения, когда в (10) $\delta = 0$ и $\delta_1 \neq 0$. Рассмотрим ее при условиях $\lambda > 0$ и $\delta_1 < 0$. В этом случае роль ряда (9) играет ряд

$$u(t, x, \varepsilon) = \varepsilon^{1/2} \xi(\tau, x) + \varepsilon u_2 \xi^2(\tau, x) + \varepsilon^{3/2} u_3 \xi^3(\tau, x) + \dots \quad (14)$$

Подставляя его в (1) и совершая стандартные действия, приходим к КНФ

$$\frac{\partial \xi}{\partial \tau} = \lambda \xi + \delta_1 \xi^3 + \gamma_0 M(\xi), \quad \xi(\tau, x + 2\pi) \equiv \xi(\tau, x). \quad (15)$$

Можно считать, что $\lambda = 1$, $\delta_1 = -1$. Таким образом, имеем краевую задачу

$$\frac{\partial \xi}{\partial \tau} = \xi - \xi^3 + \gamma M(\xi), \quad \xi(\tau, x + 2\pi) \equiv \xi(\tau, x), \quad (16)$$

где γ — действительное число, пропорциональное γ_0 .

3.2. Устойчивость однородных состояний равновесия

Как и для краевой задачи (11) рассмотрим свойства устойчивости однородных состояний равновесия.

Теорема 3. Краевая задача (16) имеет два ненулевых однородных состояния равновесия $\xi_1 = \sqrt{1 + \gamma}$ и $\xi_2 = -\sqrt{1 + \gamma}$, которые являются асимптотически устойчивыми при $\gamma > -\frac{2}{3}$ и неустойчивыми при $\gamma < -\frac{2}{3}$.

Доказательство. Существование таких решений проверяется непосредственной подстановкой ξ_1 и ξ_2 в уравнение (16).

Рассмотрим вопрос об устойчивости этих решений. Пусть

$$\xi(\tau, x) = \pm \sqrt{1 + \gamma} + \eta(\tau, x).$$

Аналогично предыдущему случаю построим линейную систему дифференциальных уравнений

$$\begin{aligned}\frac{\partial \eta_0}{\partial \tau} &= (-2 - 3\gamma)\eta_0, \\ \int_0^{2\pi} \frac{\partial \eta_1}{\partial \tau} dx &= (-2 - 2\gamma) \int_0^{2\pi} \eta_1(\tau, x) dx.\end{aligned}$$

Решения второго уравнения стремятся к нулю при $\gamma > -1$, решения первого — при $\gamma > -\frac{2}{3}$. Таким образом, состояния равновесия ξ_1 и ξ_2 являются асимптотически устойчивыми при $\gamma > -\frac{2}{3}$. $\square$

3.3. Кусочно-постоянные решения

Наконец, рассмотрим вопрос о двухступенчатых кусочно-постоянных решениях.

Теорема 4. Краевая задача (16) имеет семейство кусочно-постоянных решений вида

$$\xi(\tau, x) = \begin{cases} \rho_1, & x \in [0, \alpha), \\ \rho_2, & x \in [\alpha, 2\pi), \end{cases} \quad (17)$$

где ρ_1, ρ_2, α — решение алгебраической системы

$$\rho_j - \rho_j^3 + \frac{\gamma}{2\pi} [\rho_1 \alpha + (2\pi - \alpha) \rho_2] = 0, \quad j = 1, 2. \quad (18)$$

Решение такого вида является устойчивым, если

$$\rho_j^2 > \frac{1}{3}, \quad j = 1, 2, \quad (19)$$

$$2 - 3(\rho_1^2 + \rho_2^2) + \gamma < 0, \quad (20)$$

$$(1 - 3\rho_1^2)(1 - 3\rho_2^2) + \gamma - 3 \left[\rho_1^2 \gamma + \frac{\gamma \alpha}{2\pi} (\rho_2^2 - \rho_1^2) \right] > 0. \quad (21)$$

Доказательство. Система уравнений (18) получается при подстановке функции (17) в уравнение (16).

Рассмотрим вопрос об устойчивости решений вида (17). Пусть

$$\xi(\tau, x) = \begin{cases} \rho_1 + \eta_1(\tau, x), & x \in [0, \alpha), \\ \rho_2 + \eta_2(\tau, x), & x \in [\alpha, 2\pi). \end{cases}$$

Аналогично предыдущему случаю построим линейную систему дифференциальных уравнений.

$$\begin{aligned}\frac{\partial \eta_{10}}{\partial \tau} &= (1 - 3\rho_1^2)\eta_{10}, \quad \frac{\partial \eta_{20}}{\partial \tau} = (1 - 3\rho_2^2)\eta_{20}, \\ \int_0^\alpha \frac{\partial \eta_{11}}{\partial \tau} dx &= \left(1 - 3\rho_1^2 + \frac{\gamma \alpha}{2\pi}\right) \int_0^\alpha \eta_{11}(\tau, x) dx + \frac{\gamma \alpha}{2\pi} \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx, \\ \int_0^\alpha \frac{\partial \eta_{21}}{\partial \tau} dx &= \frac{\gamma(2\pi - \alpha)}{2\pi} \int_0^\alpha \eta_{11}(\tau, x) dx + \left(1 - 3\rho_2^2 + \frac{\gamma(2\pi - \alpha)}{2\pi}\right) \int_\alpha^{2\pi} \eta_{21}(\tau, x) dx.\end{aligned}$$

Решения первых двух уравнений стремятся к нулю, если выполнено условие (19). Третье и четвертое уравнение образуют линейную систему двух дифференциальных уравнений с матрицей

$$\begin{pmatrix} 1 - 3\rho_1^2 + \frac{\gamma\alpha}{2\pi} & \frac{\gamma\alpha}{2\pi} \\ \frac{\gamma(2\pi-\alpha)}{2\pi} & 1 - 3\rho_2^2 + \frac{\gamma(2\pi-\alpha)}{2\pi} \end{pmatrix}.$$

Решение этой системы стремится к нулю, если оба собственных значения указанной матрицы имеют отрицательную вещественную часть, что, согласно критерию Рауса–Гурвица, будет выполнено в случае, когда след матрицы отрицателен, а определитель положителен. Вычисляя след и определитель, получаем условия (20) и (21) соответственно. $\square$

3.4. Алгоритм построения кусочно-постоянных решений

Для построения решений вида (17) введем параметр c такой, что уравнение

$$\rho - \rho^3 = c$$

имеет хотя бы два вещественных корня ρ_1 и ρ_2 . Тогда из системы (18) получаем

$$\frac{\gamma}{2\pi} [\rho_1\alpha + (2\pi - \alpha)\rho_2] = -c,$$

из чего получаем формулу для вычисления α как функции от c

$$\alpha = \frac{2\pi}{\rho_2 - \rho_1} \left[\frac{c}{\gamma} + \rho_2 \right]. \quad (22)$$

На рисунке 1 показаны графики зависимости α от c . На более жирной кривой находятся те значения α , при которых решение вида (17) является асимптотически устойчивым.

Отметим, что зависимость α от c является симметричной относительно начала координат. Чтобы убедиться в этом, достаточно поменять знаки ρ_1 и ρ_2 на противоположные.

Вычисления показали, что увеличение параметра γ приводит к сужению интервала значений параметра c , на котором решения вида (17) устойчивы и расширению интервалов неустойчивости (см. рис. 1, на котором слева изображена кривая $\alpha(c)$ при $\gamma = -0.5$, а справа — при $\gamma = 1.5$).

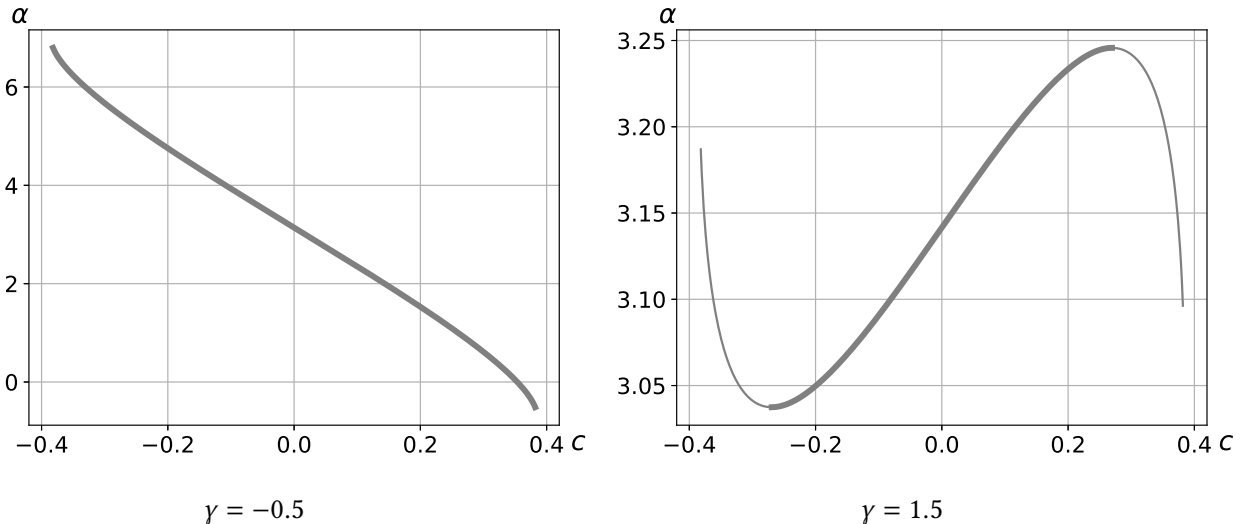


Fig. 1. The dependence of α on c when $\gamma = -0.5$ and $\gamma = 1.5$

Рис. 1. Зависимость α от c при $\gamma = -0.5$ и $\gamma = 1.5$

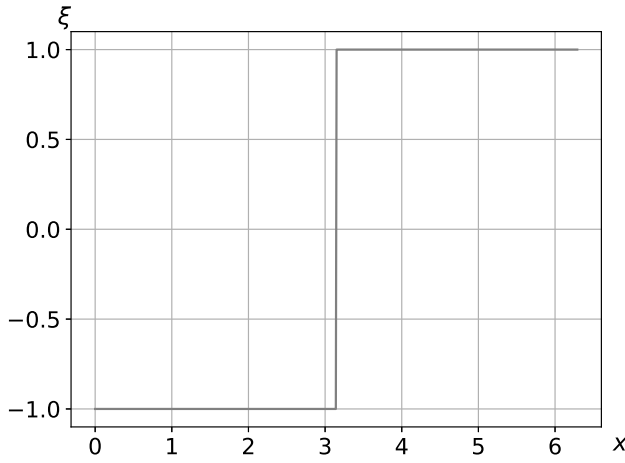

 $\gamma = -0.5$

Fig. 2. Piecewise constant solution of the boundary value problem (16) for $\gamma = -0.5$ and $\gamma = 1.5$

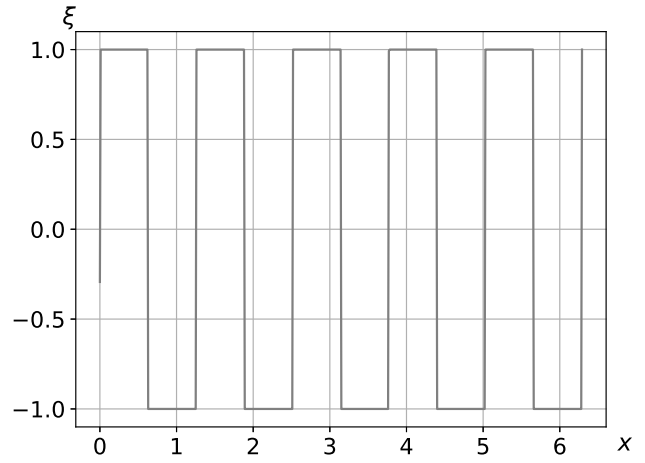

 $\gamma = 1.5$

Рис. 2. Кусочно-постоянное решение краевой задачи (16) при $\gamma = -0.5$ и $\gamma = 1.5$

На рисунке 2 представлены полученные численно графики распределения по пространственной переменной устойчивых кусочно-постоянных решений, к которым сходятся решения краевой задачи (16). Графики построены при тех же значениях параметра γ , что и выше. На левом графике двухступенчатое кусочно-постоянное решение устойчиво, а на правом — неустойчиво и решение задачи (16) сходится к кусочно-постоянному режиму с большим числом ступенек.

Для построенных в разделах 2, 3 устойчивых кусочно-постоянных решений выполнено следующее утверждение о соответствии (см. [21, 22]).

Теорема 5. Пусть краевая задача (10) (соответственно (15)) имеет асимптотически устойчивое кусочно-постоянное решение, тогда при всех достаточно малых $\varepsilon > 0$ краевая задача (1), (2) имеет устойчивое решение с асимптотикой (9) (соответственно (14)).

4. Случай пары чисто мнимых собственных чисел

В этом разделе предполагаем, что выполнены условия случая 2, когда матрица A_0 имеет пару чисто мнимых собственных значений. Решение краевой задачи (1), (2) ищем в виде следующего асимптотического ряда:

$$\begin{aligned} u(t, x, \varepsilon) = & \varepsilon^{1/2} (a\xi(\tau, x) \exp(i\omega t) + \bar{a}\bar{\xi}(\tau, x) \exp(-i\omega t)) + \varepsilon (u_{20}|\xi(\tau, x)|^2 + \\ & + u_{21}\xi^2(\tau, x) \exp(2i\omega t) + \bar{u}_{21}\bar{\xi}^2(\tau, x) \exp(-2i\omega t)) + \dots \end{aligned} \quad (23)$$

Подставляя (23) в (1) и приравнивая коэффициенты при степенях $\sqrt{\varepsilon}$, на третьем шаге алгоритма получаем КНФ вида

$$\frac{\partial \xi}{\partial \tau} = \lambda \xi + \sigma \xi |\xi|^2 + \gamma M(\xi), \quad \xi(\tau, x + 2\pi) \equiv \xi(\tau, x), \quad (24)$$

где λ и σ те же, что и в (8) и, как и ранее, $\gamma = (Da, b)$.

Для КНФ выполнено следующее основное утверждение.

Теорема 6. Пусть краевая задача (24) имеет асимптотически устойчивое состояние равновесия или орбитально асимптотически устойчивый цикл, тогда при всех достаточно малых $\varepsilon > 0$ ему соответствует устойчивое решение краевой задачи (1), (2) с асимптотикой (23).

На доказательстве данного утверждения не останавливаемся, отсылая читателей к исследованиям [21, 22], в которых доказываются теоремы о соответствии, аналогичные данной.

Предположим, что $\operatorname{Re} \lambda > 0$, $\operatorname{Re} \sigma < 0$. С помощью нормировок можно сделать так, что $\operatorname{Re} \lambda = 1$, $\operatorname{Re} \sigma = -1$, то есть $\lambda = 1 + i\lambda_0$, $\sigma = -1 + i\sigma_0$, коэффициент при $M(\xi)$ после замен обозначим снова $\gamma = \gamma_0 + i\gamma_1$.

4.1. Однородные по пространственной переменной решения

Нетрудно убедиться, что краевая задача (24) имеет пространственно однородный цикл вида

$$\xi(\tau, x) = \sqrt{1 + \gamma_0} \exp(i((\lambda_0 + \sigma_0(1 + \gamma_0) + \gamma_1)\tau + c)), \quad (25)$$

где c — произвольная действительная константа.

Рассмотрим вопрос об устойчивости этого решения. Положим

$$\xi(\tau, x) = \sqrt{1 + \gamma_0} \exp(i((\lambda_0 + \sigma_0(1 + \gamma_0) + \gamma_1)\tau + c))(1 + \eta(\tau, x)),$$

и подставим это представление в уравнение (24). Для переменной $\eta(\tau, x)$ выделим линейную часть

$$\frac{\partial \eta}{\partial \tau} = (-1 - 2\gamma_0 + i\sigma_0(1 + \gamma_0) - i\gamma_1)\eta + (-1 + i\sigma_0)(1 + \gamma_0)\bar{\eta} + \frac{\gamma_0 + i\gamma_1}{2\pi} \int_0^{2\pi} \eta(\tau, x) dx, \quad (26)$$

вместе с которой будем рассматривать комплексно сопряженное к нему уравнение для $\bar{\eta}(\tau, x)$. Представим теперь $\eta(\tau, x)$ в виде суммы двух функций

$$\eta(\tau, x) = \eta_0(\tau, x) + \eta_1(\tau, x)$$

так, что $\int_0^{2\pi} \eta_0(\tau, x) dx = 0$. Тогда уравнение (26) превращается в систему из двух уравнений

$$\begin{aligned} \frac{\partial \eta_0}{\partial \tau} &= (-1 - 2\gamma_0 + i\sigma_0(1 + \gamma_0) - i\gamma_1)\eta_0 + (-1 + i\sigma_0)(1 + \gamma_0)\bar{\eta}_0, \\ \frac{\partial \eta_1}{\partial \tau} &= (-1 - 2\gamma_0 + i\sigma_0(1 + \gamma_0) - i\gamma_1)\eta_1 + (-1 + i\sigma_0)(1 + \gamma_0)\bar{\eta}_1 + \frac{\gamma_0 + i\gamma_1}{2\pi} \int_0^{2\pi} \eta_1(\tau, x) dx, \end{aligned}$$

которую также будем рассматривать совместно с системой сопряженных уравнений для $\bar{\eta}_0$, $\bar{\eta}_1$. Интегрируя уравнения для η_1 и $\bar{\eta}_1$, получаем итоговую линейную систему

$$\begin{aligned} \frac{\partial \eta_0}{\partial \tau} &= (-1 - 2\gamma_0 + i\sigma_0(1 + \gamma_0) - i\gamma_1)\eta_0 + (-1 + i\sigma_0)(1 + \gamma_0)\bar{\eta}_0, \\ \frac{\partial \bar{\eta}_0}{\partial \tau} &= (-1 - i\sigma_0)(1 + \gamma_0)\eta_0 + (-1 - 2\gamma_0 - i\sigma_0(1 + \gamma_0) + i\gamma_1)\bar{\eta}_0, \\ \int_0^{2\pi} \frac{\partial \eta_1}{\partial \tau} dx &= (-1 - \gamma_0 + i\sigma_0(1 + \gamma_0)) \int_0^{2\pi} \eta_1 dx + (-1 + i\sigma_0)(1 + \gamma_0) \int_0^{2\pi} \bar{\eta}_1 dx, \\ \int_0^{2\pi} \frac{\partial \bar{\eta}_1}{\partial \tau} dx &= (-1 - i\sigma_0)(1 + \gamma_0) \int_0^{2\pi} \eta_1 dx + (-1 - \gamma_0 - i\sigma_0(1 + \gamma_0)) \int_0^{2\pi} \bar{\eta}_1 dx. \end{aligned}$$

Она распадается на две независимые двумерные части. Решения первых двух уравнений стремятся к нулю, если собственные числа матрицы

$$\begin{pmatrix} -1 - 2\gamma_0 + i(\sigma_0(1 + \gamma_0) - \gamma_1) & (-1 + i\sigma_0)(1 + \gamma_0) \\ (-1 - i\sigma_0)(1 + \gamma_0) & -1 - 2\gamma_0 - i(\sigma_0(1 + \gamma_0) - \gamma_1) \end{pmatrix}$$

имеют отрицательные вещественные части. Характеристическое уравнение этой матрицы имеет вид

$$\mu^2 + (2 + 4\gamma_0)\mu + \gamma_0(2 + 3\gamma_0) + \gamma_1(\gamma_1 - 2\sigma_0(1 + \gamma_0)) = 0,$$

и его решения имеют отрицательную вещественную часть, если коэффициенты при μ и μ^0 положительны. Таким образом, условием стремления к нулю решений первой пары уравнений линейаризованной системы является выполнение неравенств

$$\gamma_0 > -\frac{1}{2}, \quad \sigma_0\gamma_1 < \frac{\gamma_0(2 + 3\gamma_0) + \gamma_1^2}{2(1 + \gamma_0)}. \quad (27)$$

В свою очередь, поведение решений третьего и четвертого уравнений определяются собственными числами матрицы

$$\begin{pmatrix} -1 - \gamma_0 + i\sigma_0(1 + \gamma_0) & (-1 + i\sigma_0)(1 + \gamma_0) \\ (-1 - i\sigma_0)(1 + \gamma_0) & -1 - \gamma_0 - i\sigma_0(1 + \gamma_0) \end{pmatrix},$$

характеристическое уравнение которой имеет вид

$$\mu^2 + 2(1 + \gamma_0)\mu = 0.$$

Это уравнение имеет одно нулевое и одно отрицательное вещественное решения.

Заметим, что наличие нулевого собственного числа не является критическим в задаче об устойчивости, поскольку оно соответствует единичному мультипликатору матрицы монодромии линейной системы в окрестности периодического решения. Отрицательные собственные числа, в свою очередь, соответствуют мультипликаторам, модули которых меньше единицы. Приведенные рассуждения позволяют сформулировать теорему об устойчивости пространственно однородного цикла (25).

Теорема 7. Пусть $\gamma_0 > -1$, тогда краевая задача (24) имеет пространственно однородный цикл (25), который орбитально асимптотически устойчив при выполнении неравенств (27) и неустойчив при нарушении любого из них.

Отметим, что при вещественном γ , то есть при $\gamma_1 = 0$ условие устойчивости однородного цикла существенно упрощается и неравенства (27) превращаются в условие $\gamma_0 > 0$.

В следующем разделе рассмотрим кусочно-постоянные по пространственной переменной решения, которые могут оказаться устойчивыми в случае, если цикл (25) неустойчив или не существует.

4.2. Кусочно-постоянные по пространственной переменной решения

Будем искать кусочно-постоянное решение краевой задачи (24) в виде

$$\xi(t, x) = \rho(x)e^{i\omega\tau}, \quad (28)$$

где

$$\rho(x) = \begin{cases} \rho_1 e^{i\varphi_1}, & x \in [0, \alpha), \\ \rho_2 e^{i\varphi_2}, & x \in [\alpha, 2\pi). \end{cases}$$

С помощью нормировок и переобозначений можно сделать так, что $\varphi_1 = 0$. Будем считать в этом случае $\varphi_2 = \varphi$.

Подставляя $\xi(\tau, x)$ в уравнение (24) и выделяя вещественную и мнимую части, получаем систему алгебраических уравнений

$$\begin{aligned} \rho_1 - \rho_1^3 &= -\frac{\gamma_0}{2\pi} [\alpha \rho_1 + (2\pi - \alpha) \rho_2 \cos \varphi] + \frac{\gamma_1}{2\pi} (2\pi - \alpha) \rho_2 \sin \varphi, \\ -\omega \rho_1 + \lambda_0 \rho_1 + \sigma_0 \rho_1^3 &= -\frac{\gamma_0}{2\pi} (2\pi - \alpha) \rho_2 \sin \varphi - \frac{\gamma_1}{2\pi} [\alpha \rho_1 + (2\pi - \alpha) \rho_2 \cos \varphi], \\ \rho_2 - \rho_2^3 &= -\frac{\gamma_0}{2\pi} [\alpha \rho_1 \cos \varphi + (2\pi - \alpha) \rho_2] - \frac{\gamma_1}{2\pi} \alpha \rho_1 \sin \varphi, \\ -\omega \rho_2 + \lambda_0 \rho_2 + \sigma_0 \rho_2^3 &= \frac{\gamma_0}{2\pi} \alpha \rho_1 \sin \varphi - \frac{\gamma_1}{2\pi} [\alpha \rho_1 \cos \varphi + (2\pi - \alpha) \rho_2]. \end{aligned} \quad (29)$$

Выразим ρ_1^3 и ρ_2^3 из первого и третьего уравнения системы (29) и подставим во второе и четвертое уравнения. Получим однородную систему линейных уравнений

$$\begin{aligned} \left(-\omega + \lambda_0 + \sigma_0 + \frac{\alpha}{2\pi} (\sigma_0 \gamma_0 + \gamma_1) \right) \rho_1 + \frac{2\pi - \alpha}{2\pi} (\gamma_0 (\sigma_0 \cos \varphi + \sin \varphi) + \gamma_1 (\cos \varphi - \sigma_0 \sin \varphi)) \rho_2 &= 0, \\ \frac{\alpha}{2\pi} (\gamma_0 (\sigma_0 \cos \varphi - \sin \varphi) + \gamma_1 (\sigma_0 \sin \varphi + \cos \varphi)) \rho_1 + \left(-\omega + \lambda_0 + \sigma_0 + \frac{2\pi - \alpha}{2\pi} (\sigma_0 \gamma_0 + \gamma_1) \right) \rho_2 &= 0 \end{aligned}$$

относительно ρ_1 и ρ_2 . Эта система имеет ненулевое решение только, если ω является собственным числом матрицы

$$\begin{pmatrix} \lambda_0 + \sigma_0 + \frac{\alpha}{2\pi} (\sigma_0 \gamma_0 + \gamma_1) & \frac{2\pi - \alpha}{2\pi} (\gamma_0 (\sigma_0 \cos \varphi + \sin \varphi) + \gamma_1 (\cos \varphi - \sigma_0 \sin \varphi)) \\ \frac{\alpha}{2\pi} (\gamma_0 (\sigma_0 \cos \varphi - \sin \varphi) + \gamma_1 (\sigma_0 \sin \varphi + \cos \varphi)) & \lambda_0 + \sigma_0 + \frac{2\pi - \alpha}{2\pi} (\sigma_0 \gamma_0 + \gamma_1) \end{pmatrix}.$$

В этом случае рассматриваемая система имеет семейство решений вида $\rho_1 = c \rho_2$, где

$$c = -\frac{2\pi - \alpha}{2\pi} \frac{\gamma_0 (\sigma_0 \cos \varphi + \sin \varphi) + \gamma_1 (\cos \varphi - \sigma_0 \sin \varphi)}{-\omega + \lambda_0 + \sigma_0 + \sigma_0 + \frac{\alpha}{2\pi} (\sigma_0 \gamma_0 + \gamma_1)}.$$

Подставляя $\rho_1 = c \rho_2$ в третье уравнение системы (29), определим ρ_2 :

$$\rho_2 = \pm \sqrt{\frac{\alpha}{2\pi} (\gamma_0 \cos \varphi + \gamma_1 \sin \varphi) c + 1 + \frac{\gamma_0 (2\pi - \alpha)}{2\pi}}.$$

Наконец, подставив все полученные выражения в первое уравнение системы (29), можно при заданном значении φ определить величину α , а вслед за ней ρ_1 , ρ_2 и ω .

На рисунке 3 представлены значения α , ρ_1 , ρ_2 , ω при изменении φ от 0 до 2π . Параметры задачи зафиксированы следующим образом: $\lambda_0 = 1$, $\sigma_0 = 1$; величина γ выбрана вещественной, то есть $\gamma_1 = 0$, а вещественная часть этого параметра взята такой, что однородный цикл задачи неустойчив ($\gamma_0 = -1$). Следует отметить, что изменение этих параметров в довольно широких пределах показало близость общего вида графиков 3. Анализ представленных зависимостей показывает, что имеются промежутки значений φ , при которых решение системы (29) не существует.

Вместе с тем, исследование устойчивости решений вида (28) на промежутках изменения φ , на которых такие решения существуют, оказалось технически достаточно сложной задачей, поэтому в рамках данной работы остановимся только на результатах численного эксперимента.

На рисунках 4, 5 показаны графики зависимости вещественных частей решений $\xi(\tau, x)$ от x в начальный момент времени ($\text{Re}(\xi(0, x))$) и построенные численно в момент времени $\tau = 100$ ($\text{Re}(\xi(100, x))$). При этом в качестве начальных условий $\xi(0, x)$ для краевой задачи (24) брались

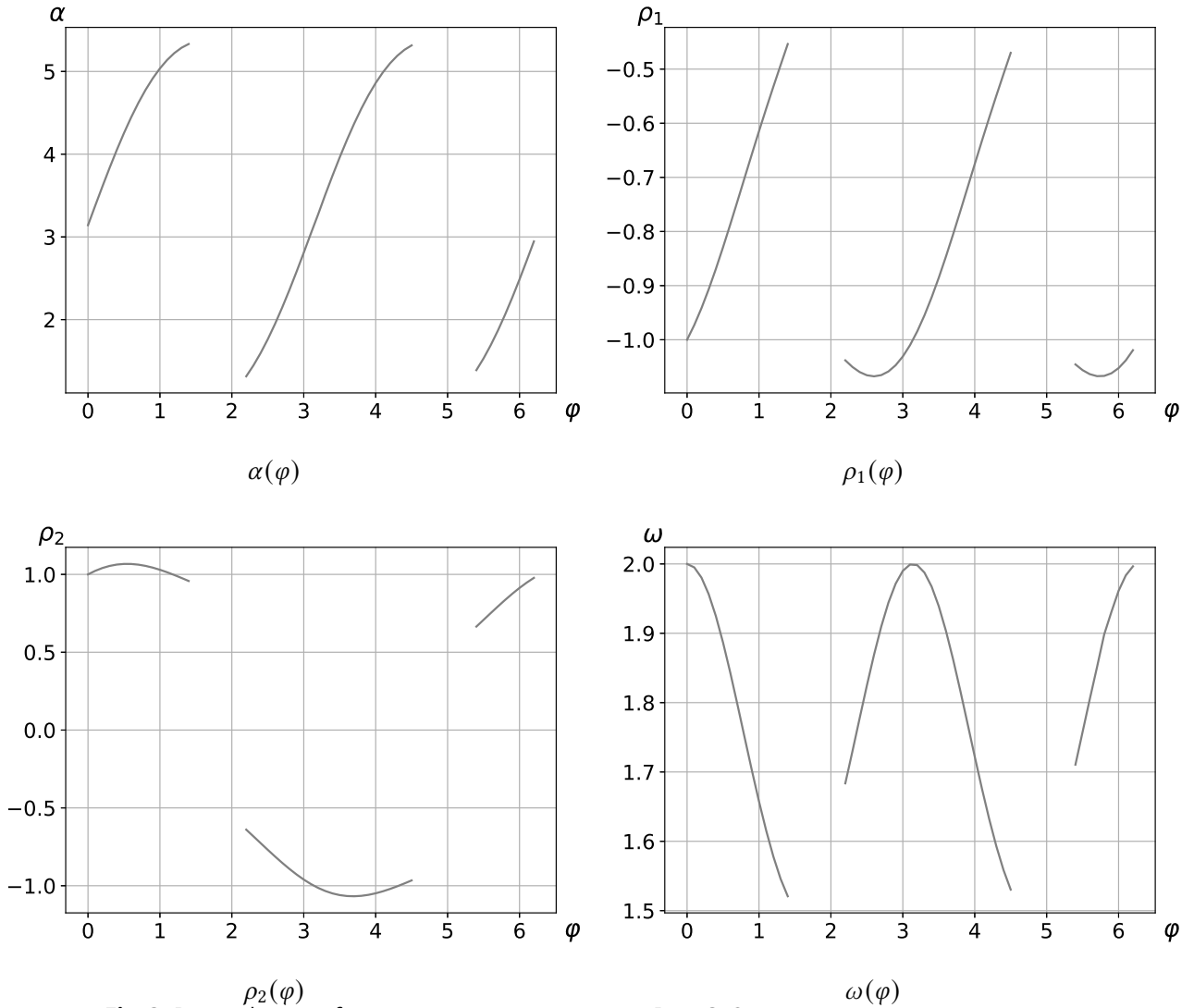

 Fig. 3. Dependences of α , ρ_1 , ρ_2 , ω on φ

 Рис. 3. Зависимости величин α , ρ_1 , ρ_2 , ω от φ

функции (28) с параметрами, вычисленными по изложенной схеме и с добавкой достаточно малого возмущения.

Начальной функцией решения, изображенного на рисунке 4, является функция (28) при $\varphi = 0$. В этом случае $\rho_1 = -1$, $\rho_2 = 1$, $\omega = 2$, а разрыв находится в точке $\alpha = \pi$. Нетрудно видеть, что функция $\text{Re}(\xi(\tau, x))$ имеет при всех τ одну точку разрыва, совпадающую с точкой разрыва начального условия, что говорит об устойчивости такого решения.

Начальная функция для примера, изображенного на рисунке 5, вычислена для $\varphi = 1$. В этом случае разрыв находится в точке $\alpha \approx 5.03$. Расчеты показали, что решение, соответствующее этому условию, неустойчиво, и приближается к кусочно-постоянному решению с другими точками разрыва, причем их количество существенно увеличилось.

Функция, к которой сходится решение, приведенное на рисунке 5, тем не менее, является периодической по τ . Для того, чтобы численно проиллюстрировать это, зафиксируем две точки x_1 и x_2 из отрезка $[0, 2\pi]$ и построим по решению $\xi(\tau, x)$ параметрическую по τ кривую $(\text{Re}(\xi(\tau, x_1)), \text{Re}(\xi(\tau, x_2)))$. На рисунке 6 представлена такая кривая при $x_1 = 0.1$, $x_2 = 0.6$. Нетрудно видеть, что она

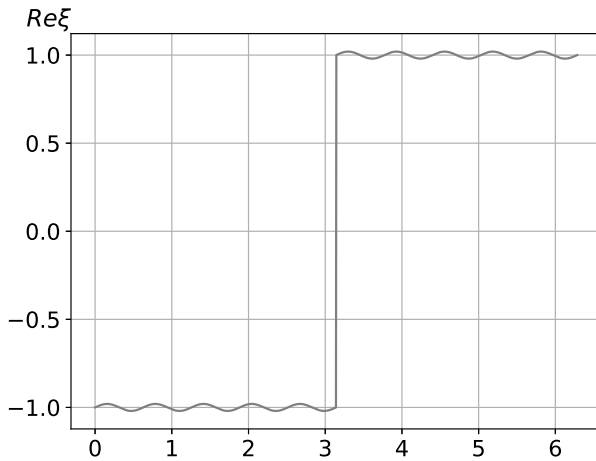


Fig. 4. Spatial distribution of the function $\text{Re}(\xi(t, x))$ at $\varphi = 0$

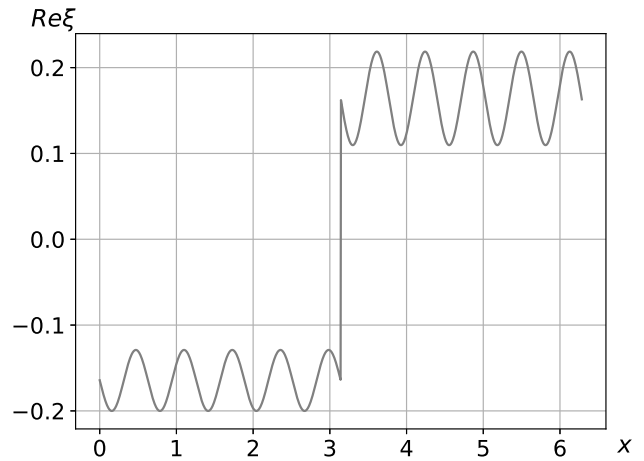


Рис. 4. Пространственное распределение функции $\text{Re}(\xi(t, x))$ при $\varphi = 0$

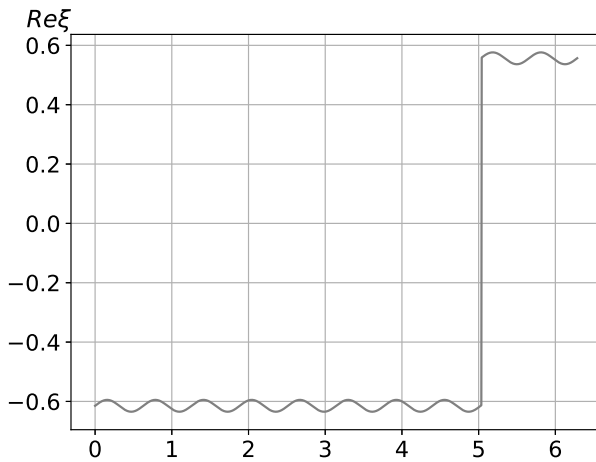


Fig. 5. Spatial distribution of the function $\text{Re}(\xi(t, x))$ for $\varphi = 1$

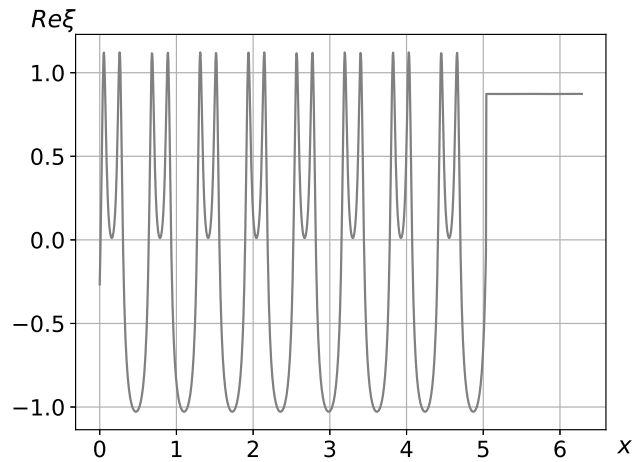


Рис. 5. Пространственное распределение функции $\text{Re}(\xi(t, x))$ при $\varphi = 1$

асимптотически приближается к замкнутой кривой. Это позволяет утверждать, что в пределах точности вычислений решение $\xi(\tau, x)$ близко к периодической по переменной τ функции.

5. Заключение

Подводя итог работы, отметим основные результаты. Интегро-дифференциальная система уравнений с периодическими краевыми условиями была исследована в простейших критических случаях, когда в задаче об устойчивости нулевого решения реализуется в одной ситуации нулевое, а в другой — пара чисто мнимых собственных чисел. В каждом из этих случаев удается построить квазинормальную форму, представляющую собой одно интегро-дифференциальное уравнение (вещественное или комплексное). Для полученных уравнений найдены условия существования континуальных семейств кусочно-постоянных двухступенчатых решений. Отдельной задачей оказалось исследование устойчивости этих решений. Отметим, что в случае, когда эти решения неустойчивы

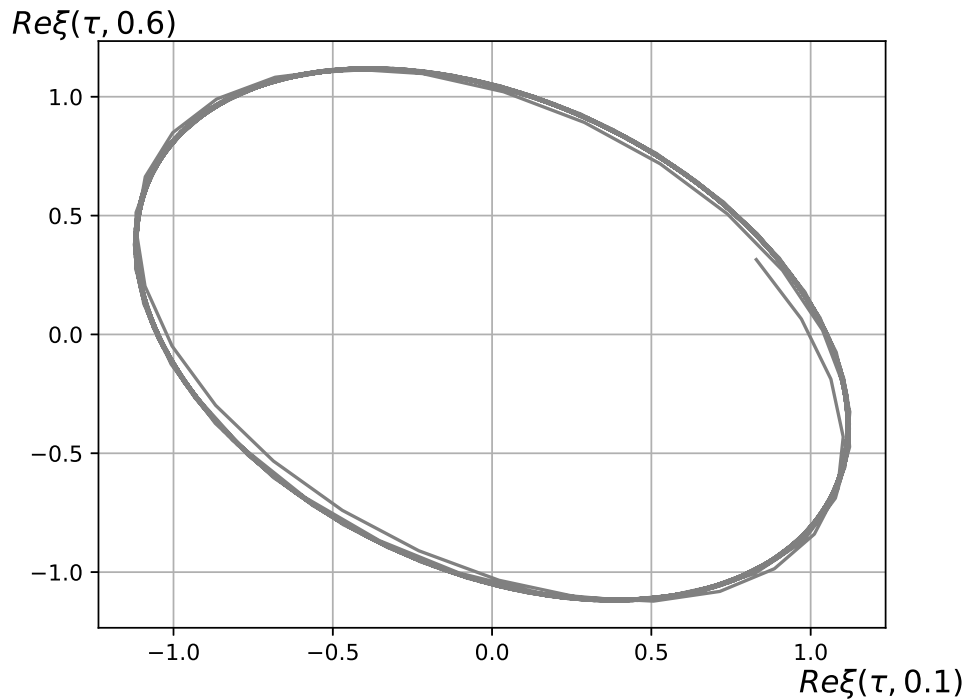


Fig. 6. Projection onto the plane of a parametrically defined curve $(\operatorname{Re}(\xi(\tau, 0, 1)), \operatorname{Re}(\xi(\tau, 0.6)))$

Рис. 6. Проекция на плоскость параметрически заданной кривой $(\operatorname{Re}(\xi(\tau, 0, 1)), \operatorname{Re}(\xi(\tau, 0.6)))$

удается найти периодические решения с числом точек разрыва по пространственной переменной больше двух.

References

- [1] S. A. Kashchenko, “Dynamics of full-coupled chains of a great number of oscillators with a large delay in couplings”, *Izvestiya VUZ. Applied Nonlinear Dynamics*, vol. 31, no. 4, pp. 523–542, 2023, in Russian. DOI: [10.18500/0869-6632-003054](https://doi.org/10.18500/0869-6632-003054).
- [2] D. S. Glyzin, S. D. Glyzin, and A. Y. Kolesov, “Hunt for chimeras in fully coupled networks of nonlinear oscillators”, *Izvestiya VUZ. Applied Nonlinear Dynamics*, vol. 30, no. 2, pp. 152–175, 2022, in Russian. DOI: [10.18500/0869-6632-2022-30-2-152-175](https://doi.org/10.18500/0869-6632-2022-30-2-152-175).
- [3] S. A. Kashchenko, “Quasinormal forms for chains of coupled logistic equations with delay”, *Mathematics*, vol. 10, no. 15, p. 2648, 2022. DOI: [10.3390/math10152648](https://doi.org/10.3390/math10152648).
- [4] S. A. Kashchenko, “Asymptotics of regular and irregular solutions in chains of coupled van der Pol equations”, *Mathematics*, vol. 11, no. 9, p. 2047, 2023. DOI: [10.3390/math11092047](https://doi.org/10.3390/math11092047).
- [5] S. D. Glyzin and A. Y. Kolesov, “Periodic two-cluster synchronization modes in fully coupled networks of nonlinear oscillators”, *Teoreticheskaya i Matematicheskaya Fizika*, vol. 212, no. 2, pp. 213–233, 2022, in Russian. DOI: [10.4213/tmf10191](https://doi.org/10.4213/tmf10191).
- [6] P. Hartman, *Ordinary differential equations*. Boston: Birkhäuser Baltimore, 1973.
- [7] J. Dalecki and M. Krein, *Stability of Solutions of Differential Equation in Banach Space*. AMS, Providence, 1974.
- [8] A. D. Myshkis, “Mixed functional differential equations”, *Journal of Mathematical Sciences*, vol. 129, pp. 4111–4226, 2005. DOI: [10.1007/s10958-005-0345-2](https://doi.org/10.1007/s10958-005-0345-2).

- [9] E. V. Grigorieva and S. A. Kashchenko, “Antiphase and in-phase dynamics in laser chain models with delayed bidirectional couplings”, *Physica D: Nonlinear Phenomena*, vol. 467, p. 134 223, 2024. DOI: [10.1016/j.physd.2024.134223](https://doi.org/10.1016/j.physd.2024.134223).
- [10] E. V. Grigorieva and S. A. Kashchenko, “Rectangular structures in the model of an optoelectronic oscillator with delay”, *Physica D: Nonlinear Phenomena*, vol. 417, p. 132 818, 2021. DOI: [10.1016/j.physd.2020.132818](https://doi.org/10.1016/j.physd.2020.132818).
- [11] J. Medlock and M. Kot, “Spreading disease: Integro-differential equations old and new”, *Mathematical Biosciences*, vol. 184, no. 2, pp. 201–222, 2003. DOI: [10.1016/S0025-5564\(03\)00041-5](https://doi.org/10.1016/S0025-5564(03)00041-5).
- [12] O. Bonnefon, J. Coville, J. Garnier, and L. Roques, “Inside dynamics of solutions of integro-differential equations”, *Discrete and Continuous Dynamical Systems – B*, vol. 19, no. 10, pp. 3057–3085, 2014. DOI: [10.3934/dcdsb.2014.19.3057](https://doi.org/10.3934/dcdsb.2014.19.3057).
- [13] Y. Jin and X.-Q. Zhao, “Spatial dynamics of a periodic population model with dispersal”, *Nonlinearity*, vol. 22, no. 5, p. 1167, 2009. DOI: [10.1088/0951-7715/22/5/011](https://doi.org/10.1088/0951-7715/22/5/011).
- [14] J. E. Marsden and M. McCracken, *The Hopf bifurcation and its applications. Applied Mathematical Sciences, volume 19*. Springer-Verlag, 1976.
- [15] B. D. Hassard, N. D. Kazarinoff, and Y.-H. Wan, *Theory and applications of Hopf bifurcation*. Cambridge; New York : Cambridge University Press, 1981.
- [16] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory, 3rd ed.* New York: Springer-Verlag, 2004.
- [17] Y. Kuang, *Delay Differential Equations with Applications in Population Dynamics*. Boston: Academic Press, 1993.
- [18] D. Henry, *Geometric Theory of Semilinear Parabolic Equations*. Springer, 1981.
- [19] J. Hale, *Theory of functional differential equations*. New-York : Springer-Verlag, 1971.
- [20] J. Wu, *Theory and applications of partial functional-differential equations*. New York: Springer-Verlag, 1996.
- [21] S. A. Kashchenko, “Quasinormal forms for parabolic equations with small diffusion”, *Soviet Mathematics. Doklady*, vol. 37, no. 5, pp. 510–513, 1988, in Russian.
- [22] S. A. Kashchenko, “Normalization in the systems with small diffusion”, *International Journal of Bifurcation and Chaos*, vol. 6, no. 6, pp. 1093–1109, 1996. DOI: [10.1142/S021812749600059X](https://doi.org/10.1142/S021812749600059X).