ISSN 2305-9052 (Print)
ISSN 2410-7034 (Online)



ВЕСТНИК ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

BULLETIN OF THE SOUTH URAL STATE UNIVERSITY

СЕРИЯ

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

2026, том 15, № 1

SERIES

COMPUTATIONAL MATHEMATICS AND SOFTWARE ENGINEERING

2026, volume 15, no. 1

1943

ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2026
Т. 15, № 1

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий, в которых должны быть опубликованы результаты диссертаций на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет (национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Искусственный интеллект и машинное обучение
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

Редакционная коллегия
Л.Б. Соколинский, д.ф.-м.н., проф., *гл. редактор*
М.Л. Цымблер, д.ф.-м.н., доц., *зам. гл. редактора*
Я.А. Краева, к.ф.-м.н., *отв. секретарь*
А.И. Гоглачев, *техн. редактор*

Редакционный совет
С.М. Абдуллаев, д.г.н., профессор
А. Андреяк, PhD, профессор (Германия)
В.И. Бердышев, д.ф.-м.н., акад. РАН, *председатель*
Дж. Донгарра, PhD, профессор (США)
С.В. Зыкин, д.т.н., профессор
И.М. Куликов, д.ф.-м.н.
Д. Маллманн, PhD, профессор (Германия)
А.В. Панюков, д.ф.-м.н., профессор
Р. Продан, PhD, профессор (Австрия)
Г.И. Радченко, к.ф.-м.н., доцент (Австрия)
В.Н. Ушаков, д.ф.-м.н., чл.-кор. РАН
М.Ю. Хачай, д.ф.-м.н., чл.-кор. РАН
А. Черных, PhD, профессор (Мексика)
П. Шумяцкий, PhD, профессор (Бразилия)

BULLETIN
OF THE SOUTH URAL STATE UNIVERSITY

2026
vol. 15, no. 1

SERIES
"COMPUTATIONAL MATHEMATICS AND SOFTWARE ENGINEERING"

ISSN 2305-9052

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya "Vychislitel'naya Matematika i Informatika"

South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- Artificial intelligence and machine learning
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

Editorial Board
L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russia)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russia)
Ya.A. Kraeva, South Ural State University (Chelyabinsk, Russia)
A.I. Goglachev, South Ural State University (Chelyabinsk, Russia)

Editorial Council
S.M. Abdullaev, South Ural State University (Chelyabinsk, Russia)
A. Andrzejak, Heidelberg University (Germany)
V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
J. Dongarra, University of Tennessee (USA)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
I.M. Kulikov, Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of RAS (Novosibirsk, Russia)
D. Mallmann, Julich Supercomputing Centre (Germany)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russia)
R. Prodan, Alpen-Adria-Universität Klagenfurt (Austria)
G.I. Radchenko, Silicon Austria Labs (Graz, Austria)
P. Shumyatsky, University of Brasilia (Brazil)
A. Tchernykh, CICESE Research Center (Mexico)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
S.V. Zykin, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Omsk, Russia)

Содержание

О ПРОЕКЦИОННОМ МЕТОДЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ
Л.Б. Соколинский, А.Э. Жулев, И.М. Соколинская 5

ИСПОЛЬЗОВАНИЕ СПУТНИКОВЫХ ИНДИКАТОРОВ БИОМАССЫ ФИТОПЛАНКТОНА В МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ РАСПРОСТРАНЕНИЯ ФОСФАТОВ В ТАГАНРОГСКОМ ЗАЛИВЕ
А.И. Сухинов, Н.Д. Панасенко, А.Е. Чистяков, А.В. Никитина 38

ПРИЧИННО-СЛЕДСТВЕННАЯ СЕМАНТИКА ДЛЯ НЕПРЕРЫВНО-ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ СО СЛАБОЙ ВРЕМЕННОЙ СТРАТЕГИЕЙ
А.Ю. Зубарев 57

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И АЛГОРИТМЫ ПОСТРОЕНИЯ ЭТОГРАММЫ ОЛЬФАКТОРНОГО ИССЛЕДОВАНИЯ ОДНОГО ОБЪЕКТА И ОДНОГО ПРОВЕРЯЕМОГО ЛИЦА
А.А. Солдатенко, Д.В. Семенова 80

СООТНОШЕНИЯ МЕЖДУ КЛАССАМИ РАЗБИЕНИЙ С ОГРАНИЧЕНИЕМ НА ЧИСЛО ЧАСТЕЙ
А.А. Самойлов 96

© Издательский центр ЮУрГУ, 2026

Contents

ON PROJECTION METHOD OF LINEAR PROGRAMMING
L.B. Sokolinsky, A.E. Zhulev, I.M. Sokolinskaya 5

USING SATELLITE INDICATORS OF PHYTOPLANKTON BIOMASS IN MATHEMATICAL MODELING OF PHOSPHATE SPREADING IN TAGANROG BAY
A.I. Sukhinov, N.D. Panasenko, A.E. Chistyakov, A.V. Nikitina 38

A CAUSAL SEMANTICS FOR TIME PETRI NETS WITH WEAK TIME POLICY
A.Y. Zubarev 57

MATHEMATICAL MODELING AND ALGORITHMS FOR PROBLEM OF CONSTRUCTING SCENT LINEUPS OF OLFACTORY ANALYSIS FOR SINGLE OBJECT AND SINGLE TEST SUBJECT
A.A. Soldatenko, D.V. Semenova 80

ON RELATIONS BETWEEN PARTITION CLASSES WITH A RESTRICTED NUMBER OF PARTS
A.A. Samoilov 96

This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

УДК 519.852, 514.172.45, 519.168 **DOI: 10.14529/cmse260101**

О ПРОЕКЦИОННОМ МЕТОДЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

© 2026 Л.Б. Соколинский, А.Э. Жулев, И.М. Соколинская

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: leonid.sokolinsky@susu.ru, zhulevae@susu.ru, irina.sokolinskaya@susu.ru

Поступила в редакцию: 14.11.2025

В статье рассматривается проблема разработки эффективных методов проекционного типа для решения задач линейного программирования (ЛП). Предложен новый гибридный проекционный алгоритм HAlEM (Hybrid Along Edges Movement), сочетающий идеи проекционного подхода и симплекс-метода. Алгоритм начинает работу из произвольной вершины многогранника допустимых решений и осуществляет движение вдоль его ребер к оптимальной вершине. Для вычисления направления движения используется оригинальный метод двух-факторной проекции, обеспечивающий высокую вычислительную точность для любых задач ЛП. Основным преимуществом HAlEM перед предшествующими разработками (AlFaMove, AlEM) является отказ от комбинаторного перебора всех возможных комбинаций гиперплоскостей за счет использования техники обновления матричного базиса, заимствованной из симплекс-метода. Это позволяет избежать экспоненциальной вычислительной сложности. В статье представлена параллельная версия алгоритма, основанная на модели параллельных вычислений BSF и схеме «мастер-рабочие», позволяющих получить эффективную реализацию для суперкомпьютерных систем кластерного типа. Проведены вычислительные эксперименты на эталонных задачах из репозитория Netlib-LP и на серии параметризованных задач. Результаты экспериментов демонстрируют, что HAlEM, в отличие от симплекс-метода, обладает более высоким ресурсом параллелизма, позволяя эффективно использовать до нескольких десятков процессорных узлов, и показывает хорошую масштабируемость с параллельной эффективностью не ниже 51%. При этом алгоритм обеспечивает высокую точность вычислений на уровне машинного нуля.

Ключевые слова: линейное программирование, алгоритм HAlEM, проекционный метод, параллельная реализация, MPI, кластерная вычислительная система, исследование масштабируемости, Netlb-LP.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Соколинский Л.Б., Жулев А.Э., Соколинская И.М. О проекционном методе линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2026. Т. 15, № 1. С. 5–37. DOI: 10.14529/cmse260101.

Введение

Линейное программирование (ЛП) [1, 2] является одной из самых востребованных математических моделей для решения задач оптимизации в индустрии, экономике, науке и многих других сферах человеческой деятельности [3–6]. Основным методом решения задач ЛП на практике является симплекс-метод [7], позволяющий эффективно решать широкий класс оптимизационных задач с числом неизвестных до 50 000 [8]. Однако симплекс-методу присущ ряд недостатков, среди которых выделим следующие. Во-первых, симплекс-метод в общем случае характеризуется ограниченной масштабируемостью и не допускает эффективного распараллеливания более чем на 32 процессорных узлах [9]. Во-вторых, симплекс-метод может зацикливаться на вырожденных задачах ЛП [10], что не гарантирует его сходимость к решению [11, 12]. Поэтому остается актуальной задача разработки альтернативных эффективных методов ЛП, свободных от указанных недостатков. Одной из таких альтернатив являются методы проекционного типа, строящие на поверхности многогранника допустимых решений путь к оптимальной точке.

Идея проекционного метода чрезвычайно проста. Пусть в евклидовом пространстве $\mathbb{R}^n$ задана задача ЛП стандартного вида

$$\arg\max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid A\boldsymbol{x} \leqslant \boldsymbol{b},\ \boldsymbol{x} \geqslant \boldsymbol{0}\}$$

с ограниченной областью допустимых решений, представляющей собой замкнутый выпуклый многогранник M. Стартуя из произвольной граничной точки $\boldsymbol{v}_0$ многогранника M, проекционный метод вычисляет точку

$$\boldsymbol{z} = \boldsymbol{v}_0 + \delta \boldsymbol{e}_{\boldsymbol{c}},$$

где $\boldsymbol{e}_{\boldsymbol{c}}$ — единичный вектор, сонаправленный с вектором $\boldsymbol{c}$, δ — некоторое положительное вещественное число (см. рис. 1). Затем выполняется метрическая проекция точки $\boldsymbol{z}$, дающая

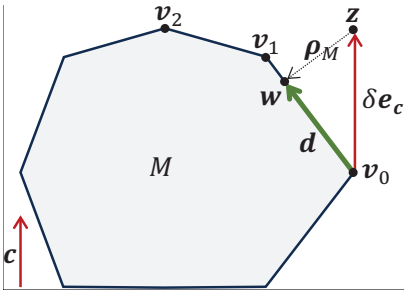


Рис. 1. Проекционный метод

на многограннике M точку $\boldsymbol{w}$:

$$\boldsymbol{w} = \boldsymbol{\rho}_M(\boldsymbol{z}).$$

Выбор числа δ должен гарантировать, что $\boldsymbol{w}$ будет на той же грани, что и $\boldsymbol{u}$. Далее вычисляется направляющий вектор $\boldsymbol{d} = \boldsymbol{w} - \boldsymbol{v}_0$, который позволяет получить следующее приближение $\boldsymbol{v}_1$:

$$\boldsymbol{v}_1 = \arg\max\{\|\boldsymbol{x} - \boldsymbol{v}_0\| \mid \boldsymbol{x} = \lambda \boldsymbol{d} + \boldsymbol{v}_0,\ \lambda \in \mathbb{R}_{>0},\ \boldsymbol{x} \in M\}.$$

Выполняя аналогичные действия для $\boldsymbol{v}_1$, получим приближение $\boldsymbol{v}_2$. Процесс заканчивается на k-том приближении, для которого

$$\boldsymbol{\rho}_M(\boldsymbol{v}_k + \delta \boldsymbol{e}_{\boldsymbol{c}}) = \boldsymbol{v}_k.$$

На рис. 1 этой ситуации соответствует точка $\boldsymbol{v}_2$. Таким образом получаем решение задачи ЛП.

В работе [13] описан апекс-метод решения задачи ЛП, реализующий этот подход в виде итерационного численного алгоритма. Для приближенного вычисления метрической проекции на выпуклый многогранник M апекс-метод использует M-фейеровское отображение $\boldsymbol{\varphi}_M : \mathbb{R}^n \to \mathbb{R}^n$, характеризующееся тем, что образ $\boldsymbol{\varphi}_M(\boldsymbol{x})$ любой точки $\boldsymbol{x} \notin M$ будет ближе к многограннику M, чем $\boldsymbol{x}$:

$$\forall \boldsymbol{x} \notin M, \forall \boldsymbol{y} \in M : \|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{y}\| < \|\boldsymbol{x} - \boldsymbol{y}\|.$$

Многократно применяя фейеровское отображение к произвольной внешней точке $\boldsymbol{x}^{(0)} \notin M$, мы получим точку на границе многогранника M, являющуюся некоторым приближением

метрической проекции. Указанный фейеровский процесс называется псевдопроектированием, а получившаяся точка — псевдопроекцией. Хотя апекс-метод способен решать некоторые реальные задачи ЛП, он обладает двумя существенными недостатками. Первый недостаток заключается в том, что фейеровский процесс, используемый для приближенного вычисления метрической проекции, имеет низкую линейную скорость сходимости:

$$\left\| \boldsymbol{x}^{(k+1)} - \boldsymbol{\varphi}_M \left(\boldsymbol{x}^{(0)} \right) \right\| \leqslant C q^k,$$

где $0 < C < +\infty$ — некоторая константа, а $q \in (0, 1)$ — параметр, зависящий от углов между гиперплоскостями, соответствующими граням многогранника M. Это означает, что расстояние между соседними приближениями с каждой итерацией уменьшается в геометрической прогрессии с коэффициентом, меньше единицы. Для малых углов скорость сходимости может падать до значений, близких к нулю. Второй недостаток связан с ограничением на длину «выводящего» вектора $\delta \boldsymbol{e_c}$: когда исходная граничная точка $\boldsymbol{v}$ находится близко к противоположному краю грани, значение δ должно быть достаточно малым, чтобы метрическая проекция не оказалась за пределами этой грани. Это существенно ограничивает точность вычисления метрической проекции.

В работах [14, 15] предложен алгоритм AlFaMove (Along Faces Movement), в котором для решения задачи ЛП предлагается вычислять метрическую проекцию не на многогранник, а на аффинное подпространство L, образующее грань допустимого многогранника M, как это показано на рис. 2. В этом случае выводящий вектор $\delta \boldsymbol{e_c}$ может иметь большую

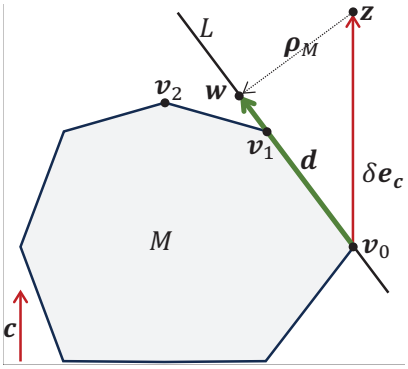


Рис. 2. Алгоритм AlFaMove

длину, что существенно увеличивает точность вычисления направляющего вектора $\boldsymbol{d}$. Для любой грани многогранника M существует комбинация гиперплоскостей из системы ограничений задачи ЛП, пересечением которых будет аффинное подпространство L, образующее эту грань. В соответствии с этим алгоритм AlFaMove перебирает все возможные комбинации гиперплоскостей, проходящих через точку $\boldsymbol{v}_0$. Для каждой комбинации вычисляется свой направляющий вектор $\boldsymbol{d}$. Отбрасываются комбинации, для которых луч $X = \{\boldsymbol{x} \in \mathbb{R} \mid \boldsymbol{x} = \boldsymbol{v}_0 + \lambda \boldsymbol{d},\ \lambda \in \mathbb{R}_{\geqslant 0}\}$ имеет с допустимым многогранником M только одну общую точку, являющуюся его начальной точкой $\boldsymbol{v}_0$. Для каждой оставшейся комбинации вычисляется точка $\boldsymbol{v}_{next}$ по формуле

$$\boldsymbol{v}_{next} = \boldsymbol{v}_0 + \frac{b_{i^*} - \langle \boldsymbol{a}_{i^*}, \boldsymbol{v}_0 \rangle}{\langle \boldsymbol{a}_{i^*}, \boldsymbol{d} \rangle} \boldsymbol{d}, \qquad \text{(i)}$$

где

$$i^* \in \operatorname{Arg}\min_i \left\{ \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v}_0 \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \;\middle|\; \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0 \right\}, \tag{ii}$$

i пробегает строки матрицы A. В качестве следующего приближения $\boldsymbol{v}_1$ из всех полученных точек $\boldsymbol{v}_{next}$ выбирается та, в которой достигается наибольшее значение целевой функции. Повторяя описанные действия, алгоритм AlFaMove за конечное число итераций придет в допустимую точку с максимальным значением целевой функции. Основным недостатком этого подхода является необходимость комбинаторного перебора возможных комбинаций гиперплоскостей, проходящих через точку текущего приближения. Если через точку проходит k гиперплоскостей, то количество комбинаций будет равно 2^k. Таким образом, алгоритм AlFaMove имеет экспоненциальную вычислительную сложность. Это делает затруднительным его применение даже в случае простых многогранников[1] размерности больше 30. Также отметим, что для вычисления метрической проекции алгоритм AlFaMove по-прежнему использует медленные фейеровские процессы.

В недавней работе [16] предложен новый проекционный алгоритм ЛП, получивший название AlEM (Along Edges Movement). В отличие от алгоритма AlFaMove, алгоритм AlEM начинает свою работу в произвольной вершине многогранника допустимых решений и двигается к оптимальной вершине только по ребрам. Для любого ребра существует $n-1$ гиперплоскостей из системы ограничений задачи ЛП, пересечением которых будет прямая, образующая это ребро. Если через вершину проходит k гиперплоскостей ($k \geqslant n$), то количество комбинаций может быть вычислено по следующей формуле:

$$C_k^{n-1} = \frac{k!}{(n-1)!(k-n+1)!}.$$

Для вычисления метрической проекции точки на прямую алгоритм AlEM вместо фейеровских отображений использует формулу ортогональной проекции на аффинное подпространство [17], что оказывается несравненно более эффективным. Вычислительные эксперименты показали, что быстродействие алгоритма AlEM на задачах ЛП, область допустимых решений которых является простым многогранником, сравнимо с быстродействием симплекс-метода. Однако, в отличии от симплекс-метода, алгоритм AlEM допускает эффективное распараллеливание и исключает возможность зацикливания на вырожденных задачах. К сожалению, в реальных задачах ЛП количество гиперплоскостей, проходящих через вершины многогранника допустимых решений, существенно превышает размерность пространства, что приводит к необходимости комбинаторного перебора.

В этой статье мы предлагаем новый проекционный алгоритм HAlEM (Hybrid Along Edges Movement), который совмещает в себе свойства алгоритма AlEM и симплекс-метода. Принцип работы алгоритма HAlEM заключается в следующем. В начальной вершине $\boldsymbol{v}_0$ формируется матричный базис A_0 из n линейно независимых строк матрицы A, соответствующих n гиперплоскостям, проходящим через $\boldsymbol{v}_0$. Каждая комбинация из $n-1$ таких гиперплоскостей определяет прямую, являющуюся результатом их пересечения. Всего получится n линейно независимых прямых, называемых реберными. Каждая реберная прямая проходит через ребро, инцидентное вершине $\boldsymbol{v}_0$. С помощью ортогональной проекции для каждой реберной прямой вычисляется направляющий вектор $\boldsymbol{d}$. Затем для всех реберных

[1] Выпуклый многогранник размерности n является простым, если из любой его вершины выходит ровно n ребер.

прямых с помощью формул (i), (ii) вычисляются точки $\boldsymbol{v}_{next}$, являющиеся смежными вершинами по отношению к $\boldsymbol{v}_0$. В качестве $\boldsymbol{v}_1$ выбирается та вершина, в которой достигается наибольшее значение целевой функции. На основе матричного базиса A_0 формируется новый матричный базис A_1 путем замены строки, не участвовавшей в формировании ребра перехода, на некоторую строку из матрицы A, соответствующую гиперплоскости, проходящей через вершину $\boldsymbol{v}_1$, так, чтобы сохранялась линейная независимость матричного базиса A_1. Процесс продолжается до тех пор, пока не будет достигнута вершина, у которой отсутствуют ребра, ведущие к увеличению целевой функции. Эта вершина будет искомым решением задачи ЛП.

Статья организована следующим образом. Раздел 1 содержит обозначения, определения и утверждения, необходимые для описания алгоритма HAlEM и его численной реализации. В разделе 2 дается формализованное описание алгоритма HAlEM. Раздел 3 посвящен параллельной версии алгоритма HAlEM. В разделе 4 представлены информация о программной реализации алгоритма HAlEM и результаты вычислительных экспериментов. В заключении суммируются полученные результаты и намечаются направления дальнейших исследований. В конце статьи приводится сводка используемых математических обозначений.

1. Необходимые определения и утверждения

В этом разделе приводятся определения и утверждения, необходимые для формального описания алгоритма HAlEM.

В евклидовом пространстве $\mathbb{R}^n$ размерности $n > 1$ имеется задача ЛП общего вида с системой из m ограничений, включающей в себя k линейных уравнений и $m - k$ линейных неравенств:

$$\begin{aligned} &\langle \boldsymbol{a}_1, \boldsymbol{x} \rangle = b_1, \\ &\cdots\cdots\cdots \\ &\langle \boldsymbol{a}_k, \boldsymbol{x} \rangle = b_k, \\ &\langle \boldsymbol{a}_{k+1}, \boldsymbol{x} \rangle \leqslant b_{k+1}, \\ &\cdots\cdots\cdots \\ &\langle \boldsymbol{a}_m, \boldsymbol{x} \rangle \leqslant b_m. \end{aligned} \tag{1}$$

Здесь и далее $\langle *, * \rangle$ обозначает скалярное произведение двух векторов. Предполагается, что система (1) включает в себя также неравенства вида

$$\begin{aligned} &-x_1 \leqslant 0, \\ &\cdots\cdots\cdots \\ &-x_n \leqslant 0. \end{aligned} \tag{2}$$

Необходимо найти точку в области допустимых решений системы (1), в которой достигается максимум линейной целевой функции

$$F(\boldsymbol{x}) = \langle \boldsymbol{c}, \boldsymbol{x} \rangle .$$

Для $J = \{j_1, \ldots, j_q\} \subseteq \{1, \ldots, m\}$, $|J| = q > 0$ определим матрицу

$$A_J = \begin{bmatrix} \boldsymbol{a}_{j_1} \\ \vdots \\ \boldsymbol{a}_{j_q} \end{bmatrix}$$

и столбец

$$\boldsymbol{b}_J = \begin{bmatrix} b_{j_1} \\ \vdots \\ b_{j_q} \end{bmatrix}.$$

Обозначим через $\bar{I}$ множество индексов уравнений системы (1), и через $\hat{I}$ — множество индексов неравенств системы (1):

$$\bar{I} = \{1, \ldots, k\},$$
$$\hat{I} = \{k+1, \ldots, m\}.$$

Тогда систему ограничений (1) можно представить в матричном виде

$$A_{\bar{I}}\boldsymbol{x} = \boldsymbol{b}_{\bar{I}},$$
$$A_{\hat{I}}\boldsymbol{x} \leqslant \boldsymbol{b}_{\hat{I}}.$$

Без ограничения общности мы можем считать, что матрица $A_{\bar{I}}$ имеет полный ранг:

$$\operatorname{rank}(A_{\bar{I}}) = k. \tag{3}$$

Мы также будем полагать, что

$$k < n,$$

так как в противном случае область допустимых решений вырождается в точку.

Известно, что область допустимых решений системы (1) представляет собой замкнутый выпуклый многогранник

$$M = \left\{\boldsymbol{x} \in \mathbb{R}^n \middle| A_{\bar{I}}\boldsymbol{x} = \boldsymbol{b}_{\bar{I}},\ A_{\hat{I}}\boldsymbol{x} \leqslant \boldsymbol{b}_{\hat{I}}\right\},$$

называемый допустимым. Мы будем предполагать, что допустимый многогранник M является ограниченным множеством. В этом случае задача ЛП всегда имеет решение. Поскольку система (1) включает в себя неравенства (2), то из ограниченности допустимого многогранника M следует, что общее число неравенств должно превышать размерность пространства:

$$m - k > n. \tag{4}$$

Каждому индексу $i \in \bar{I} \cup \hat{I}$ соответствует гиперплоскость

$$H_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle = b_i\}. \tag{5}$$

В совокупности эти гиперплоскости образуют грани допустимого многогранника M.

Определение 1. Пусть точка $\boldsymbol{v}$ является вершиной допустимого многогранника M. Базисным индексом вершины $\boldsymbol{v}$ будем называть множество индексов $\mathcal{V}$, удовлетворяющее сле-

дующим условиям:

$$\mathcal{V} \subseteq \hat{I}, \tag{6}$$

$$|\mathcal{V}| = n - k, \tag{7}$$

$$\forall i \in \mathcal{V} : \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle = b_i, \tag{8}$$

$$\operatorname{rank}\left(A_{\bar{I} \cup \mathcal{V}}\right) = n. \tag{9}$$

Квадратную матрицу $A_{\bar{I} \cup \mathcal{V}}$ размера $n \times n$ будем называть базисной.

С геометрической точки зрения это означает, что пересечение гиперплоскостей, соответствующих строкам базисной матрицы $A_{\bar{I} \cup \mathcal{V}}$, образует вершину $\boldsymbol{v}$:

$$\{\boldsymbol{v}\} = \bigcap_{j \in \bar{I} \cup \mathcal{V}} H_j.$$

Базисная матрица $A_{\bar{I} \cup \mathcal{V}}$ задает $n - k$ различных реберных прямых[2] проходящих через вершину $\boldsymbol{v}$:

$$\forall i \in \mathcal{V} : \; L_i = \bigcap_{j \in \bar{I} \cup \mathcal{E}_i} H_j, \tag{10}$$

где

$$\mathcal{E}_i = \mathcal{V} \setminus \{i\}.$$

Множество индексов $\mathcal{E}_i$, однозначно определяющих i-тую реберную прямую, будем называть реберным индексом. Реберную прямую L_i ($i \in \bar{I} \cup \mathcal{V}$) удобно представлять в параметрическом виде

$$L_i = \left\{ \boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \boldsymbol{v} + \lambda \boldsymbol{d}, \, \lambda \in \mathbb{R} \right\}.$$

Направляющий вектор $\boldsymbol{d} \in \mathbb{R}^n$ можно получить следующим образом. Выберем произвольный ненулевой «выводящий» вектор $\boldsymbol{g}$, не являющийся ортогональным по отношению к прямой L_i, и вычислим вторую точку $\boldsymbol{w} \neq \boldsymbol{v}$ на прямой L_i:

$$\boldsymbol{w} = \boldsymbol{\pi}_{\bar{I} \cup \mathcal{E}_i}(\boldsymbol{v} + \boldsymbol{g}).$$

Здесь $\boldsymbol{\pi}_{\bar{I} \cup \mathcal{E}_i}(*)$ обозначает ортогональную проекцию на подпространство[3] $\bigcap_{j \in \bar{I} \cup \mathcal{E}_i} H_j$, образующее реберную прямую L_i в соответствии с формулой (10). Ортогональная проекция на подпространство вычисляется через псевдообратную матрицу по известной формуле [17]:

$$\boldsymbol{\pi}_{\bar{I} \cup \mathcal{E}_i}(\boldsymbol{x}) = \boldsymbol{x} - A^T_{\bar{I} \cup \mathcal{E}_i} \left(A_{\bar{I} \cup \mathcal{E}_i} A^T_{\bar{I} \cup \mathcal{E}_i} \right)^{-1} \left(A_{\bar{I} \cup \mathcal{E}_i} \boldsymbol{x} - \boldsymbol{b}_{\bar{I} \cup \mathcal{E}_i} \right). \tag{11}$$

Заметим, что матрица $A_{\bar{I} \cup \mathcal{E}_i}$ имеет полностью строчный ранг в силу того, что базисная матрица $A_{\bar{I} \cup \mathcal{V}}$ имеет полный ранг и $\mathcal{E}_i \subset \mathcal{V}$. Поэтому матрица $A_{\bar{I} \cup \mathcal{E}_i} A^T_{\bar{I} \cup \mathcal{E}_i}$ является обратимой (см. утверждение 3F в [18]). В итоге получаем искомый направляющий вектор

$$\boldsymbol{d} = \boldsymbol{w} - \boldsymbol{v}.$$

Следующие утверждения понадобятся нам для выполнения прохода по ребру от одной вершины к другой.

[2] Реберная прямая — это прямая, которой принадлежит некоторое ребро многогранника.

[3] Здесь и далее мы опускаем прилагательное «аффинное».

Утверждение 1. [16] Пусть в контексте системы ограничений (1) заданы точка $\boldsymbol{v} \in M$, произвольный вектор $\boldsymbol{d} \in \mathbb{R}^n$ и полупространство

$$P_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \leqslant b_i\},$$

ограниченное гиперплоскостью

$$H_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i\},$$

где $i \in \hat{I}$. Определим луч, исходящий из точки $\boldsymbol{v}$ в направлении вектора $\boldsymbol{d}$:

$$X = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \boldsymbol{v} + \lambda \boldsymbol{d},\ \lambda \in \mathbb{R}_{\geqslant 0}\}.$$

Тогда

$$\text{если} \quad \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle \leqslant 0, \quad \text{то} \quad X \subset P_i, \tag{12}$$

$$\text{если} \quad \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0, \quad \text{то} \quad X \cap H_i = \left\{ \boldsymbol{v} + \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \boldsymbol{d} \right\}. \tag{13}$$

Другими словами, луч, исходящий из принадлежащей допустимому многограннику M точки $\boldsymbol{v}$ в направлении $\boldsymbol{d}$, пересечет гиперплоскость H_i в том и только в том случае, когда $\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0$. При этом точка пересечения может быть вычислена по формуле (13).

Утверждение 2. Пусть в контексте системы ограничений (1) заданы точка $\boldsymbol{v} \in M$ и вектор $\boldsymbol{d} \in \mathbb{R}^n$ такой, что

$$\forall i \in \bar{I}: \ \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle = 0. \tag{14}$$

Определим луч, исходящий из точки $\boldsymbol{v}$ в направлении вектора $\boldsymbol{d}$:

$$X = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \boldsymbol{v} + \lambda \boldsymbol{d},\ \lambda \in \mathbb{R}_{\geqslant 0}\}. \tag{15}$$

Тогда

$$X \cap M = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \lambda \boldsymbol{v} + (1 - \lambda)\boldsymbol{q},\ 0 \leqslant \lambda \leqslant 1\},$$

где

$$\boldsymbol{q} = \boldsymbol{v} + \gamma \boldsymbol{d}, \tag{16}$$

$$\gamma = \min \left\{ \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \middle| i \in \hat{I},\ \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0 \right\}. \tag{17}$$

Другими словами, пересечение луча X и допустимого многогранника M представляет собой отрезок между точками $\boldsymbol{v}$ и $\boldsymbol{q}$, где $\boldsymbol{q}$ вычисляется по формулам (16) и (17).

Доказательство. Прежде всего удостоверимся, что точка $\boldsymbol{q}$, вычисляемая по формуле (16), принадлежит лучу X. Поскольку $\boldsymbol{v} \in M$, то для всех $i \in \hat{I}$ справедливо

$$\langle \boldsymbol{a}_i, \boldsymbol{v} \rangle \leqslant b_i.$$

Это равносильно

$$b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle \geqslant 0.$$

Следовательно,

$$\text{если} \quad \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0, \quad \text{то} \quad \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \geqslant 0. \tag{18}$$

С учетом (17) отсюда получаем

$$\gamma \geqslant 0.$$

В соответствии с (15) и (16) это означает, что точка $\boldsymbol{q}$ принадлежит лучу X.

Теперь нужно удостовериться, что точка $\boldsymbol{q}$ принадлежит допустимому многограннику M. Сначала покажем, что точка $\boldsymbol{q}$, вычисляемая по формуле (16), удовлетворяет всем уравнениям системы ограничений (1). Возьмем произвольное $i \in \bar{I}$ и покажем, что $\boldsymbol{q}$ удовлетворяет уравнению

$$\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i. \tag{19}$$

Так как $\boldsymbol{v} \in M$, имеем

$$\langle \boldsymbol{a}_i, \boldsymbol{v} \rangle = b_i. \tag{20}$$

Подставляя $\boldsymbol{q}$ в левую часть неравенства (19) вместо $\boldsymbol{x}$, с учетом (14), (16) и (20) получаем:

$$\langle \boldsymbol{a}_i, \boldsymbol{q} \rangle = \langle \boldsymbol{a}_i, \boldsymbol{v} + \gamma \boldsymbol{d} \rangle = \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle + \gamma \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle = \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle = b_i.$$

То есть, $\boldsymbol{q}$ удовлетворяет уравнению (19).

Теперь покажем, что точка $\boldsymbol{q}$, вычисляемая по формулам (16) и (17), удовлетворяет всем неравенствам системы ограничений (1). Возьмем произвольное $i \in \hat{I}$ и покажем, что $\boldsymbol{q}$ удовлетворяет неравенству

$$\langle \boldsymbol{a}_i, \boldsymbol{q} \rangle \leqslant b_i. \tag{21}$$

Сначала предположим, что выполняется условие

$$\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle \leqslant 0. \tag{22}$$

В этом случае, в соответствии с (12), все точки луча X, включая $\boldsymbol{q}$, принадлежат полупространству P_i. Следовательно, $\boldsymbol{q}$ удовлетворяет неравенству (21).

Теперь предположим, что выполняется условие

$$\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle > 0. \tag{23}$$

Согласно (13) луч X пересекает гиперплоскость H_i в точке

$$\boldsymbol{v} + \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \boldsymbol{d}.$$

Это означает, что

$$\left\langle \boldsymbol{a}_i, \boldsymbol{v} + \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \boldsymbol{d} \right\rangle = b_i.$$

Отсюда получаем

$$\langle \boldsymbol{a}_i, \boldsymbol{v} \rangle + \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_i, \boldsymbol{d} \rangle} \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle = b_i.$$

С учетом (17), (18) и (23) из последней формулы следует

$$\langle \boldsymbol{a}_i, \boldsymbol{v} \rangle + \gamma \langle \boldsymbol{a}_i, \boldsymbol{d} \rangle \leqslant b_i,$$

что равносильно

$$\langle \boldsymbol{a}_i, \boldsymbol{v} + \gamma \boldsymbol{d} \rangle \leqslant b_i.$$

Вместе с (16) это дает

$$\langle \boldsymbol{a}_i, \boldsymbol{q} \rangle \leqslant b_i.$$

Таким образом, точка $\boldsymbol{q}$, вычисляемая по формулам (16) и (17), принадлежит допустимому многограннику M. Поскольку M выпуклый, это означает, что весь отрезок между точками $\boldsymbol{v}$ и $\boldsymbol{q}$ принадлежит этому многограннику.

Осталось убедиться, что точка $\boldsymbol{q}$ является граничной точкой допустимого многогранника M. Для этого достаточно показать, что для любого $\varepsilon > 0$ точка $\boldsymbol{q}+\varepsilon\boldsymbol{d}$ не принадлежит M. В соответствии с (13), (16) и (17) существует $i' \in \hat{I}$ такой, что

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{d} \rangle > 0 \tag{24}$$

и

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{v} + \gamma \boldsymbol{d} \rangle = b_{i'}.$$

Последнее равносильно

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{v} \rangle + \gamma \langle \boldsymbol{a}_{i'}, \boldsymbol{d} \rangle = b_{i'}.$$

В силу (24) отсюда следует

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{v} \rangle + \gamma \langle \boldsymbol{a}_{i'}, \boldsymbol{d} \rangle + \varepsilon \langle \boldsymbol{a}_{i'}, \boldsymbol{d} \rangle > b_{i'},$$

что эквивалентно

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{v} + \gamma \boldsymbol{d} + \varepsilon \boldsymbol{d} \rangle > b_{i'}.$$

Таким образом, с учетом (16) получаем

$$\langle \boldsymbol{a}_{i'}, \boldsymbol{q} + \varepsilon \boldsymbol{d} \rangle > b_{i'}.$$

Это означает, что точка $\boldsymbol{q} + \varepsilon\boldsymbol{d}$ не принадлежит полупространству $P_{i'}$ и, следовательно, не принадлежит допустимому многограннику M.

Утверждение доказано. □

Следствие 1. Пусть в контексте системы ограничений (1) имеется вершина $\boldsymbol{v}' \in M$, являющаяся начальной точкой ребра, соответствующего реберной прямой

$$L = \left\{ \boldsymbol{x} \in \mathbb{R}^n \middle| \boldsymbol{x} = \boldsymbol{v}' + \lambda \boldsymbol{d},\ \lambda \in \mathbb{R} \right\}.$$

Тогда вершина $\boldsymbol{v}'' \in M$, являющаяся конечной точкой этого ребра, может быть вычислена по формуле

$$\boldsymbol{v}'' = \boldsymbol{v}' + \frac{b_{j^*} - \langle \boldsymbol{a}_{j^*}, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle} \boldsymbol{d}, \tag{25}$$

где

$$j^* \in \operatorname{Arg\,min} \left\{ \frac{b_j - \langle \boldsymbol{a}_j, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle} \,\middle|\, j \in \hat{I},\ \langle \boldsymbol{a}_j, \boldsymbol{d} \rangle > 0 \right\}. \tag{26}$$

Гиперплоскость H_{j^*} в этом случае называется ограничивающей по отношению к реберной прямой L.

Утверждение 3. Пусть в контексте системы ограничений (1) имеется вершина $\boldsymbol{v}' \in M$ с базисным индексом $\mathcal{V}'$. Зафиксируем некоторый $i^* \in \mathcal{V}'$. Ему соответствует реберная прямая L_{i^*}, вычисляемая по формуле

$$L_{i^*} = \bigcap_{j \in \bar{I} \cup \mathcal{V}' \setminus \{i^*\}} H_j. \tag{27}$$

Пусть также известен направляющий вектор $\boldsymbol{d}$ прямой L_{i^*}:

$$L_{i^*} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \middle| \boldsymbol{x} = \boldsymbol{v}' + \lambda \boldsymbol{d},\ \lambda \in \mathbb{R} \right\}. \tag{28}$$

Пусть точка $\boldsymbol{v}''$ является конечной вершиной рассматриваемого ребра. Положим

$$J^* = \operatorname{Arg\,min} \left\{ \frac{b_j - \langle \boldsymbol{a}_j, \boldsymbol{v}' \rangle}{\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle} \middle| j \in \hat{I},\ \langle \boldsymbol{a}_j, \boldsymbol{d} \rangle > 0 \right\}. \tag{29}$$

Тогда для любого $j^* \in J^*$ множество

$$\mathcal{V}'' = \left(\mathcal{V}' \setminus \{i^*\} \right) \cup \{j^*\} \tag{30}$$

будет являться базисным индексом для вершины $\boldsymbol{v}''$.

Доказательство. Необходимо для вершины $\boldsymbol{v}''$ проверить выполнение условий (6)–(9) из определения 1. По построению очевидно, что выполняются условия (6) и (7).

Проверим выполнение условия (8). В соответствии со следствием 1 имеем

$$\boldsymbol{v}'' = \boldsymbol{v}' + \frac{b_{j^*} - \langle \boldsymbol{a}_{j^*}, \boldsymbol{v}' \rangle}{\langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle} \boldsymbol{d}.$$

Отсюда

$$\langle \boldsymbol{a}_{j^*}, \boldsymbol{v}'' \rangle = \left\langle \boldsymbol{a}_{j^*}, \boldsymbol{v}' + \frac{b_{j^*} - \langle \boldsymbol{a}_{j^*}, \boldsymbol{v}' \rangle}{\langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle} \boldsymbol{d} \right\rangle = \langle \boldsymbol{a}_{j^*}, \boldsymbol{v}' \rangle + \frac{b_{j^*} - \langle \boldsymbol{a}_{j^*}, \boldsymbol{v}' \rangle}{\langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle} \langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle = b_{j^*},$$

то есть,

$$\langle \boldsymbol{a}_{j^*}, \boldsymbol{v}'' \rangle = b_{j^*}. \tag{31}$$

Таким образом, условие (8) также выполняется.

Осталось проверить выполнение условия (9) для $\boldsymbol{v}''$. Так как $\mathcal{V}'$ является базисным индексом для $\boldsymbol{v}'$, то в соответствии с (9) имеем

$$\operatorname{rank}\left(A_{\bar{I} \cup \mathcal{V}'}\right) = n.$$

Поскольку $A_{\bar{I} \cup \mathcal{V}'}$ является квадратной матрицей размера $n \times n$, отсюда получаем

$$\operatorname{rank}\left(A_{\bar{I} \cup \mathcal{V}' \setminus \{i^*\}}\right) = n - 1. \tag{32}$$

Предположим, что для $\boldsymbol{v}''$ не выполняется условие (9). В сочетании с (32) это означает, что

$$\operatorname{rank}\left(A_{\bar{I} \cup (\mathcal{V}' \setminus \{i^*\}) \cup \{j^*\}}\right) = n - 1.$$

Согласно (27) в этом случае

$$L_{i^*} = \bigcap_{j \in \bar{I} \cup (\mathcal{V}' \setminus \{i^*\}) \cup \{j^*\}} H_j.$$

Вместе с (28) отсюда следует

$$\forall j \in \bar{I} \cup \left(\mathcal{V}' \setminus \{i^*\}\right) \cup \{j^*\} : \ \langle \boldsymbol{a}_j, \boldsymbol{d} \rangle = 0.$$

В частности,

$$\langle \boldsymbol{a}_{j^*}, \boldsymbol{d} \rangle = 0.$$

Получили противоречие с (29).

Утверждение доказано. □

2. Алгоритм HAlEM

Данный раздел посвящен описанию алгоритма HAlEM, строящего из произвольной вершины $\boldsymbol{v}$ путь вдоль ребер допустимого многогранника M к вершине, в которой достигается максимум целевой функции задачи ЛП. Псевдокод HAlEM представлен в виде алгоритма 1. Поясним шаги алгоритма 1. На шаге 1 вводятся размерность пространства n, количество ограничений m, количество уравнений k в системе ограничений, матрица коэффициентов A и столбец свободных членов (правых частей) $\boldsymbol{b}$ системы ограничений, градиент $\boldsymbol{c}$ линейной целевой функции и координаты начальной вершины $\boldsymbol{v}_0$ допустимого многогранника M. Эти данные являются глобальными в том смысле, что они доступны всем подпрограммам-функциям, используемым в алгоритме 1. Предполагается, что ограничения задачи ЛП приведены к виду (1). Также предполагается, что среди уравнений нет избыточных, то есть $\operatorname{rank}(A_{\bar{I}}) = k$, где k — число уравнений. Допускается случай, когда $k = 0$. Шаги 2 и 3 формируют множество индексов уравнений $\bar{I}$ и множество индексов неравенств $\hat{I}$, составляющих систему ограничений (1). Шаг 4 с помощью функции BasisIndex формирует базисный индекс $\mathcal{V}_0 \subseteq \hat{I}$ для начальной вершины $\boldsymbol{v}_0$. Такой базисный индекс существует в силу (4). Псевдокод функции BasisIndex представлен в виде алгоритма 2. Шаг 5 устанавливает счетчик итераций t в значение 0. На шаге 6 логической переменной *exit* присваивается значение «ложь».

Основной цикл **repeat**/**until** (шаги 7–36) выполняет последовательные переходы от вершины к вершине по ребрам допустимого многогранника, пока не будет достигнута оптимальная вершина. Один переход соответствует одной итерации этого цикла.

Вложенный цикл **loop** (шаги 8–35) выполняет переход от вершины $\boldsymbol{v}_t$ к следующей вершине $\boldsymbol{v}_{t+1}$ с бóльшим значением целевой функции. На шаге 9 точке $\boldsymbol{v}^*$ присваиваются координаты текущей вершины $\boldsymbol{v}_t$.

Следующий за этим цикл **for all** (шаги 10–22) находит для текущей вершины $\boldsymbol{v}_t$ ребро с индексом i^*, ведущее к вершине $\boldsymbol{v}^*$ с наибольшим значением целевой функции среди всех ребер с индексами из $\mathcal{V}_t$. Прокомментируем шаги в теле этого цикла. Шаг 11 с помощью функции DirectionVector вычисляет направляющий вектор $\boldsymbol{d}$ текущей реберной прямой L_i в направлении увеличения целевой функции. Псевдокод функции DirectionVector представлен ниже в виде алгоритма 3. Оператор **if** на шаге 12 проверяет выполнение условия $\boldsymbol{d} = \boldsymbol{0}$. В этом случае на шаге 13 выполняется оператор **continue**, осуществляющий досрочный переход к следующей итерации цикла **for all**. В противном случае на шаге 15 с помощью функции BoundingHyperplaneIndex вычисляется индекс j гиперплоскости H_j, ограничивающей текущую реберную прямую L_i. Псевдокод функции BoundingHyperplaneIndex представлен ниже в виде алгоритма 4. Шаг 16 вычисляет конечную вершину $\boldsymbol{v}_{nex}$ текущего ребра в соответствии с формулой (25). Если в операторе **if** на шаге 17 значение целевой

функции в вершине $\boldsymbol{v}_{nex}$ оказывается больше значения целевой функции в вершине $\boldsymbol{v}^*$, то вершине $\boldsymbol{v}^*$ присваиваются координаты вершины $\boldsymbol{v}_{nex}$ (шаг 18), индекс i текущей реберной прямой сохраняется в переменной i^* (шаг 19), а индекс соответствующей ограничивающей гиперплоскости H_j сохраняется в переменной j^* (шаг 20). Шаг 21 закрывает оператор **if**. Шаг 22 заканчивает цикл **for all**.

Алгоритм 1 Алгоритм HAlEM

```
1:  input n, m, k, A, b, c, v_0
2:  Ī := {1, ..., k}
3:  Î := {k + 1, ..., m}
4:  V_0 := BasisIndex(v_0)                                   // см. алгоритм 2
5:  t := 0
6:  exit := false
7:  repeat
8:      loop
9:          v* := v_t
10:         for all i ∈ V_t do
11:             d := DirectionVector(v_t, V_t \ {i})          // см. алгоритм 3
12:             if d = 0 then
13:                 continue
14:             end if
15:             j := BoundingHyperplaneIndex(v_t, d)         // см. алгоритм 4
16:             v_nex := v_t + ((b_j − ⟨a_j, v_t⟩) / ⟨a_j, d⟩) d
17:             if ⟨c, v_nex⟩ > ⟨c, v*⟩ then
18:                 v* := v_nex
19:                 i* := i
20:                 j* := j
21:             end if
22:         end for
23:         if v* ≠ v_t then
24:             v_{t+1} := v*
25:             V_{t+1} := (V_t \ {i*}) ∪ {j*}
26:             t := t + 1
27:             break
28:         end if
29:         V'_t := ScrollBasisIndex(v_t, V_t)               // см. алгоритм 5
30:         if V'_t = V_t then
31:             exit := true
32:             break
33:         end if
34:         V_t := V'_t
35:     end loop
36: until exit
37: output v_t
38: stop
```

Если в операторе **if** на шаге 23 удалось получить вершину $\boldsymbol{v}^*$ с координатами, отличными от координат текущей вершины $\boldsymbol{v}_t$, то выполняются следующие действия. На шаге 24 эта вершина становится следующим приближением $\boldsymbol{v}_{t+1}$. На шаге 25 для новой вершины $\boldsymbol{v}_{t+1}$ вычисляется базисный индекс $\mathcal{V}_{t+1}$ по формуле (30). Шаг 26 увеличивает на единицу счетчик итераций t. На шаге 27 оператор **break** осуществляет выход из цикла **loop**, после чего выполняется следующая итерация основного цикла **repeat**/**until**.

Если условие оператора **if** на шаге 23 не выполняется, происходит переход к шагу 29. Эта ситуация возникает, когда базисный индекс $\mathcal{V}_t$ оказался «тупиковым»: в нем отсутствуют ребра, ведущие к увеличению целевой функции. В этом случае на шаге 29 выполняется «прокручивание» базисного индекса вершины $\boldsymbol{v}_t$ с помощью функции ScrollBasisIndex, псевдокод которой представлен в виде алгоритма 5. В результате получается обновленный базисный индекс $\mathcal{V}'_t \neq \mathcal{V}_t$. В этом случае оператор **if** (шаги 30–33) пропускается, и выполняется шаг 34, который заменяет предыдущий базисный индекс на новый. После этого повторяется итерация цикла **loop** для текущей вершины $\boldsymbol{v}_t$, но уже с обновленным базисным индексом.

Если обновить базисный индекс невозможно, функция ScrollBasisIndex возвращает исходный базисный индекс. Это означает, что оптимальная вершина уже достигнута. В этом случае выполняется условие в операторе **if** на шаге 30, что приводит к выполнению шага 31, который устанавливает логическую переменную *exit* в значение «истина». После этого оператор **break** на 32 шаге производит выход из цикла **loop**. Поскольку *exit* = **true**, оператор **until** на шаге 36 завершает выполнение основного цикла **repeat**/**until**. Шаг 37 выводит координаты оптимальной вершины. Шаг 38 завершает работу алгоритма HAlEM.

Псевдокод функции BasisIndex, используемой на шаге 4 алгоритма 1 для построения базисного индекса начальной вершины, представлен в виде алгоритма 2. В силу предполо-

Алгоритм 2 Построение базисного индекса для начальной вершины

```
1:  function BasisIndex(v)
2:      V := ∅
3:      for j = k + 1, ..., n do                    // k — число уравнений в системе (1)
4:          for all i ∈ Î do
5:              if ⟨a_i, v⟩ = b_i then
6:                  if rank(A_{Ī∪V∪{i}}) = j then
7:                      V := V ∪ {i}
8:                      break
9:                  end if
10:             end if
11:         end for
12:     end for
13:     return V
14: end function
```

жения (3) система ограничений (1) не содержит избыточных уравнений[4]. Поскольку любая точка допустимого многогранника M должна удовлетворять всем уравнениям из системы (1), любой матричный базис в любой вершине $\boldsymbol{v}$ допустимого многогранника M должен

[4]Если система ограничений содержит избыточные уравнения, их можно элиминировать, включая в первоначально пустую матрицу A строки коэффициентов уравнений, увеличивающих ее ранг, и игнорируя строки коэффициентов уравнений, при добавлении которых ранг матрицы A остается неизменным.

включать в себя коэффициенты всех уравнений. Для того, чтобы получился полный матричный базис в вершине $\boldsymbol{v}$, к матрице $A_{\bar{I}}$ необходимо добавить $n-k$ строк коэффициентов неравенств, соответствующих ограничивающим гиперплоскостям, проходящим через $\boldsymbol{v}$, так, чтобы ранг получившейся квадратной матрицы был равен размерности пространства n. Алгоритм 2 вычисляет базисный индекс, содержащий индексы соответствующих неравенств. На шаге 2 создается пустой базисный индекс $\mathcal{V}$. Внешний цикл **for** выполняет $n-k$ итераций, добавляя каждый раз в $\mathcal{V}$ новый индекс i из $\hat{I}$ так, чтобы ранг матрицы $A_{\bar{I}\cup\mathcal{V}\cup\{i\}}$ оставался полным. Это обеспечивается путем выполнения вложенного цикла **for all** (шаги 4–9), который для каждого i-того неравенства на шаге 5 проверяет условие $\langle \boldsymbol{a}_i, \boldsymbol{v}\rangle = b_i$. Если это условие выполняется, то граничная гиперплоскость, соответствующая i-тому неравенству, проходит через вершину $\boldsymbol{v}$. В этом случае на шаге 6 проверяется условие $\operatorname{rank}(A_{\bar{I}\cup\mathcal{V}\cup\{i\}}) = j$. Выполнение этого условия означает, что матрица $A_{\bar{I}\cup\mathcal{V}\cup\{i\}}$ имеет полностроч ный ранг. В этом случае шаг 7 добавляет индекс i в $\mathcal{V}$, после чего оператор **break** на шаге 8 выполняет досрочный выход из вложенного цикла **for all**. Процесс продолжается, пока не будут выполнены все итерации внешнего цикла **for**. В итоге получается полный базисный индекс $\mathcal{V}$, который на шаге 13 возвращается в качестве результата функции BasisIndex.

Псевдокод функции DirectionVector, используемой на шаге 11 алгоритма 1 для вычисления направляющего вектора реберной прямой с реберным индексом $\mathcal{E}$, представлен в виде алгоритма 3. Реализация этого алгоритма основывается на предложенном нами вычисли-

Алгоритм 3 Вычисление направляющего вектора

1: **function** DirectionVector($\boldsymbol{v}, \mathcal{E}$)
2: $\quad\boldsymbol{e_c} := \boldsymbol{c}/\|\boldsymbol{c}\|$
3: $\quad\boldsymbol{z} := \boldsymbol{v} + \delta\boldsymbol{e_c}$
4: $\quad\boldsymbol{w} := \boldsymbol{\pi}_{\bar{I}\cup\mathcal{E}}(\boldsymbol{z})$
5: $\quad$**if** $\|\boldsymbol{w}-\boldsymbol{v}\| = 0$ **then**
6: $\quad\quad$**return 0**
7: $\quad$**end if**
8: $\quad\boldsymbol{d} := \boldsymbol{w} - \boldsymbol{v}$
9: $\quad\boldsymbol{e_d} := \boldsymbol{d}/\|\boldsymbol{d}\|$
10: $\quad\boldsymbol{z}' := \boldsymbol{v} + \delta\boldsymbol{e_d}$
11: $\quad\boldsymbol{w}' := \boldsymbol{\pi}_{\bar{I}\cup\mathcal{E}}(\boldsymbol{z}')$
12: $\quad\boldsymbol{d}' := \boldsymbol{w}' - \boldsymbol{v}$
13: $\quad$**return** $\boldsymbol{d}'$
14: **end function**

тельном методе, названным двух-факторной проекцией. Суть этого метода заключается в следующем. В контексте задачи ЛП (1) с помощью реберного индекса $\mathcal{E}$ задана реберная прямая $L_{\mathcal{E}}$, проходящая через вершину $\boldsymbol{v} \in M$. Необходимо найти направляющий вектор этой прямой, указывающий в сторону увеличения значения линейной целевой функции с градиентом $\boldsymbol{c}$. Для этого на шаге 2 алгоритма 3 вычисляется единичный вектор $\boldsymbol{e_c} = \boldsymbol{c}/\|\boldsymbol{c}\|$, сонаправленный с вектором $\boldsymbol{c}$. Шаг 3 находит точку $\boldsymbol{z} = \boldsymbol{v} + \delta\boldsymbol{e_c}$. Здесь δ — положительная вещественная константа, являющаяся параметром алгоритма. Шаг 4 с помощью формулы (11) вычисляет точку $\boldsymbol{w} = \boldsymbol{\pi}_{\bar{I}\cup\mathcal{E}}(\boldsymbol{z})$, являющуюся ортогональной проекцией точки $\boldsymbol{z}$ на прямую $L_{\mathcal{E}}$ (см. рис. 3). Если на шаге 5 выясняется, что расстояние между точками $\boldsymbol{w}$ и $\boldsymbol{v}$ равно нулю, это означает, что градиент $\boldsymbol{c}$ целевой функции перпендикулярен прямой $L_{\mathcal{E}}$,

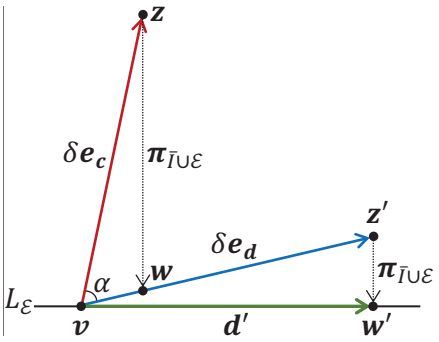


Рис. 3. Двух-факторная проекция

и она не может вести к увеличению целевой функции. В этом случае на шаге 6 функция DirectionVector в качестве результата возвращает нулевой вектор. В противном случае на шаге 8 получаем ненулевой вектор $\boldsymbol{d} = \boldsymbol{w} - \boldsymbol{v}$, направленный в сторону увеличения целевой функции. Точность вычисления координат точки $\boldsymbol{w}$ критически зависит от двух факторов: величины δ и угла α между векторами $\boldsymbol{c}$ и $\boldsymbol{d}$. Вычислительные эксперименты показали, что для достижения высокой точности на реальных задачах ЛП значение параметра δ должно находится в следующих пределах: $10^5 \leqslant \delta \leqslant 10^9$. Но этого оказывается недостаточно. Если величина угла α оказывается близкой к 90°, то наблюдается катастрофическая потеря точности для любых значений параметра δ. На рис. 3 такая потеря точности выражается в том, что точка $\boldsymbol{w}$ оказывается на некотором расстоянии от прямой $L_{\mathcal{E}}$. Для нейтрализации этого эффекта выполняется вторая фаза проектирования. На шаге 9 вычисляется единичный вектор $\boldsymbol{e_d} = \boldsymbol{d}/\|\boldsymbol{d}\|$, сонаправленный с вектором $\boldsymbol{d}$. Шаг 10 находит точку $\boldsymbol{z}' = \boldsymbol{v} + \delta\boldsymbol{e_d}$. Шаг 11 с помощью формулы (11) вычисляет точку $\boldsymbol{w}' = \boldsymbol{\pi}_{\bar{I}\cup\mathcal{E}}(\boldsymbol{z}')$, являющуюся ортогональной проекцией точки $\boldsymbol{z}'$ на прямую $L_{\mathcal{E}}$. Шаг 12 с высокой точностью вычисляет направляющий вектор $\boldsymbol{d}' = \boldsymbol{w}' - \boldsymbol{v}$ реберной прямой $L_{\mathcal{E}}$, указывающий в сторону увеличения целевой функции. Шаг 13 возвращает вектор $\boldsymbol{d}'$ в качестве результата функции DirectionVector.

Псевдокод функции BoundingHyperplaneIndex, используемой на шаге 15 алгоритма 1 для вычисления индекса j гиперплоскости H_j, ограничивающей реберную прямую L_i, представлен в виде алгоритма 4. Здесь $\boldsymbol{v}$ — вершина, через которую проходит реберная пря-

Алгоритм 4 Вычисление индекса ограничивающей гиперплоскости

```
1:  function BoundingHyperplaneIndex(v, d)
2:      γ_min := +∞                                  // см. формулы (17) и (18)
3:      for all j ∈ Î do
4:          if ⟨a_j, d⟩ > 0 then
5:              γ := (b_{j*} − ⟨a_{j*}, v⟩) / ⟨a_{j*}, d⟩
6:              if γ < γ_min then
7:                  γ_min := γ
8:                  j* := j
9:              end if
10:         end if
11:     end for
12:     return j*
13: end function
```

мая L_i, $\boldsymbol{d}$ — ее направляющий вектор. Алгоритм 4 в точности реализует вычисления по формуле (29) и не нуждается в пояснении. Заметим только, что ограничивающая гиперплоскость всегда существует в силу ограниченности допустимого многогранника M.

Следующий алгоритм реализует функцию «прокручивания» базисного индекса.

Алгоритм 5 Прокручивание базисного индекса для выхода из тупика

Require: $\boldsymbol{g} = (g_1, \ldots, g_n)$; $\boldsymbol{y} = (y_1, \ldots, y_n)$; $\boldsymbol{u} = (u_1, \ldots, u_m)$; $\mathcal{V} = \{v_1, \ldots, v_{n-k}\}$

function ScrollBasisIndex($\boldsymbol{v}, \mathcal{V}$)
  repeat
    $\boldsymbol{g} := \boldsymbol{c} A^{-1}_{\bar{I} \cup \mathcal{V}}$
    $\boldsymbol{u} := \boldsymbol{0}$
    for $i = 1, \ldots, n-k$ **do**
      $u_{v_i} := g_{k+i}$
    end for
    $i^* := 0$
    for $i = 1, \ldots, m$ **do**
      if $u_i < 0$ **then**
        $i^* := i$
        break
      end if
    end for
    if $i^* = 0$ **then**
      return $\mathcal{V}$
    end if
    for $j = 1, \ldots, n-k$ **do**
      if $v_j = i^*$ **then**
        for $i = 1, \ldots, n$ **do**
          $y_i := -A^{-1}_{\bar{I} \cup \mathcal{V}}[i, k+j]$
        end for
      end if
    end for
    $\lambda_{min} := +\infty$
    for all $j \in \hat{I}$ **do**
      if $\langle \boldsymbol{a}_j, \boldsymbol{y} \rangle > 0$ **then**
        $\lambda := \frac{b_j - \langle \boldsymbol{a}_j, \boldsymbol{v} \rangle}{\langle \boldsymbol{a}_j, \boldsymbol{y} \rangle}$
        if $\lambda < \lambda_{min}$ **then**
          $\lambda_{min} := \lambda$
          $j^* := j$
        end if
      end if
    end for
    $\mathcal{V} := (\mathcal{V} \setminus \{i^*\}) \cup \{j^*\}$
  until $\lambda_{min} > 0$
  return $\mathcal{V}$
end function

В некоторых случаях алгоритм HAlEM, находясь в вершине $\boldsymbol{v}_t$, может попасть на шаге 29 в «тупиковую» ситуацию, когда в базисном индексе $\mathcal{V}_t$ отсутствуют ребра, ведущие к увеличению целевой функции. С помощью комбинаторного перебора всевозможных сочетаний ограничивающих гиперплоскостей, проходящих через $\boldsymbol{v}_t$, всегда можно найти ребро, ведущее к увеличению целевой функции (если такое ребро не существует, мы уже находимся в оптимальной вершине). Этот метод был нами реализован в алгоритме AlEM [16]. Однако алгоритм AlEM имеет неприемлемо высокую экспоненциальную вычислительную сложность для многих практических задач ЛП. Вместо этого в алгоритме HAlEM на шаге 29 мы используем функцию ScrollBasisIndex, которая «прокручивает» в вершине $\boldsymbol{v}_t$ различные варианты базисного индекса $\mathcal{V}_t$ до тех пор, пока не будет найдена комбинация, в которой есть ребро, ведущее к увеличению целевой функции. Функция ScrollBasisIndex, представленная в виде алгоритма 5, для этой цели использует подход, применяемый в симплекс-методе, и детально описанный в разделе 11.1 монографии [19]. При написании алгоритма 5, реализующего функцию ScrollBasisIndex, мы сохранили обозначения, использованные в [19], за исключением следующих: вершине x_0 соответствует $\boldsymbol{v}_t$; матрице A_0 соответствует матрица $A_{\bar{I}\cup\mathcal{V}}$; столбцу b_0 соответствует $\boldsymbol{b}_{\bar{I}\cup\mathcal{V}}$. Для своей работы алгоритм 5 использует три вспомогательных вектора: $\boldsymbol{g}, \boldsymbol{y} \in \mathbb{R}^n$ и $\boldsymbol{u} \in \mathbb{R}^m$. Основной цикл **repeat/until** (шаги 2–32) осуществляет «прокрутку» базисных индексов, пока не будет найден вариант с ребром, ведущим к увеличению целевой функции. Шаг 3 вычисляет вспомогательный вектор $\boldsymbol{g}$. Шаги 4–7 заполняют координаты вектора $\boldsymbol{u}$ из двойственной задачи. В отличии от [19], мы заполняем нулями координаты $\boldsymbol{u}$, соответствующие уравнениям. На шагах 8–14 находим индекс i^*, соответствующий первой отрицательной координате вектора $\boldsymbol{u}$. Если в $\boldsymbol{u}$ таких координат нет, то мы находимся в оптимальной вершине. В этом случае функция ScrollBasisIndex на шаге 16 возвращает исходный базисный индекс. В противном случае на шагах 18–24 вычисляются координаты направляющего вектора $\boldsymbol{y}$ для i^*-го ребра. Шаги 25–34 вычисляют индекс j^* ограничивающей гиперплоскости подобно тому, как это делалось в алгоритме 4. Шаг 35 обновляет («прокручивает») базисный индекс $\mathcal{V}$, удаляя из него i^* и добавляя j^*. Процесс продолжается до тех пор, пока не будет получен вариант с $\lambda_{min} > 0$, означающий, что j^*-тое ребро ведет к увеличению целевой функции.

Следует отметить, что в редких случаях могут встретиться вырожденные задачи ЛП, для которых через определенное число итераций цикла **repeat/until** мы можем получить базисный индекс, который уже встречался ранее. В этом случае произойдет зацикливание алгоритма 5. Это характерно и для оригинального симплекс-метода, разработанного Данцигом. Для таких случаев предложены различные техники [11], позволяющие выйти из зацикливания. Однако обсуждение этих вопросов выходит за рамки настоящей статьи. Для невырожденных задач ЛП алгоритм HAlEM за конечное число шагов гарантированно сходится к оптимальной вершине.

3. Параллельная версия алгоритма HAlEM

Алгоритм 1 перебирает в цикле **for all** (шаги 10–22) $n-k$ ребер базисного индекса $\mathcal{V}_t$ с целью нахождения ребра, ведущего к вершине с наибольшим значением целевой функции. Этот цикл поддается эффективному распараллеливанию, что мы реализовали в параллельной версии алгоритма HAlEM.

Параллельная версия алгоритма HAlEM основана на модели параллельных вычислений BSF [20], ориентированной на кластерные вычислительные системы. Модель BSF использу-

ет схему распараллеливания «мастер–рабочие» и требует представление алгоритма в виде операций над списками с использованием функций высшего порядка Map и Reduce.

Функция высшего порядка Map(F, *MapList*) выполняет функцию F для всех элементов списка *MapList*, в результате чего формируется список *ReduceList*. Функция высшего порядка Reduce($\oplus$, *ReduceList*) попарно выполняет ассоциативную бинарную операцию $\oplus$ для всех элементов списка *ReduceList*, в результате чего получается один элемент.

В данном случае список *MapList* имеет длину $n-k$ и включает в себя все элементы базисного индекса $\mathcal{V}_t$, взятые в определенном порядке:

$$MapList = [i_1, \ldots, i_{n-k}], \tag{33}$$

где $\{i_1, \ldots, i_{n-k}\} = \mathcal{V}_t$. Семантика функции F_t определяется алгоритмом 6, шаги которого соответствуют шагам 11–16 алгоритма 1.

Алгоритм 6 Функция F_t

1: **function** $\mathrm{F}_t(i)$
2: $\quad \boldsymbol{d} := \mathrm{DirectionVector}(\boldsymbol{v}_t, \mathcal{V}_t \setminus \{i\})$ // см. алгоритм 3
3: $\quad$ **if** $\boldsymbol{d} = \boldsymbol{0}$ **then**
4: $\quad\quad$ **return** $(\boldsymbol{v}_t, i, 0)$
5: $\quad$ **end if**
6: $\quad j := \mathrm{BoundingHyperplaneIndex}(\boldsymbol{v}_t, \boldsymbol{d})$ // см. алгоритм 4
7: $\quad \boldsymbol{v}^* := \boldsymbol{v}_t + \frac{b_j - \langle \boldsymbol{a}_j, \boldsymbol{v}_t \rangle}{\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle} \boldsymbol{d}$
8: $\quad$ **return** $(\boldsymbol{v}^*, i, j)$
9: **end function**

Список *ReduceList*, получаемый в результате вызова функции высшего порядка Map(F_t, *MapList*), будет выглядеть следующим образом:

$$ReduceList = \left[(\boldsymbol{v}_1^*, i_1^*, j_1^*), \ldots, (\boldsymbol{v}_{n-k}^*, i_{n-k}^*, j_{n-k}^*)\right],$$

где для любого $l \in \{1, \ldots, n-k\}$ точка $\boldsymbol{v}_l^*$ является смежной вершиной по отношению к вершине $\boldsymbol{v}_t$, i_l^* — индекс соответствующего ребра, j_l^* — индекс гиперплоскости, ограничивающей соответствующую реберную прямую, за исключением случая, когда $j_l^* = 0$. В этом случае $\boldsymbol{v}_l^* = \boldsymbol{v}_t$.

Бинарная операция $\oplus$ над элементами списка *ReduceList* определяется следующей формулой:

$$\left(\boldsymbol{v}', i', j'\right) \oplus \left(\boldsymbol{v}'', i'', j''\right) = \begin{cases} \left(\boldsymbol{v}', i', j'\right), & \textbf{if } \langle \boldsymbol{c}, \boldsymbol{v}' \rangle > \langle \boldsymbol{c}, \boldsymbol{v}'' \rangle, \\ \left(\boldsymbol{v}'', i'', j''\right), & \textbf{if } \langle \boldsymbol{c}, \boldsymbol{v}' \rangle \leqslant \langle \boldsymbol{c}, \boldsymbol{v}'' \rangle, \end{cases} \tag{34}$$

семантика которой соответствует шагам 17–21 алгоритма 1.

Псевдокод параллельной версии алгоритма HAlEM представлен в виде алгоритма 7. Распараллеливанию подвергается цикл **for all** (шаги 10–22) последовательного алгоритма 1. Параллельные вычисления организуются по схеме «мастер–рабочие» и включают в себя $L+1$ процесс: один процесс–мастер и L процессов–рабочих, где $L \leqslant n-k$. Для простоты мы будем предполагать, что $n-k$ кратно L.

Процесс–мастер (далее — просто «мастер») выполняет последовательную часть алгоритма 7 и организует выполнение параллельной части процессами–рабочими (далее — про-

Алгоритм 7 Параллельная версия алгоритма HAlEM

мастер	**l-тый рабочий** $(l = 1, \ldots, L)$
1: **input** n, m, k, A, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{v}_0$	1: **input** n, m, k, A, $\boldsymbol{b}$, $\boldsymbol{c}$
2: $\bar{I} := \{1, \ldots, k\}$	2: $L :=$ **NumberOfWorkers**
3: $\hat{I} := \{k+1, \ldots, m\}$	3: $l :=$ **MyNumber**
4: $\mathcal{V}_0 := \mathrm{BasisIndex}(\boldsymbol{v}_0)$	4:
5: $t := 0$	5:
6: $exit :=$ **false**	6:
7: **repeat**	7: **repeat**
8: **loop**	8:
9: **Broadcast**$(\boldsymbol{v}_t, \mathcal{V}_t)$	9: **Recv**$(\boldsymbol{v}_t, \mathcal{V}_t)$ // $\mathcal{V}_t = \{i_1, \ldots, i_{n-k}\}$
10:	10: $K := \frac{n-k}{L}$
11:	11: $MapList_l := \left[i_{1+(l-1)K}, \ldots, i_{lK}\right]$
12:	12: $ReduceList_l := \mathrm{Map}(\mathrm{F}_t, MapList_l)$
13:	13: $(\boldsymbol{v}^*, i^*, j^*) := \mathrm{Reduce}(\oplus, ReduceList_l)$
14: **Gather**$(ReduceList^*)$	14: **Send**$(\boldsymbol{v}^*, i^*, j^*)$
15: $(\boldsymbol{v}^*, i^*, j^*) := \mathrm{Reduce}(\oplus, ReduceList^*)$	15:
16: **if** $\boldsymbol{v}^* \neq \boldsymbol{v}_t$ **then**	16:
17: $\boldsymbol{v}_{t+1} := \boldsymbol{v}^*$	17:
18: $\mathcal{V}_{t+1} := (\mathcal{V}_t \setminus \{i^*\}) \cup \{j^*\}$	18:
19: $t := t + 1$	19:
20: **break**	20:
21: **end if**	21:
22: $\mathcal{V}'_t := \mathrm{ScrollBasisIndex}(\boldsymbol{v}_t, \mathcal{V}_t)$	22:
23: **if** $\mathcal{V}'_t = \mathcal{V}_t$ **then**	23:
24: $exit :=$ **true**	24:
25: **break**	25:
26: **end if**	26:
27: $\mathcal{V}_t := \mathcal{V}'_t$	27:
28: **end loop**	28:
29: **Broadcast**$(exit)$	29: **Recv**$(exit)$
30: **until** $exit$	30: **until** $exit$
31: **output** $\boldsymbol{v}_t$	31:
32: **stop**	32: **stop**

сто «рабочие»). Шаги 1–8 мастера соответствуют шагам 1–8 последовательного алгоритма 1. На шаге 9 мастер с помощью системной функции **Broadcast** рассылает всем рабочим координаты текущей вершины $\boldsymbol{v}_t$ и соответствующий базисный индекс $\mathcal{V}_t$. На шаге 14 мастер с помощью системной функции **Gather** собирает результаты всех рабочих, представленных в виде троек вида $(\boldsymbol{v}^*, i^*, j^*)$, и организует их в единый список $ReduceList^*$. На шаге 15 мастер с помощью функции высшего порядка Reduce редуцирует список $ReduceList^*$ к одной тройке $(\boldsymbol{v}^*, i^*, j^*)$ путем попарного применения бинарной операции $\oplus$ по формуле (34). Далее мастер выполняет шаги 16–28, соответствующие шагам 23–35 последовательного ал-

горитма 1. На шаге 29 мастер с помощью системной функции **Broadcast** рассылает всем рабочим значение логической переменной *exit*, в зависимости от которого рабочие продолжают или заканчивают свою работу. Финальные шаги 30–32 мастера соответствуют финальным шагам 36–38 последовательного алгоритма 1.

Все рабочие выполняют одну и ту же последовательность шагов, но над разными данными. На шаге 1 каждый рабочий вводит исходные данные задачи ЛП. На шаге 2 с помощью системной функции **NumberOfWorkers** рабочий получает общее число рабочих L, задействованных в вычислениях. На шаге 3 с помощью системной функции **MyNumber** рабочий получает свой номер l. Далее рабочий входит в вычислительный цикл **repeat**/**until** (шаги 7–30). На шаге 9 рабочий с помощью системной функции **Recv** ожидает получения от мастера координат текущей вершины $\boldsymbol{v}_t$ и элементов текущего базисного индекса $\mathcal{V}_t$. На шаге 10 рабочий вычисляет длину K той части списка *MapList*, которую ему необходимо обработать. На шаге 11 рабочий строит свою часть списка $MapList_l$. На шаге 12 с помощью функции высшего порядка Map рабочий вычисляет список $ReduceList_l$. На шаге 13 рабочий с помощью функции высшего порядка Reduce редуцирует список $ReduceList_l$ к одной тройке $(\boldsymbol{v}^*, i^*, j^*)$ путем попарного применения бинарной операции $\oplus$ по формуле (34). На шаге 14 с помощью системной функции **Send** рабочий отсылает полученную тройку мастеру. После этого рабочий на шаге 29 с помощью системной функции **Recv** ожидает получения от мастера значения логической переменной *exit*. Если на шаге 30 переменная *exit* принимает значение **false**, рабочий переходит к следующей итерации вычислительного цикла **repeat**/**until**. В противном случает рабочий заканчивает свою работу.

Отметим, что с помощью системных функций **Recv** и **Gather** осуществляется неявная синхронизация параллельных процессов.

4. Реализация и вычислительные эксперименты

Мы реализовали параллельную версию алгоритма HAlEM на языке C++ с использованием программного BSF-каркаса [21], базирующегося на модели параллельных вычислений BSF [20]. BSF-каркас инкапсулирует все аспекты, связанные с распараллеливанием программы на основе библиотеки MPI. Исходные коды параллельной реализации свободно доступны в репозитории GitVerse по адресу `https://gitverse.ru/sokolinsky/HAlEM`. С использованием этой реализации были проведены вычислительные эксперименты по исследованию масштабируемости параллельной версии алгоритма HAlEM на различных задачах ЛП. Все эксперименты проводились на суперкомпьютере «Торнадо ЮУрГУ» [22], характеристики которого представлены в табл. 1. Для сборки программы использовался

Таблица 1. Характеристики суперкомпьютера «Торнадо ЮУрГУ»

Параметр	Значение
Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores, 3.33 GHz)
Число процессоров на узел	2
Память на узел	24 GB DDR3
Соединительная сеть	InfiniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Таблица 2. Тестирование параллельного алгоритма HAlEM на задачах Netlib-LP

Задача	m	n	d_M	d_M/n	δ	K_{peak}	$K_{\max}$	$T_{K_{\max}}$	T_1	α	ϵ
1	2	3	4	5	6	7	8	9	10	11	12
adlittle	153	97	82	0.85	0	6	5	0.55	2.22	4.04	0.81
afiro	59	32	24	0.75	2.5×10^{-16}	2	2	0.0015	0.0027	1.8	0.90
agg	651	163	127	0.78	0	10	6	2.57	10.40	3.84	0.67
agg2	818	302	242	0.80	7.4×10^{-16}	20	8	93.4	616.2	6.6	0.82
beaconfd	435	262	122	0.47	2.2×10^{-16}	10	6	9.18	31.62	3.44	0.57
blend	157	83	40	0.48	1.2×10^{-16}	3	3	0.35	0.52	1.5	0.51
israel	316	142	142	1	1.2×10^{-15}	11	7	5.46	29.09	5.33	0.76
kb2	93	41	25	0.61	0	2	2	0.022	0.030	1.34	0.67
recipe	366	180	92	0.51	0	7	6	2.07	6.94	3.35	0.56
sc105	207	103	58	0.56	0	4	4	0.13	0.32	2.47	0.62
sc50a	97	48	28	0.58	0	2	2	0.008	0.009	1.2	0.62
sc50b	96	48	28	0.58	2×10^{-16}	2	2	0.005	0.007	1.3	0.65
scagr7	269	140	56	0.4	0	4	4	0.81	2.13	2.6	0.66
share2b	175	79	66	0.83	1.5×10^{-15}	5	3	0.18	0.43	2.34	0.78
stocfor1	228	111	48	0.43	1.1×10^{-14}	4	4	0.18	0.43	2.32	0.58

компилятор g++, распространяемый в составе пакета компиляторов GCC 10, и библиотека Intel MPI 5.0. Компиляция выполнялась с опцией оптимизации O3. Задачи запускались на различном количестве процессорных узлов кластера. При этом общее число MPI-процессов было равно $12K$, где K — количество задействованных процессорных узлов. Таким образом, на каждом узле работало 12 MPI-процессов — по числу физических ядер.

В первой серии экспериментов мы исследовали ускорение и эффективность распараллеливания алгоритма HAlEM на эталонных задачах ЛП из репозитория Netlib-LP [23], доступного по адресу `https://netlib.org/lp/data`. В качестве начальной вершины $\boldsymbol{v}_0$ всегда выбиралась точка $\mathbf{0}$, если она являлась вершиной допустимого многогранника M. В противном случае начальная вершина вычислялась с помощью программы VeRSAl (Vertex Retrieve by Simplex Algorithm), написанной на языке C++, исходные тексты которой свободно доступны в репозитории GitVerse по адресу `https://gitverse.ru/sokolinsky/VeRSAl`. Программа VeRSAl строит на основе исходной задачи ЛП расширенную задачу ЛП в соответствии с методом, описанными в [19] (см. «Общий случай», стр. 199). Точка $\mathbf{0}$ по построению является вершиной допустимого многогранника для расширенной задачи ЛП. Программа VeRSAl решает эту задачу с помощью стандартного симплекс-метода, в результате чего получается вершина допустимого многогранника для исходной задачи ЛП.

Результаты экспериментов с эталонными задачами из репозитория Netlib-LP представлены в табл. 2. В столбце 1 перечислены имена задач из репозитория Netlib-LP, использованные для тестирования параллельного алгоритма HAlEM. Файлы со спецификациями этих задач в формате MPS [24] доступны в репозитории GitVerse по адресу `https://gitverse.ru/sokolinsky/Set-of-LP-Problems/content/master/NetLib-LP`. В этом же репозитории в формате MatrixMarket [25] сохранены координаты начальных вершин для всех указанных задач. В столбце 2 указано количество ограничений m для соответствующей задачи ЛП, включая неравенства, уравнения (если они присутствуют) и ограничения вида (2). В столбце 3 указано количество переменных n (размерность пространства решений). Столбец 4 содержит размерность d_M допустимого многогранника M, совпадающую с размерностью его аффинной оболочки: $d_M = \dim(M) = \dim(\text{aff}(M))$. Если

система ограничений не содержит избыточных уравнений и неявных равенств[5], то размерность допустимого многогранника M может быть вычислена по следующей формуле:

$$\dim(M) = n - k,$$

где k — количество уравнений в системе ограничений. В соответствии с этим количество уравнений k в системе ограничений задачи ЛП из табл. 2 может быть вычислено по формуле

$$k = n - d_M. \tag{35}$$

Столбец 5 содержит отношение размерности d_M допустимого многогранника к размерности n пространства решений. В столбце 6 указана относительная погрешность максимума целевой функции, вычисленного с помощью алгоритма HAlEM, в сравнении с «точным» значением, приведенным в работе Коха [26]:

$$\delta = \left| \frac{F_{Koch} - F_{HAlEM}}{F_{Koch}} \right|.$$

Здесь F_{HAlEM} — значение, вычисленное алгоритмом HAlEM, F_{Koch} — значение, указанное в работе [26]. Практически во всех случаях относительная погрешность оказалась в пределах машинного нуля (машинного эпсилон) [27], равного 2.2×10^{-16} для типа double (64 бита). Столь высокая точность вычислений была достигнута благодаря использованию метода двух-факторной проекции в алгоритме 3.

В столбце 7 приведена верхняя теоретическая граница K_{peak} масштабируемости алгоритма HAlEM, которая указывает максимальное число процессорных узлов, доступных для распараллеливания при использовании BSF-каркаса. Величина K_{peak} вычисляется следующим образом. Алгоритм 7 организует параллельные вычисления путем деления списка *MapList* на равные части по числу рабочих. Обработка этих подсписков выполняется рабочими на шаге 12 независимо друг от друга. Согласно (33) список *MapList* имеет длину $n - k$. Из (35) следует, что длина списка *MapList* равна d_M. Это означает, что для параллельной обработки списка *MapList* нельзя использовать более d_M рабочих. Учитывая, что при прогоне задачи на каждом процессорном узле запускалось 12 рабочих MPI-процессов, приходим к выводу, что

$$K_{\text{peak}} = \lfloor d_M/12 \rfloor .$$

Столбец 8 содержит реальную границу масштабируемости $K_{\max}$, определенную для каждой задачи ЛП в ходе вычислительных экспериментов. Под реальной границей масштабируемости параллельного алгоритма понимается число процессорных узлов вычислительного кластера, при превышении которого время решения задачи перестает сокращаться. В соответствии с характеристиками вычислительной платформы, использованной при проведении экспериментов (см. табл. 1), число процессорных ядер, задействованных для решения задачи на $K_{\max}$ узлах будет равно $K_{\max} \times 12$. В столбце 9 указано время $T_{K_{\max}}$ решения задачи ЛП на $K_{\max}$ узлах[6]. В столбце 10 указано время T_1 решения этой же задачи на одном процессорном узле. Значения ускорения, достигнутые на границе масштабируемости,

[5]Неравенство $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \leqslant b$ в системе ограничений $A\boldsymbol{x} \leqslant \boldsymbol{b}$ является неявным равенством, если $\langle \boldsymbol{a}, \boldsymbol{x} \rangle = b$ для всех $\boldsymbol{x}$, удовлетворяющих $A\boldsymbol{x} \leqslant \boldsymbol{b}$.

[6]Время везде указано в секундах.

представлены в столбце 11. Ускорение вычислялось по формуле

$$\alpha = \frac{T_1}{T_{K_{max}}}.$$

В столбце 12 содержится параллельная эффективность, достигаемая на границе масштабируемости. Параллельная эффективность вычислялась по формуле

$$\epsilon = \frac{T_1}{K_{max} \cdot T_{K_{max}}}.$$

Анализ полученных результатов позволяет сделать несколько выводов. Первый вывод касается границы масштабируемости $K_{\max}$. Для допустимых многогранников малой размерности ($d_M < 70$) теоретическая граница масштабируемости совпадает с реальной:

$$K_{\text{peak}} = K_{\max}.$$

Для допустимых многогранников средней размерности ($70 \leqslant d_M \leqslant 100$) теоретическая и реальная границы масштабируемости различаются не более чем на 1:

$$K_{\text{peak}} - K_{\max} \leqslant 1.$$

В случае допустимых многогранников большой размерности ($d_M > 100$) реальная граница масштабируемости может быть существенно меньше теоретической:

$$K_{\max} \ll K_{\text{peak}}.$$

Это связано с тем, что при использовании большого количества MPI-процессов для решения задачи ЛП методом HAlEM накладные расходы на организацию параллельного выполнения могут полностью нивелировать ускорение, получаемое при добавлении новых процессорных узлов.

Второй вывод связан с корреляцией между величиной d_M/n и эффективностью распараллеливания ϵ. Столбчатая диаграмма, представленная на рис. 4, показывает, что большому значению d_M/n соответствует высокая параллельная эффективность ϵ, и наоборот, низкая параллельная эффективность ϵ характерна для задач ЛП с небольшим значением d_M/n. Из этого ряда выпадает задача israel, которая характерна тем, что у нее единственной в системе ограничений отсутствуют уравнения, то есть размерность пространства решений совпадает с размерностью допустимого многогранника. Также отметим, что во всех случаях параллельная эффективность не опускалась ниже 51%, что является неплохим показателем для численного алгоритма.

Для сравнения разработанной параллельной версии алгоритма HAlEM с симплекс-методом мы использовали программу Simplex, исходные тексты которой на языке C++ свободно доступны по адресу `https://gitverse.ru/sokolinsky/Simplex`. Программа Simplex реализует параллельную версию классического симплекс-метода, описанного в [19] (стр. 197–198). Используя эту реализацию, мы решили все упомянутые задачи из репозитория Netlib-LP на той же вычислительной платформе, на которой исследовали HAlEM. Для всех задач ЛП программа Simplex запускалась на двух процессорных узлах по 12 MPI-процессов на узел. Использование большего количества процессорных узлов приводило к деградации ускорения программы Simplex. Результаты сравнения приведены в табл. 3. Эксперименты показали, что метод HAlEM обладает существенно большим

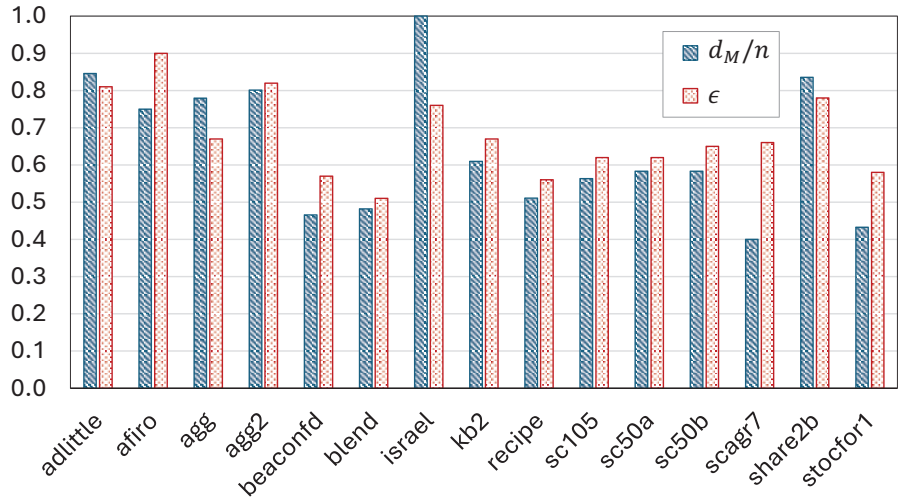


Рис. 4. Корреляция между d_M/n и параллельной эффективностью ϵ

Таблица 3. Сравнение метода HAlEM с симплекс-методом на задачах Netlib-LP

Задача	Процессорных узлов		Число итераций		Время (сек)		Относительная погрешность	
	HAlEM	Simplex	HAlEM	Simplex	HAlEM	Simplex	HAlEM	Simplex
adlittle	5	2	67	59	0.55	0.12	0	3.9×10^{-15}
afiro	2	2	3	4	0.0015	0.0009	2.5×10^{-16}	1.2×10^{-16}
agg	6	2	23	63	2.57	1.06	0	0
agg2	8	2	99	132	93.4	19.3	7.4×10^{-16}	0
beaconfd	6	2	15	23	9.18	2.07	2.2×10^{-16}	2.2×10^{-16}
blend	3	2	35	70	0.35	0.09	1.2×10^{-16}	4.7×10^{-14}
israel	7	2	146	160	5.46	1.26	1.2×10^{-15}	3.9×10^{-15}
kb2	2	2	23	76	0.02	0.02	0	4×10^{-15}
recipe	7	2	12	17	2.07	0.41	0	0
sc105	4	2	13	13	0.13	0.03	0	0
sc50a	2	2	7	7	0.008	0.002	0	0
sc50b	2	2	5	5	0.005	0.002	2×10^{-16}	0
scagr7	4	2	30	30	0.81	0.21	0	1.8×10^{-15}
share2b	3	2	27	33	0.18	0.05	1.5×10^{-15}	2.3×10^{-15}
stocfor1	4	2	12	22	0.18	0.07	1.1×10^{-14}	1.1×10^{-14}

ресурсом параллелизма, чем симплекс-метод. По количеству итераций оба метода имеют близкие показатели, однако симплекс-метод превосходит HAlEM по быстродействию на всех исследованных задачах ЛП. При этом разница в быстродействии находится в пределах одного порядка, что не является катастрофическим. Оба метода демонстрируют высокую точность вычислений на границе машинного нуля для типа double (64 бит). Однако существуют классы задач ЛП, на которых метод HAlEM показывает более высокую эффективность, чем симплекс-метод. В качестве примера можно привести циклические многогранники. Используя пример циклического многогранника $C_4(8)$ из [28] (стр. 29–30), мы сконструировали задачу ЛП zieglerC4_8, исходные данные которой в формате MPS можно найти по адресу `https://gitverse.ru/sokolinsky/Set-of-LP-Problems/content/master/Miscellaneous-LP`. Алгоритм HAlEM решает эту задачу за одну итерацию, в то время как программе Simplex для этого требуется 11 итераций.

В финальной серии экспериментов с параллельной версией алгоритма HAlEM мы исследовали зависимость ускорения и параллельной эффективности от размера решаемой задачи. С этой целью мы сконструировали специальную параметризованную задачу ЛП «гиперкуб с отсеченной вершиной», для которой размерность n пространства решений является параметром. Ограничения этой задачи содержат $2n+1$ неравенств следующего вида:

$$\begin{array}{rcl} x_1 & \leqslant & 200, \\ x_2 & \leqslant & 200, \\ & \vdots & \\ x_n & \leqslant & 200, \\ x_1 + x_2 + \cdots + x_n & \leqslant & 200(n-1)+100, \\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad \cdots \quad , x_n \geqslant 0. & & \end{array} \tag{36}$$

Градиент целевой функции задается вектором

$$\boldsymbol{c} = (1, 2, \ldots, n)\,.$$

Задача предполагает нахождение максимума целевой функции и имеет единственное решение в точке $(100, 200, \ldots, 200)$ со значением целевой функции, равным

$$F_{\max}(n) = 100(n^2 + n - 1). \tag{37}$$

Очевидно, что точка $\mathbf{0}$ является вершиной допустимого многогранника системы ограничений (36). Для произвольного n эта задача может быть получена в формате MatrixMarket [25] с помощью генератора FRaGenLP [29], если в качестве количества случайных неравенств задать 0. Исходные коды генератора FRaGenLP на языке C++ свободно доступны по адресу `https://gitverse.ru/sokolinsky/FRaGenLP`. Сгенерированные задачи для различных n доступны по адресу `https://gitverse.ru/sokolinsky/Set-of-LP-Problems/content/master/Tcube` под именами `lp_tcube<s>K<h>`, где в качестве `<s>` и `<h>` указаны две цифры, задающие размерность задачи: $n =$ sh00. В экспериментах мы использовали три задачи: tcube0K2, tcube0K3 и tcube0K4, информация о которых приведена в табл. 4. Семантика столбцов такая же, как в табл. 2, за исключением столбца $F_{\max}$, содержащего

Таблица 4. Тестирование HAlEM на задачах Tcube

Задача	m	n	$F_{\max}$	K_{peak}	$K_{\max}$	$T_{K_{\max}}$	T_1	α	ϵ
tcube0K2	401	200	4019900	16	16	16.1	136	8.5	0.53
tcube0K3	601	300	9029900	24	24	92.5	1126	12.2	0.51
tcube0K4	801	400	16039900	33	24	387	5681	14.7	0.61

максимальное значение целевой функции. Относительная погрешность для всех трех задач была равна нулю. Соответствующие графики ускорения и параллельной эффективности приведены на рис. 5. Отметим, что, как и в предыдущих экспериментах, на каждом процессорном узле запускалось 12 MPI-процессов. В качестве начальной вершины всегда выбиралась точка $\mathbf{0}$. Эксперименты показали, что для задач размерностей 200 и 300 теоретическая граница масштабируемости K_{peak} совпала с реальной $K_{\max}$. Однако при увеличении размерности задачи до 400, начиная с 25 узлов, накладные расходы на распараллеливание стали превалировать над ускорением. В этом случае реальная граница масштабируемости

оказалась заметно меньше теоретической. Отметим, что и в этой серии экспериментов параллельная эффективность на реальной границе масштабируемости не опускалась ниже 51%, что можно считать хорошим результатом.

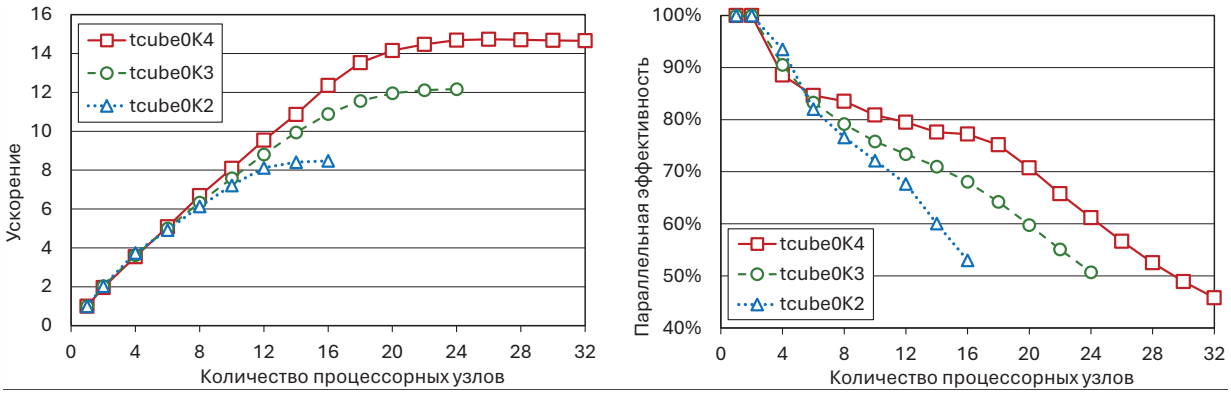


Рис. 5. Ускорение и параллельная эффективность алгоритма HAlEM

Заключение

В данной работе представлен новый проекционный алгоритм HAlEM для решения задач линейного программирования, который объединяет сильные стороны разработанного ранее авторами алгоритма AlEM (проекционный подход) и классического симплекс-метода (обновление матричного базиса). Основная идея HAlEM заключается в построении пути к оптимальной вершине путем последовательного перехода по ребрам допустимого многогранника, причем на каждом шаге для определения направления движения используется эффективная процедура двух-факторной ортогональной проекции, а для переключения на новое ребро применяется механизм ротации базисных индексов, аналогичный симплекс-методу. Такой гибридный подход позволил преодолеть главные недостатки предшествующих алгоритмов: избежать экспоненциального комбинаторного перебора (в отличие от AlFaMove и AlEM) и проблем, связанных с низкой скоростью сходимости фейеровских процессов при вычислении проекций (в отличие от апекс-метода).

Для предложенного алгоритма разработана параллельная версия, ориентированная на кластерные вычислительные системы. Распараллеливание, основанное на BSF-модели и схеме «мастер-рабочие», заключается в одновременной обработке всех ребер текущей вершины, что позволяет значительно сократить время выполнения итерации.

Проведенные вычислительные эксперименты на задачах из репозитория Netlib-LP и на серии специально сконструированных задач подтвердили работоспособность и эффективность алгоритма HAlEM. Основные выводы по результатам исследования можно сформулировать следующим образом. Благодаря использованию метода двух-факторной проекции алгоритм HAlEM демонстрирует высокую точность вычисления оптимального значения целевой функции, находящуюся в пределах машинного нуля для типа double (64 бита). Алгоритм HAlEM имеет ресурс параллелизма, существенно превосходящий симплекс-метод, что позволяет эффективно задействовать значительно большее количество процессорных узлов. Хотя по абсолютному времени счета на малом числе узлов HAlEM уступает симплекс-методу (разница в пределах одного порядка), на некоторых задачах (например, на циклических многогранниках) HAlEM может находить решение за меньшее число итераций.

Исследование масштабируемости показало, что граница эффективного распараллеливания алгоритма HAlEM напрямую зависит от размерности допустимого многогранника. Для задач с небольшой и средней размерностью теоретическая граница масштабируемости совпадает с реальной. При увеличении размерности накладные расходы на коммуникацию могут ограничивать рост ускорения, однако параллельная эффективность на реальной границе масштабируемости остается выше 50% во всех случаях, что является хорошим показателем для численного алгоритма.

Дальнейшие исследования планируется направить на разработку искусственной нейронной сети, которая позволит идентифицировать ребра допустимого многогранника быстрее, чем это делает алгоритм HAlEM с помощью двух-факторной проекции. Подобная модернизация позволит алгоритму HAlEM составить конкуренцию симплекс-методу по быстродействию на реальных задачах.

Обозначения

n	число переменных в системе ограничений (размерность пространства)
m	число ограничений
k	количество уравнений в системе ограничений
$\mathbb{R}^n$	вещественное евклидово пространство размерности n
$\langle \cdot, \cdot \rangle$	скалярное произведение двух векторов
$\|\cdot\|$	евклидова норма
$\bar{I}$	множество индексов ограничений в виде уравнений: $\bar{I} = \{1, \ldots, k\}$
$\hat{I}$	множество индексов ограничений в виде неравенств: $\hat{I} = \{k+1, \ldots, m\}$
A_J	матрица коэффициентов ограничений с индексами из $J \subseteq \{1, \ldots, m\}$
$\boldsymbol{b}_J$	столбец правых частей ограничений с индексами из $J \subseteq \{1, \ldots, m\}$
$\boldsymbol{a}_i$	i-тая строка матрицы A_J $(i \in J)$
P_i	полупространство, определяемое формулой $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \leqslant b_i$ $\left(i \in \hat{I}\right)$
H_i	гиперплоскость, определяемая формулой $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i$ $(i \in \{1, \ldots, m\})$
M	допустимый многогранник (область допустимых решений)
$\operatorname{rank}(A)$	ранг матрицы A
$\operatorname{aff}(X)$	аффинная оболочка множества X
$\dim(X)$	размерность множества X: $\dim(X) = \dim(\operatorname{aff}(X))$
$\boldsymbol{\pi}_J(\boldsymbol{x})$	ортогональная проекция точки $\boldsymbol{x}$ на подпространство $\bigcap\limits_{i \in J} H_i$

Литература

1. Pan P.-Q. Linear Programming Computation. Berlin, Heidelberg: Springer, 2014. 747 p. DOI: 10.1007/978-3-642-40754-3.
2. Bixby R.E. Solving Real-World Linear Programs: A Decade and More of Progress // Operations Research. 2002. Vol. 50, no. 1. P. 3–15. DOI: 10.1287/opre.50.1.3.17780.
3. Rasmussen S. Production Economics. The Basic Theory of Production Optimisation. 2nd ed. Berlin, Heidelberg: Springer, 2013. XII, 292 p. Springer Texts in Business and Economics. DOI: 10.1007/978-3-642-30200-8.
4. Xu Y. Solving Large Scale Optimization Problems in the Transportation Industry and Beyond Through Column Generation // Optimization in Large Scale Problems. Springer

Optimization and Its Applications, vol. 152. Cham: Springer, 2019. P. 269–292. DOI: 10.1007/978-3-030-28565-4_23.

5. Chung W. Applying large-scale linear programming in business analytics // 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2015. P. 1860–1864. DOI: 10.1109/IEEM.2015.7385970.
6. Gondzio J., Gruca J.A., Hall J.A.J., *et al.* Solving large-scale optimization problems related to Bell's Theorem // Journal of Computational and Applied Mathematics. 2014. Vol. 263. P. 392–404. DOI: 10.1016/j.cam.2013.12.003.
7. Dantzig G.B., Thapa M.N. Linear Programming 2: Theory and Extensions. Springer Series in Operations Research and Financial Engineering. New York, NY, USA: Springer, 2003. XXVI, 448 p. DOI: 10.1007/b97283.
8. Tolla P. A Survey of Some Linear Programming Methods // Concepts of Combinatorial Optimization / ed. by V.T. Paschos. 2nd ed. Hoboken, NJ, USA: John Wiley, Sons, 2014. Chap. 7. P. 157–188. DOI: 10.1002/9781119005216.ch7.
9. Mamalis B., Pantziou G. Advances in the Parallelization of the Simplex Method // Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295 / ed. by C. Zaroliagis, G. Pantziou, S. Kontogiannis. Cham: Springer, 2015. P. 281–307. DOI: 10.1007/978-3-319-24024-4_17.
10. Im H., Wolkowicz H. Revisiting degeneracy, strict feasibility, stability, in linear programming // European Journal of Operational Research. 2023. Vol. 310, no. 2. P. 495–510. DOI: 10.1016/j.ejor.2023.03.021.
11. Gass S.I., Vinjamuri S. Cycling in linear programming problems // Computers and Operations Research. 2004. Vol. 31, no. 2. P. 303–311. DOI: 10.1016/S0305-0548(02)00226-5.
12. Hall J., McKinnon K. The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling // Mathematical Programming, Series B. 2004. Vol. 100, no. 1. P. 133–150. DOI: 10.1007/s10107-003-0488-1.
13. Соколинский Л.Б., Соколинская И.М. О новой версии апекс-метода для решения задач линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 2. С. 5–46. DOI: 10.14529/cmse230201.
14. Ольховский Н.А., Соколинский Л.Б. О новом методе линейного программирования // Вычислительные методы и программирование. 2023. Т. 24, № 4. С. 408–429. DOI: 10.26089/NumMet.v24r428.
15. Соколинский Л.Б., Ольховский Н.А., Соколинская И.М. Численная реализация метода поверхностного движения для решения задач линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2024. Т. 13, № 3. С. 5–31. DOI: 10.14529/cmse240301.
16. Жулев А.Э., Соколинский Л.Б. О поиске оптимальной вершины на многограннике допустимых решений задачи линейного программирования // Вычислительные методы и программирование. 2026. Т. 27, № 1. С. 1–18. DOI: 10.26089/NumMet.v27r101.
17. Murty K.G. Computational and Algorithmic Linear Algebra and n-Dimensional Geometry. Singapore: World Scientific Publishing Company, 2014. 480 p. DOI: 10.1142/8261.
18. Стренг Г. Линейная алгебра и ее применения. Москва: Мир, 1980. 455 с.

19. Схрейвер А. Теория линейного и целочисленного программирования: в 2 т. Т. 1. Москва: Мир, 1991. 360 с.

20. Sokolinsky L.B. BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems // Journal of Parallel and Distributed Computing. 2021. Vol. 149. P. 193–206. DOI: 10.1016/j.jpdc.2020.12.009.

21. Sokolinsky L.B. BSF-skeleton: A Template for Parallelization of Iterative Numerical Algorithms on Cluster Computing Systems // MethodsX. 2021. Vol. 8. Article number 101437. DOI: 10.1016/j.mex.2021.101437.

22. Dolganina N., Ivanova E., Bilenko R., Rekachinsky A. HPC Resources of South Ural State University // Parallel Computational Technologies. PCT 2022. Communications in Computer and Information Science, vol. 1618 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2022. P. 43–55. DOI: 10.1007/978-3-031-11623-0_4.

23. Gay D.M. Electronic mail distribution of linear programming test problems // Mathematical Programming Society COAL Bulletin. 1985. Vol. 13. P. 10–12.

24. Муртаф Б. Современное линейное программирование. М.: Мир, 1984. 224 с.

25. Boisvert R.F., Pozo R., Remington K.A. The Matrix Market Exchange Formats: Initial Design: tech. rep. / NISTIR 5935. National Institute of Standards; Technology. Gaithersburg, MD, 1996. P. 14. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir5935.pdf.

26. Koch T. The final NETLIB-LP results // Operations Research Letters. 2004. Vol. 32, no. 2. P. 138–142. DOI: 10.1016/S0167-6377(03)00094-4.

27. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Vol. 37. 2nd ed. Berlin, Heidelberg: Springer, 2007. XVIII, 657 p. Texts in Applied Mathematics. DOI: 10.1007/b98885.

28. Циглер Г.М. Теория многогранников. М.: МЦНМО, 2014. 568 с.

29. Соколинский Л.Б., Соколинская И.М. О генерации случайных задач линейного программирования на кластерных вычислительных системах // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 38–52. DOI: 10.14529/cmse210103.

Соколинский Леонид Борисович, д.ф.-м.н., профессор, заведующий кафедрой системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Жулев Александр Эдуардович, аспирант кафедры системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Соколинская Ирина Михайловна, к.ф.-м.н., доцент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse260101

ON PROJECTION METHOD OF LINEAR PROGRAMMING

© 2026 L.B. Sokolinsky, A.E. Zhulev, I.M. Sokolinskaya

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: leonid.sokolinsky@susu.ru, zhulevae@susu.ru, irina.sokolinskaya@susu.ru

Received: 14.11.2025

The paper addresses the problem of developing efficient projection-type methods for linear programming (LP). A new hybrid projection algorithm called HAlEM (Hybrid Along Edges Movement) is proposed, combining ideas from the projection approach and the simplex method. The algorithm starts from an arbitrary vertex of the feasible solution polytope and moves along its edges toward the optimal vertex. An original two-factor projection method is used to compute the direction of movement, ensuring high computational accuracy for any LP problem. The main advantage of HAlEM over previous projection algorithms (AlFaMove, AlEM) is its avoidance of exhaustive combinatorial enumeration of all possible combinations of hyperplanes by employing a matrix basis update technique borrowed from the simplex method. This allows one to circumvent exponential computational complexity. The paper presents a parallel version of the algorithm based on the BSF parallel computing model and a master-worker scheme, enabling an efficient implementation for cluster-type supercomputer systems. Computational experiments were conducted on benchmark problems from the Netlib-LP repository and on a series of parameterized problems. The experimental results demonstrate that HAlEM, unlike the simplex method, possesses a higher degree of parallelism, allowing the efficient use of up to several dozen processor nodes, and exhibits good scalability with parallel efficiency not falling below 51%. Furthermore, the algorithm provides high computational accuracy at the level of machine epsilon.

Keywords: linear programming, HAlEM algorithm, projection method, parallel implementation, MPI, cluster computing system, scalability evaluation, Netlb-LP.

FOR CITATION

Sokolinsky L.B., Zhulev A.E., Sokolinskaya I.M. On Projection Method of Linear Programming. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2026. Vol. 15, no. 1. P. 5–37. (in Russian) DOI: 10.14529/cmse260101.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Pan P.-Q. Linear Programming Computation. Berlin, Heidelberg: Springer, 2014. 747 p. DOI: 10.1007/978-3-642-40754-3.
2. Bixby R.E. Solving Real-World Linear Programs: A Decade and More of Progress. Operations Research. 2002. Vol. 50, no. 1. P. 3–15. DOI: 10.1287/opre.50.1.3.17780.
3. Rasmussen S. Production Economics. The Basic Theory of Production Optimisation. 2nd ed. Berlin, Heidelberg: Springer, 2013. XII, 292 p. Springer Texts in Business and Economics. DOI: 10.1007/978-3-642-30200-8.
4. Xu Y. Solving Large Scale Optimization Problems in the Transportation Industry and Beyond Through Column Generation. Optimization in Large Scale Problems. Springer Optimization and Its Applications, vol. 152. Cham: Springer, 2019. P. 269–292. DOI: 10.1007/978-3-030-28565-4_23.

5. Chung W. Applying large-scale linear programming in business analytics. 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2015. P. 1860–1864. DOI: 10.1109/IEEM.2015.7385970.
6. Gondzio J., Gruca J.A., Hall J.A.J., *et al.* Solving large-scale optimization problems related to Bell's Theorem. Journal of Computational and Applied Mathematics. 2014. Vol. 263. P. 392–404. DOI: 10.1016/j.cam.2013.12.003.
7. Dantzig G.B., Thapa M.N. Linear Programming 2: Theory and Extensions. Springer Series in Operations Research and Financial Engineering. New York, NY, USA: Springer, 2003. XXVI, 448 p. DOI: 10.1007/b97283.
8. Tolla P. A Survey of Some Linear Programming Methods. Concepts of Combinatorial Optimization / ed. by V.T. Paschos. 2nd ed. Hoboken, NJ, USA: John Wiley, Sons, 2014. Chap. 7. P. 157–188. DOI: 10.1002/9781119005216.ch7.
9. Mamalis B., Pantziou G. Advances in the Parallelization of the Simplex Method. Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295 / ed. by C. Zaroliagis, G. Pantziou, S. Kontogiannis. Cham: Springer, 2015. P. 281–307. DOI: 10.1007/978-3-319-24024-4_17.
10. Im H., Wolkowicz H. Revisiting degeneracy, strict feasibility, stability, in linear programming. European Journal of Operational Research. 2023. Vol. 310, no. 2. P. 495–510. DOI: 10.1016/j.ejor.2023.03.021.
11. Gass S.I., Vinjamuri S. Cycling in linear programming problems. Computers and Operations Research. 2004. Vol. 31, no. 2. P. 303–311. DOI: 10.1016/S0305-0548(02)00226-5.
12. Hall J., McKinnon K. The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling. Mathematical Programming, Series B. 2004. Vol. 100, no. 1. P. 133–150. DOI: 10.1007/s10107-003-0488-1.
13. Sokolinsky L.B., Sokolinskaya I.M. Apex Method: A New Scalable Iterative Method for Linear Programming. Mathematics. 2023. Vol. 11, no. 7. P. 1–28. DOI: 10.3390/MATH11071654.
14. Olkhovsky N.A., Sokolinsky L.B. Surface Movement Method for Linear Programming. Lobachevskii Journal of Mathematics. 2024. Vol. 45, no. 10. P. 5061–5079. DOI: 10.1134/S1995080224605745.
15. Olkhovsky N.A., Sokolinsky L.B. AlFaMove: Scalable Implementation of Surface Movement Method for Cluster Computing Systems. Supercomputing Frontiers and Innovations. 2024. Vol. 11, no. 3. P. 4–26. DOI: 10.14529/jsfi240301.
16. Zhulev A.E., Sokolinsky L.B. Finding optimal vertex on polytope of feasible solutions to linear programming problem. Numerical Methods and Programming. 2026. Vol. 27, no. 1. P. 1–18. (in Russian) DOI: 10.26089/NumMet.v27r101.
17. Murty K.G. Computational and Algorithmic Linear Algebra and n-Dimensional Geometry. Singapore: World Scientific Publishing Company, 2014. 480 p. DOI: 10.1142/8261.
18. Strang G. Linear Algebra and Its Applications. 2nd. Academic Press, 1980. 425 p.
19. Schrijver A. Theory of Linear and Integer Programming. Chichester, New York, Brisbane, Toronto, Singapore: Wiley, Sons, 1998. 484 p.

20. Sokolinsky L.B. BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems. Journal of Parallel and Distributed Computing. 2021. Vol. 149. P. 193–206. DOI: 10.1016/j.jpdc.2020.12.009.

21. Sokolinsky L.B. BSF-skeleton: A Template for Parallelization of Iterative Numerical Algorithms on Cluster Computing Systems. MethodsX. 2021. Vol. 8. Article number 101437. DOI: 10.1016/j.mex.2021.101437.

22. Dolganina N., Ivanova E., Bilenko R., Rekachinsky A. HPC Resources of South Ural State University. Parallel Computational Technologies. PCT 2022. Communications in Computer and Information Science, vol. 1618 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2022. P. 43–55. DOI: 10.1007/978-3-031-11623-0_4.

23. Gay D.M. Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Bulletin. 1985. Vol. 13. P. 10–12.

24. Murtagh B.A. Advanced linear programming: computation and practice. New York, London: McGraw-Hill, 1981. xii, 202 p.

25. Boisvert R.F., Pozo R., Remington K.A. The Matrix Market Exchange Formats: Initial Design: tech. rep. / NISTIR 5935. National Institute of Standards; Technology. Gaithersburg, MD, 1996. P. 14. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir5935.pdf.

26. Koch T. The final NETLIB-LP results. Operations Research Letters. 2004. Vol. 32, no. 2. P. 138–142. DOI: 10.1016/S0167-6377(03)00094-4.

27. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Vol. 37. 2nd ed. Berlin, Heidelberg: Springer, 2007. XVIII, 657 p. Texts in Applied Mathematics. DOI: 10.1007/b98885.

28. Ziegler G.M. Lectures on Polytopes. Vol. 152. New York, NY: Springer New York, 1995. XI, 370 p. Graduate Texts in Mathematics. DOI: 10.1007/978-1-4613-8431-1.

29. Sokolinsky L.B., Sokolinskaya I.M. FRaGenLP: A Generator of Random Linear Programming Problems for Cluster Computing Systems. Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science, vol. 1437 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2021. P. 164–177. DOI: 10.1007/978-3-030-81691-9_12.

УДК 519.6, 004.94 **DOI: 10.14529/cmse260102**

ИСПОЛЬЗОВАНИЕ СПУТНИКОВЫХ ИНДИКАТОРОВ БИОМАССЫ ФИТОПЛАНКТОНА В МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ РАСПРОСТРАНЕНИЯ ФОСФАТОВ В ТАГАНРОГСКОМ ЗАЛИВЕ

© 2026 А.И. Сухинов, Н.Д. Панасенко, А.Е. Чистяков, А.В. Никитина

Донской государственный технический университет

(344003 Ростов-на-Дону, пл. Гагарина, д. 1)

E-mail: sukhinov@gmail.com, natalija93_93@mail.ru, cheese_05@mail.ru, nikitina.vm@gmail.com

Поступила в редакцию: 05.10.2025

Представлена нестационарная трехмерная математическая модель транспорта многокомпонентных примесей в мелководном водоеме, основанная на системе уравнений теории мелкой воды с учетом биогеохимических процессов. Модель позволяет рассчитывать пространственно-временное распределение концентраций биогенных веществ (фосфатов, соединений азота), фито- и зоопланктона, растворенного кислорода и сероводорода. Особенностью подхода является ассимиляция данных дистанционного зондирования Земли (спутники Sentinel-2, Landsat) через преобразование спектрального индекса NDCI в поле концентраций для инициализации и верификации модели. Разработан программный модуль на языке C++, в котором реализована конечно-разностная аппроксимация уравнений предложенной математической модели с использованием попеременно-треугольного итерационного метода. На примере Таганрогского залива Азовского моря проведена серия численных экспериментов по моделированию распространения фосфатов от точечного источника (глубоководный выпуск очистных сооружений в районе н.п. Дмитриадовка). Верификация модели выполнена путем сравнения результатов расчетов с обработанными спутниковыми данными и натурными замерами. Оценка погрешности по критерию Тейла составила около 14%. На основе анализа спутниковых снимков за период 2021–2024 гг. выявлена сезонная динамика границ «цветения» воды, обусловленная эвтрофикацией. Предложенный подход демонстрирует повышение точности прогноза по сравнению с моделями, не использующими ассимиляцию спутниковых данных.

Ключевые слова: численное моделирование, фосфаты, Таганрогский залив, ассимиляция данных, дистанционное зондирование Земли, индекс NDCI, математическая модель, программный модуль.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Сухинов А.И., Панасенко Н.Д., Чистяков А.Е., Никитина А.В. Использование спутниковых индикаторов биомассы фитопланктона в математическом моделировании распространения фосфатов в Таганрогском заливе // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2026. Т. 15, № 1. С. 38–56. DOI: 10.14529/cmse260102.

Введение

Значительный вклад в развитие методов математического моделирования геофизической гидродинамики и распространения примесей внесли работы Кибеля И.А. [1], Марчука Г.И. [2] и Монина А.С. [3]. Фундаментальный вклад в развитие численных методов математического моделирования процессов гидродинамики внесли труды академика РАН Самарского А.А. В области моделирования морских экосистем и распространения биогенных веществ значимы работы Матишова Г.Г., Лыкосова В.Н., Якушева Е.В. и др. [4–6]. Разработкой трехмерных математических моделей транспорта загрязняющих веществ в прибрежных системах занимался Холлинг К.С. В работах [7–10] приводятся результаты исследований водных экосистем на предмет эвтрофикации в том числе с учетом загряз-

нения многочисленными биогенными веществами. Численному моделированию распределения соединений фосфора, азота и кремния в воде посвятил свои труды Якушев Е.В. и последователи [11–14].

Современные исследования по моделированию трансформации биогенных элементов (азота, фосфора) в эстуариях рек и прибрежных зонах демонстрируют важность учета гидробиогеохимических процессов [6, 12]. В условиях Азовского моря и Таганрогского залива актуальны работы по спутниковому мониторингу фитопланктона и его сезонной динамике [15, 16]. Методы усвоения спутниковых данных в моделях морских экосистем позволяют повысить точность прогноза распределения загрязняющих веществ [17, 18]. Однако, для достижения цели исследования в математической модели транспорта загрязняющих веществ (ЗВ), должны учитывать не только гидрологические условия, но и гидрохимические, биологические и экологические факторы, включая продукционно-деструкционные процессы в основных биотопах конкретного водоема. Расчеты основных гидрофизических характеристик должны производиться с учетом динамически перестраиваемой геометрии расчетной области [19–23].

Ниже приведено описание существующих математических моделей распространения загрязняющих веществ. В работах [17, 24] описан комплекс математических моделей, обладающих следующими свойствами: математическая модель на основе «мелкой воды» включает систему уравнений для двумерного случая с целью изучения гидродинамических процессов; строится с учетом испарения с поверхности, осадков, задается линейной интерполяцией результатов; разработана с учетом сложной береговой линии; включает систему уравнений транспорта многокомпонентных загрязнений; для расчетов используется криволинейная сетка на основе оптимизации сеточных функционалов качества; реализована конечно-элементная аппроксимация по пространственным переменным. Выделим основные свойства математической модели, описанной в работе [14]: включает уравнения теории «мелкой воды» с сохранением нелинейных членов; в ней реализован учет возмущения поверхности воды; разработан итерационный метод специального вида, применяемый для решения СЛАУ с несимметричной исходной матрицей. Стоит отметить, что модели распространения загрязняющих веществ, описанные выше, эффективно справляются с решением конкретных задач и учитывают широкий спектр параметров. Однако, для достижения цели исследования, математическая модель и ее численная реализация в виде программного модуля должны из себя представлять удобный прикладной инструментарий, в котором учитываются необходимые для расчета параметры, а также динамически изменяющаяся расчетная область.

Существует ряд программных комплексов, предназначенных для моделирования гидрофизических процессов. ShallowHMT2.0 — программа для научных расчетов, предназначенная для моделирования процессов гидрофизики, а также распространения тепла и солей в мелководных водоемах с учетом движения свободной поверхности [25]. Она может использоваться как самостоятельный инструмент для изучения и прогнозирования последствий антропогенного воздействия, сценариев развития чрезвычайных ситуаций (ЧС). В модели учитывается: вертикальный турбулентный обмен; переменная плотность жидкости; движение свободной поверхности водоема. Ее численная реализация представляет собой модуль для расчета полей температуры, солености, а также для моделирования эволюции биологических систем мелководных водоемов. Программный комплекс STREAM 2D предназначен для расчета течений, деформаций дна и переноса загрязнений в открытых потоках с ис-

пользованием технологии OpenMP (на многопроцессорных системах с общей памятью) [26]. Пакет программ Ansys Fluent представляет собой набор решателей и пре-/постпроцессоров, предназначенный для моделирования полей течений жидкостей и газов с учетом турбулентности, межфазного взаимодействия, химических реакций, горения и аэроакустических эффектов [27]. Программное обеспечение для анализа Star-CD предназначено для моделирования процессов гидро-газовой динамики и тепло-массопереноса [28]. FlowVision — Российский CAE-пакет. Он широко используется для междисциплинарного моделирования устройств и физических процессов [29]. Программа PHOENICS представляет собой модуль для исследования процессов теплопроводности и конвекции [30]. Пакеты являются универсальными и широко используются для построения и численной реализации математических моделей гидродинамических и гидрофизических процессов. Данные программные комплексы в том числе ориентированы и на многопроцессорные системы, но при этом они позволяют решать достаточно узкий круг задач. В них заложен крайне ограниченный набор моделей и алгоритмов, при расширении которого задачи будут решаться значительно эффективней.

В данной работе предлагается трехмерная нестационарная математическая модель транспорта многокомпонентных примесей в мелководном водоеме, основанная на системе уравнений теории мелкой воды с учетом биогеохимических процессов. Модель реализована в виде программного модуля на языке C++ и включает процедуру ассимиляции спутниковых данных для верификации и начальной инициализации полей концентраций основных загрязняющих веществ, фито- и зоопланктона. Статья организована следующим образом. В разделе 1 представлены материалы и методы исследования, включая описание используемых данных и процедуры ассимиляции спутниковых наблюдений. Раздел 2 содержит постановку математической задачи гидрофизики мелководного водоема и описание разностной схемы – дискретного аналога непрерывной математической модели. В разделе 3 приведены результаты численных экспериментов по моделированию распространения фосфатов, включая верификацию модели и анализ точности. В заключении подведены итоги работы и сформулированы основные выводы.

1. Материалы и методы

В исследовании использовались данные различных типов, включая спутниковые наблюдения, гидрометеорологические данные и результаты натурных измерений, полученных в ходе экспедиционных работ в акватории Таганрогского залива Азовского моря. Ниже приведем их детальное описание.

1.1. Данные

Для анализа состояния акватории использовались данные дистанционного зондирования Земли (ДЗЗ), доступные через открытый каталог [31], со спутников Sentinel-2 (Европейское космическое агентство) и Landsat-8/9 (NASA/USGS). Снимки Sentinel-2 имеют разрешение до 10 м с периодичностью съемки 5 дней. Снимки Landsat-8/9 имеют разрешение до 30 м с периодичностью 16 дней. Все снимки были предварительно обработаны: проведена атмосферная коррекция, географическая привязка, калибровка. Для выделения водной поверхности использовался нормализованный разностный индекс воды (NDWI).

Для задания граничных условий и верификации модели использовались гидрометеорологические данные: данные реанализа ERA5 (ECMWF) о полях приземного ветра, температуре воздуха, влажности, атмосферном давлении с пространственным разрешением 0.25°

и временным разрешением 1 час; данные о речном стоке р. Дон (гидропост г. Ростов-на-Дону) — среднесуточные расходы воды; данные о температуре и солености воды в Азовском море и Таганрогском заливе, полученные из базы данных Института океанологии РАН и региональных гидрометеорологических наблюдений, а также данные, полученные авторским коллективом в ходе экспедиционных исследований акватории Таганрогского залива Азовского моря.

В качестве источника загрязнения рассматривался глубоководный выпуск очистных сооружений канализации в районе населенного пункта (н.п.) Дмитриадовка Неклиновского района Ростовской области. Параметры точечного источника определены на основе анализа проектной документации и данных экологического мониторинга: труба диаметром 1400 мм, глубина расположения выпуска 8 м, расчетный расход сточных вод 0.5 м^3/с, концентрация фосфатов на выпуске варьируется от 0.05 до 0.10 мг/дм^3 (по данным натурных замеров за 2015–2021 гг., табл. 1), температура сточных вод 18–22°C. Для верификации модели использовались результаты натурных измерений концентраций фосфатов, проведенных в Таганрогском заливе в период с 2015 по 2021 год (табл. 1). О распределении фосфатов в акватории водоема можно косвенно судить по данным спутникового зондирования исследуемого района на предмет прозрачности воды (рис. 1).

Таблица 1. Концентрация фосфатов в Таганрогском заливе по данным натурных измерений

	2015 г.	**2017 г.**	**2020 г.**	**2021 г.**
Среднегодовая концентрация, мг/дм^3	0.032	0.0175	0.0146	0.0117
Максимальные величины, мг/дм^3	0.097	0.0717	0.0815	0.0504

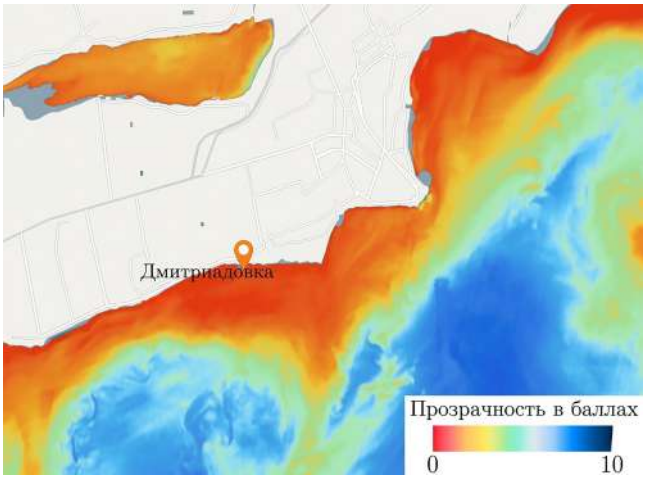


Рис. 1. Прозрачность воды в Азовском море по спутниковым данным от 05.10.2020. Область пониженной прозрачности в Таганрогском заливе указывает на наличие загрязняющих веществ

На рис. 1 можно наблюдать пониженную прозрачность воды в Азовском море (данные приведены для 05.10.2020), что свидетельствует о повышенной концентрации загрязняющих веществ в этом районе. Повышение содержания микроорганизмов в воде свидетельствует о возросшей концентрации биогенных веществ, к которым относятся фосфаты и соединения

азота [8]. Увеличенный и восстановленный фрагмент данного изображения используется в работе в целях проверки достоверности полученных результатов численных экспериментов.

1.2. Ассимиляция спутниковых данных

Для инициализации полей концентрации загрязняющих веществ и верификации результатов моделирования использовалась процедура ассимиляции спутниковых данных, основанная на преобразовании спектральных индексов в концентрации загрязняющих биогенных веществ. Основные этапы включают расчет спектральных индексов, калибровку и объективный анализ.

По снимкам Sentinel-2 рассчитывался индекс NDCI (Normalized Difference Chlorophyll Index):

$$\mathrm{NDCI} = \frac{B_{705} - B_{665}}{B_{705} + B_{665}},$$

где B_{705} и B_{665} — спектральная яркость в каналах 705 нм и 665 нм соответственно. Индекс NDCI коррелирует с концентрацией хлорофилла-a, который является индикатором содержания биогенных веществ (фосфатов, азота) в воде [16]. Для преобразования значений NDCI в концентрацию фосфатов использовалась эмпирическая зависимость, полученная на основе сопоставления спутниковых данных с результатами натурных измерений в Таганрогском заливе за 2020 год. Максимальное значение NDCI в районе выпуска соответствовало концентрации 0.08 мг/дм^3 (максимальная зафиксированная концентрация in-situ).

Построенное поле концентраций фосфатов использовалось в качестве начального условия для численного моделирования. Для верификации модели на последующих временных шагах также использовались обработанные спутниковые снимки, позволяющие отслеживать пространственное распределение загрязняющего вещества. Пример обработки спутникового снимка Sentinel-2 (расчет NDCI) представлен на рис. 2. Желтой линией обозначено расположение глубоководного выпуска в Таганрогском заливе, район н.п. Дмитриадовка.

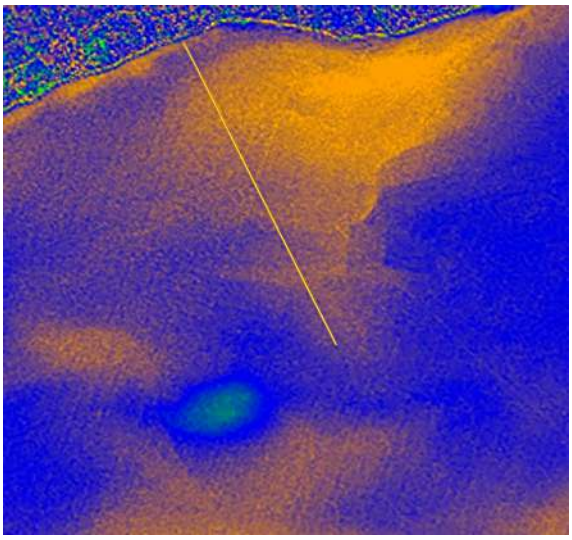


Рис. 2. Окно сервиса Google Earth Engine с результатами расчета индекса NDCI по спутниковому снимку Sentinel-2 от 26.07.2020

2. Математическая модель

2.1. Постановка задачи

Для описания транспорта многокомпонентных загрязняющих веществ в водной среде используется система уравнений диффузии-конвекции-реакции, учитывающая физико-химические и биологические процессы. Уравнение для i-й субстанции имеет следующий вид:

$$l\frac{\partial S_i}{\partial t} + u\frac{\partial S_i}{\partial x} + v\frac{\partial S_i}{\partial y} + (w + w_{gi})\frac{\partial S_i}{\partial z} = \frac{\partial}{\partial x}\left(\mu_i\frac{\partial S_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_i\frac{\partial S_i}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_i\frac{\partial S_i}{\partial z}\right) - \\ - (k_i(T) + d_i)S_i + \psi_i(x, y, z, t), \tag{1}$$

где S_i — концентрация i-й примеси, $i = \overline{1,6}$; 1 — общий органический азот (N); 2 — фосфаты (PO_4); 3 — фитопланктон; 4 — зоопланктон; 5 — растворенный кислород (O_2); 6 — сероводород (H_2S); x, y, z — декартовы координаты; u, v, w — составляющие вектора скорости водного потока; w_{gi} — скорость осаждения частиц; μ_i, ν_i — коэффициенты турбулентного обмена соответственно по горизонтальному и вертикальному направлениям; $k_i(T)$ — коэффициент растворимости для поллютантов, убыли — для кислорода и сероводорода, смертности — для гидробионтов; T, C — температура и соленость водной среды; d_i — коэффициент: уменьшения ЗВ за счет потребления синезелеными водорослями (цианопрокаристами); уменьшения за счет дыхания (для кислорода) и химических реакций (для кислорода и углекислого газа); выедания — для гидробионтов представителями высших трофических уровней; ψ_i — химико-биологический источник (сток).

Расчетная область G представляет собой замкнутый бассейн, ограниченный невозмущенной поверхностью моря $z = 0$, дном $H_0 = H_0(x, y)$ — глубина до твердой поверхности водоема. Пусть Γ — граница области G, $\bar{G} = G \cup \Gamma$; σ — боковая граничная поверхность; $\sum_p$ — часть свободной поверхности и $\sum_d$ — поверхности дна, $\Gamma = \sigma \cup \sum_p \cup \sum_d$.

Считаем, что в водном объеме G находятся субстанции, которые в точке (x, y, z) и в момент времени t имеют концентрацию $S_i = S_i(x, y, z, t)$, [мг/л], $i = \overline{1,6}$; t — временная переменная, [с].

Зададим начальные условия при $t = 0$

$$S_i(x, y, z, 0) \equiv S_{i0}(x, y, z). \tag{2}$$

Граничные условия на боковой поверхности σ в любой момент времени $\sigma \times (0, T]$ имеют вид:

$$\frac{\partial S_i}{\partial \mathbf{n}} = 0, \quad \text{если } (\mathbf{U_\Gamma}, \mathbf{n}) < 0, \tag{3}$$

$$\frac{\partial S_i}{\partial \mathbf{n}} = -\frac{u_\Gamma}{\mu_i}S_i, \quad \text{если } (\mathbf{U_\Gamma}, \mathbf{n}) \geq 0, \tag{4}$$

где $\mathbf{n}$ — внешняя нормаль к границе области σ, $\mathbf{U_\Gamma}$ — вектор скорости движения жидкости на границе Γ, u_Γ — проекция вектора скорости $\mathbf{U_\Gamma}$ на направление нормали $\mathbf{n}$ на границе области G.

Граничные условия на поверхности воды $\sum_p \times (0 < t \leq T]$ зададим следующим образом:

$$\frac{\partial S_i}{\partial \mathbf{n}} = 0. \tag{5}$$

Граничные условия на дне $\sum_d \times (0 < t \leq T]$ имеют вид:

$$\frac{\partial S_i}{\partial \mathbf{n}} = - \frac{w_{gi}}{\nu_i} S_i. \tag{6}$$

2.2. Численная схема и параметризация

Для численного решения системы уравнений (1) используется конечно-разностная аппроксимация на равномерной сетке с шагами h_x, h_y по пространственным координатам и h_t по времени. Для аппроксимации уравнений предложенной математической модели используется разностная схема с весами. Для решения возникающих систем линейных алгебраических уравнений применяется симметризованный адаптивный попеременно-треугольный итерационный метод [32].

В табл. 2 приведены основные параметры модели, используемые в численных экспериментах. Значения коэффициентов турбулентного обмена μ_i, ν_i определены на основе полуэмпирических формул, учитывающих стратификацию и скорость течения водной среды [33]. Значение газовой постоянной $R = 0.4$ кДж/(моль·К) представляет собой эффективную газовую постоянную, используемую в параметризации уравнения Аррениуса для описания температурной зависимости скоростей биогеохимических реакций трансформации фосфатов и органического азота в водной среде. Данное значение получено эмпирическим путем при калибровке модели по натурным данным для условий Таганрогского залива и отличается от универсальной газовой постоянной ($R = 8.314$ Дж/(моль·К)), так как учитывает специфику водной среды и комплексный характер протекающих процессов [33].

Таблица 2. Параметры математической модели транспорта примесей

Параметр	Обозначение	Значение
Коэффициент турбулентности по горизонтали, м^2/с	μ_i	10^{-1}–10
Коэффициент турбулентности по вертикали, м^2/с	ν_i	10^{-2}–1
Скорость осаждения частиц, м/с	w_{gi}	0.001–0.1
Базовая скорость реакции (фосфаты), 1/с	k_{02}	0.1
Энергия активации, кДж/моль	E_a	1.0
Газовая постоянная, кДж/(моль·К)	R	0.4
Коэффициент поглощения фосфатов фитопланктоном, 1/с	d_2	0.05

Гидродинамические поля (компоненты вектора скорости движения среды — u, v, w, уровень поверхности водной среды) рассчитываются по математической модели [13]. Для учета влияния атмосферных процессов используются собственные экспедиционные исследования, данные реанализа ERA5 (скорость ветра, атмосферное давление), речного стока (р. Дон) и приливных явлений. Соленость и температура воды задаются по данным наблюдений и также влияют на коэффициенты турбулентного обмена и скорости химико-биологических реакций.

3. Численные эксперименты и результаты

3.1. Настройка модели и верификация

Для численной реализации математической модели транспорта загрязняющих веществ (уравнения (1)–(6)) разработан программный модуль на на языке программирования C++.

Программный модуль включает управляющий блок (задание граничных условий, временной цикл), блок ввода начальных данных (параметры уравнений, поля скорости, солености, температуры, начальные распределения концентраций, геометрия расчетной области), блок построения сеточных уравнений (формирование коэффициентов и правых частей в канонической форме), блок решения сеточных уравнений (применение симметризованного адаптивного попеременно-треугольного итерационного метода [32]) и блок вывода результатов (запись полей концентраций в файлы).

В программной реализации использовались следующие основные параметры: шаги по пространственным координатам $h_x = 1$, $h_y = 1$ (условные единицы, соответствующие 40 м в реальной области) и по времени $h_t = 10$ с; размеры расчетной сетки $N_x = 250$, $N_y = 250$ узлов; временной интервал $l_t = 50, 500, 1000, 1500$ с; базовая скорость реакции $k_{01} = 0.1$ с$^{-1}$; коэффициент влияния солености $\alpha_z = 1.5$; температура воды $T = 24°$C; соленость $C = 10$ ‰ (остальные параметры приведены в табл. 2).

Верификация результатов моделирования проводилась путем сравнения с обработанными спутниковыми данными Sentinel-2 за период 24–26 июля 2020 года. Для количественного сравнения использовалась обработка спутниковых снимков в программном комплексе QGIS (рис. 3): значения каждого пикселя в расчетной области нормировались на максимальную концентрацию фосфатов, зафиксированную в натурных замерах на выпуске в июне 2020 года (0.08 мг/дм^3). Нормирование на данные июня выполнено для приведения спутниковых данных к абсолютным значениям концентрации фосфатов, так как в июне были проведены наиболее полные натурные замеры на выпуске. Относительные величины на рис. 3 представляют собой нормированные значения, а не абсолютные концентрации. Числовые значения округлены до второго знака после запятой. Для количественной оценки точности использовался критерий Тейла:

$$\delta = \sqrt{\sum_{k=1}^{n} (S_k - S_{k\ nat})^2 / \sum_{k=1}^{n} S_{k\ nat}^2} \cdot 100\%,$$

где $S_{k\ nat}$ — концентрация, полученная на основе обработанных натурных и спутниковых данных; S_k — концентрация, рассчитанная с помощью предложенной математической модели; n — количество замеров.

Оценка погрешности моделирования составила $\delta \approx 14\%$. По сравнению с результатами, полученными другими авторами для схожих условий моделирования распространения загрязняющих веществ в прибрежных зонах [6, 33, 34], данный показатель свидетельствует о повышении точности примерно на 6%.

3.2. Анализ результатов моделирования

На рис. 4 представлены результаты моделирования распространения фосфатов от точечного источника на временных интервалах 50, 500, 1000 и 1500 с при следующих значениях параметров: компоненты скорости течения $u = -3.5$ м/с, $v = -3.5$ м/с, коэффициент турбулентности $\mu_i = 3$ м^2/с. Цветовая шкала соответствует концентрации. На начальном этапе ($t_l = 50$ с) формируется локальное пятно загрязнения в непосредственной близости от выпуска недостаточно очищенных вод. С течением времени под действием течения и турбулентной диффузии происходит распространение пятна в юго-западном направлении, соответствующем преобладающему течению в данном районе Таганрогского залива. К моменту времени $t_l = 1500$ с (25 мин) площадь загрязненной зоны увеличивается, а мак-

2.89	4.05	5.70	6.76	7.31	8.00	8.44	7.20
4.08	5.92	7.23	8.68	8.85	9.33	9.49	7.19
5.74	7.30	8.94	9.94	10.43	10.18	8.81	6.55
6.63	8.47	10.06	11.07	11.64	10.67	8.26	6.75
7.23	8.78	10.79	12.19	11.90	11.05	8.36	6.68
7.26	8.94	12.04	12.31	12.10	10.56	8.02	6.68
7.84	9.37	10.61	11.40	11.35	9.96	7.65	6.20
8.37	8.22	9.66	11.04	10.95	8.74	7.06	5.49
7.32	7.52	9.30	9.82	9.09	7.46	6.08	4.96
6.39	6.73	7.48	7.91	6.92	6.13	5.70	4.62
5.99	6.05	6.57	6.67	6.22	5.63	5.08	4.47
5.77	6.12	6.30	6.42	5.92	5.58	4.83	4.73

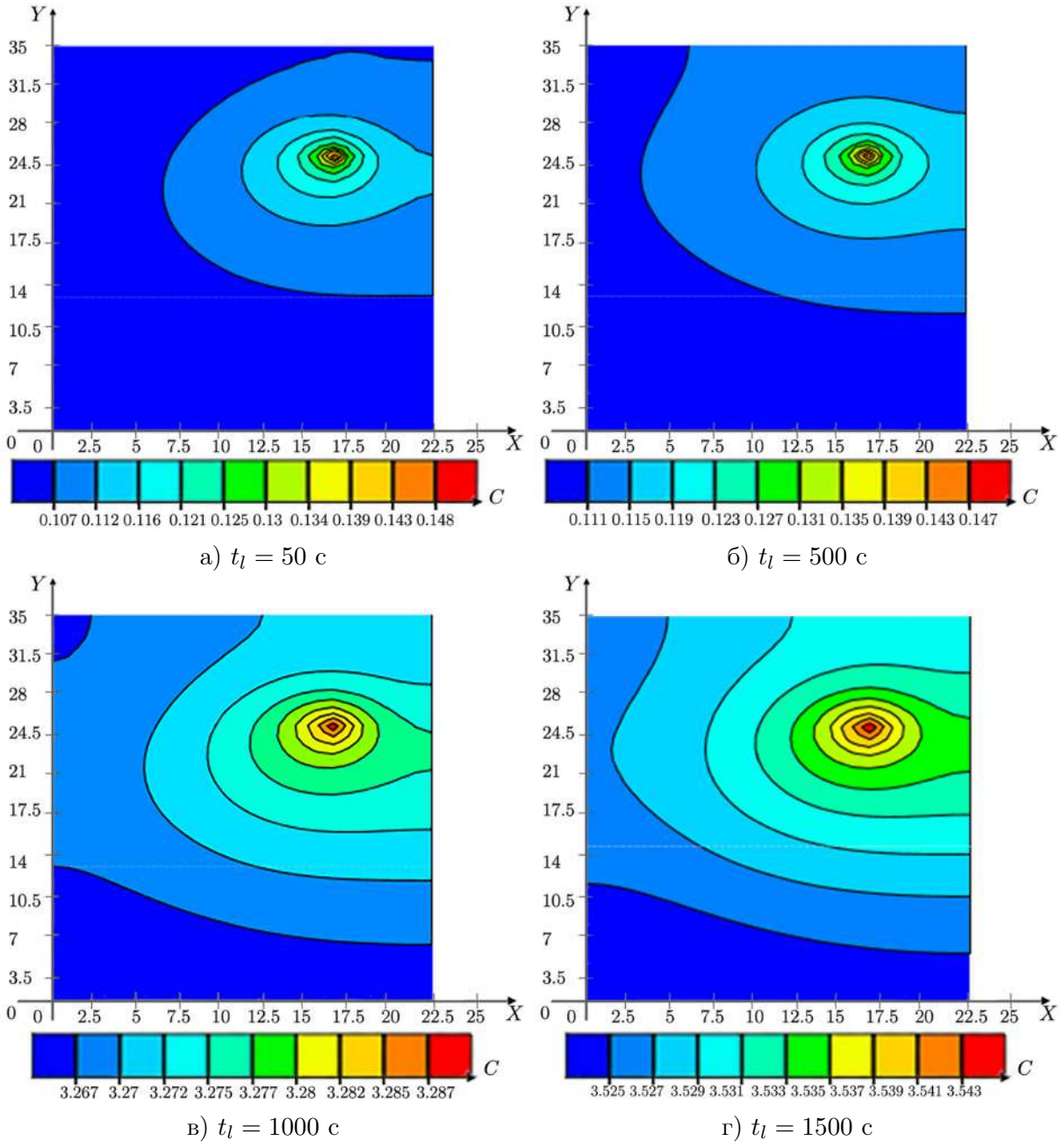
Рис. 3. Нормированные значения концентрации фосфатов в расчетной области, полученные на основе обработки спутниковых снимков Sentinel-2 (NDCI). Цветовая шкала отображает абсолютные значения концентрации

симальная концентрация в центре пятна снижается с 0.08 до 0.04 мг/дм^3 за счет процессов переноса и турбулентного перемешивания.

На рис. 5 представлено сравнение спутниковых данных и результатов моделирования. Качественное соответствие наблюдается между формой и расположением пятна загрязнения, зафиксированного на спутниковых снимках и результатами моделирования через 0.5 ч (рис. 5б) и 1 ч (рис. 5г) после начала сброса. На рис. 5а представлена исходная форма пятна загрязнения (обработанный спутниковый снимок), на рис. 5в представлена конечная форма пятна по спутниковым данным, на рис. 5г отображена конечная форма пятна по результатам расчетов.

Для оценки долгосрочного распространения загрязняющего вещества осуществлено прогнозное моделирование на период до 28 суток. На рис. 6 представлено распределение концентрации загрязняющего вещества S в Азовском море и Таганрогском заливе через 10, 16, 22 и 28 дней после начала сброса. Результаты демонстрируют неравномерное распределение концентрации загрязняющего вещества по акватории залива, обусловленное гидродинамическими процессами и изменением метеорологических условий в период моделирования. Максимальные концентрации наблюдаются вблизи источника сброса, с формированием шлейфа в западном направлении.

На основе разработанного программного модуля и методов обработки спутниковых данных осуществлен мониторинг границ «цветения» вод в районе н.п. Дмитриадовка за май 2021–2024 гг. (рис. 7). Для обработки спутниковых снимков и их анализа использовался программный комплекс [35], включающий модули прогнозирования гидродинамических и гидрофизических процессов. Результаты показывают четкую сезонную цикличность процессов эвтрофикации: максимальные площади цветения воды наблюдаются в мае каждого года, что связано с сочетанием повышенной температуры воды, интенсивной солнечной радиации и продолжающимся поступлением со сточными водами биогенных веществ, вызывающих особенно в весенне-летний период интенсивный рост водорослей. В 2022 и 2023 гг. отмечено увеличение площади «цветения» по сравнению с 2021 г., что может быть связано с изменениями гидрологических условий и антропогенной нагрузкой.



а) $t_l = 50$ с б) $t_l = 500$ с

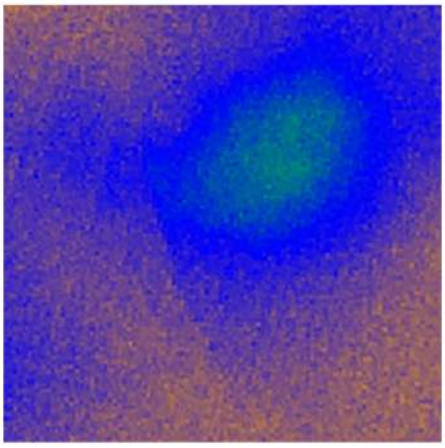
в) $t_l = 1000$ с г) $t_l = 1500$ с

Рис. 4. Распределение концентрации фосфатов (мг/дм^3) в различные моменты времени. По осям X и Y — пространственные координаты (условные единицы)

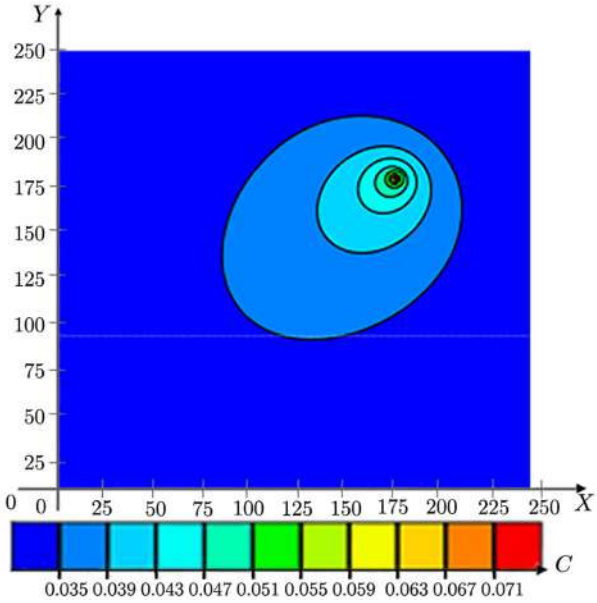
Заключение

Разработана трехмерная нестационарная математическая модель транспорта многокомпонентных примесей в мелководном водоеме, учитывающая гидродинамические и биогеохимические процессы. Модель основана на системе уравнений теории мелкой воды с включением уравнений диффузии-конвекции для шести компонентов: общего органического азота, фосфатов, фитопланктона, зоопланктона, растворенного кислорода и сероводорода.

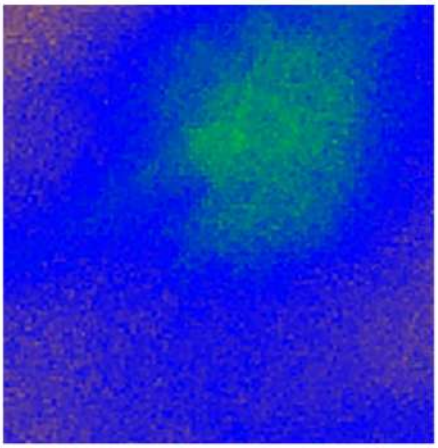
Создан программный модуль на языке C++, реализующий конечно-разностную аппроксимацию уравнений с использованием симметризованного адаптивного попеременно-треугольного итерационного метода. Особенностью разработки является реализация про-



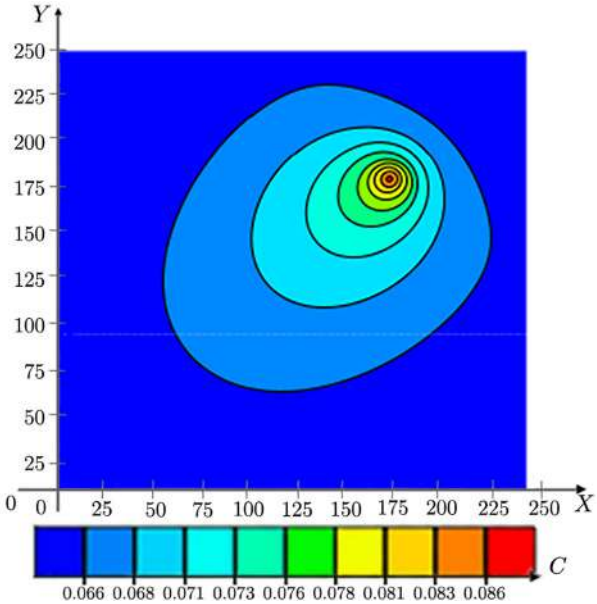
а) Исходная форма пятна загрязнения (спутниковые данные)



б) Результаты моделирования ($t_l = 0.5$ ч)



в) Конечная форма пятна (спутниковые данные)

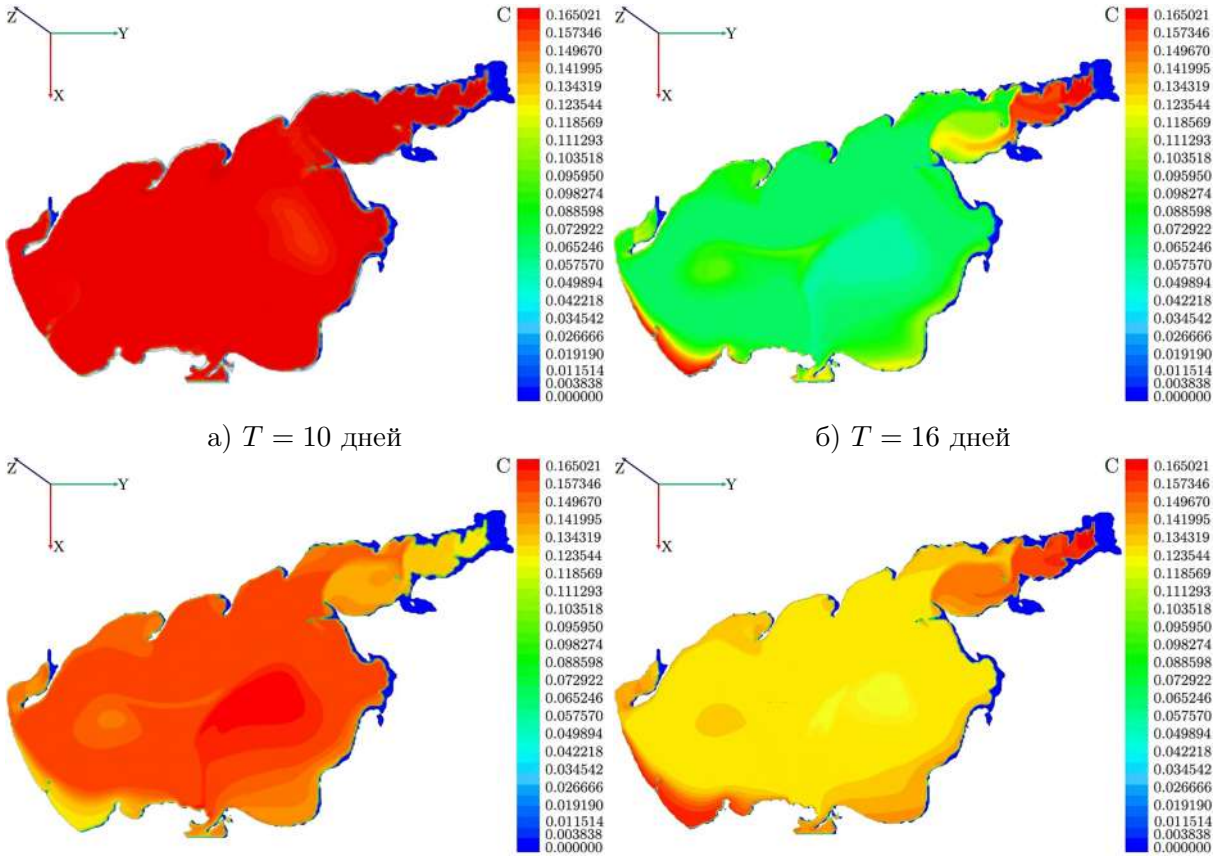


г) Результаты моделирования ($t_l = 1$ ч)

Рис. 5. Верификация математической модели: сравнение спутниковых данных и результатов моделирования распространения фосфатов

цедуры ассимиляции спутниковых данных: преобразование спектрального индекса NDCI, рассчитанного по снимкам Sentinel-2, в поле концентраций фосфатов для инициализации модели и верификации результатов.

Проведены численные эксперименты по моделированию распространения фосфатов от точечного источника в Таганрогском заливе. Верификация модели показала хорошее качественное и количественное соответствия результатов расчетов данным спутникового зондирования. Оценка погрешности по критерию Тейла составила около 14%, что примерно



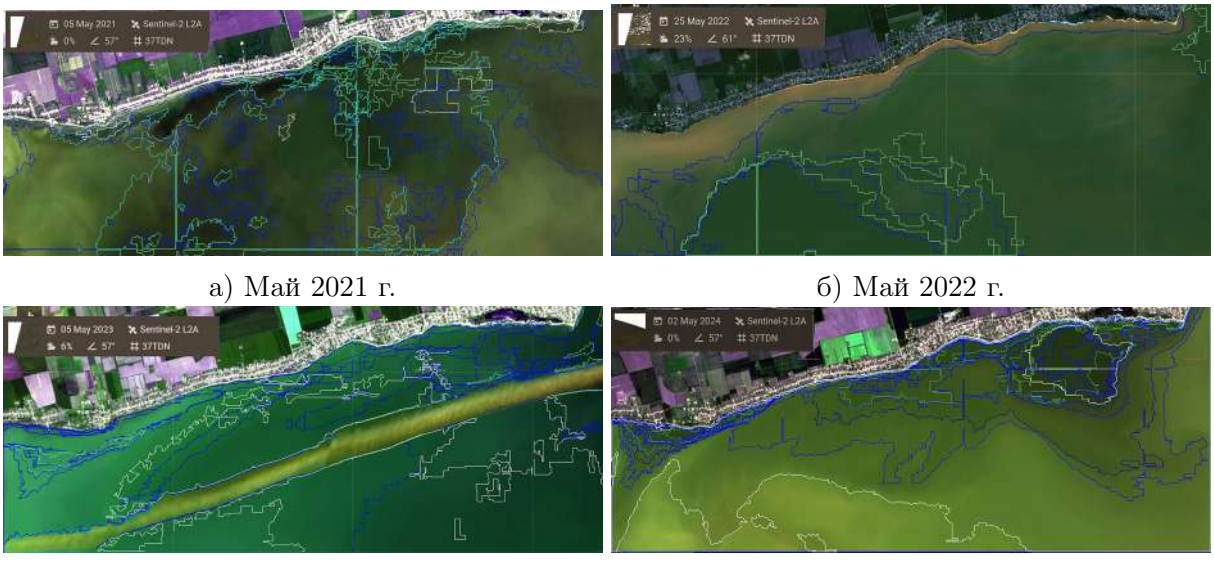
а) $T = 10$ дней

б) $T = 16$ дней

в) $T = 22$ дней

г) $T = 28$ дней

Рис. 6. Распределение концентрации загрязняющего вещества S (мг/дм^3) в Азовском море и Таганрогском заливе через T дней после начала сброса



а) Май 2021 г.

б) Май 2022 г.

в) Май 2023 г.

г) Май 2024 г.

Рис. 7. Границы «цветения» вод в районе н.п. Дмитриадовка (май 2021–2024 гг.), полученные на основе обработки спутниковых данных. Цветом отражена интенсивность индекса NDCI

на 6% точнее по сравнению с результатами, полученными без использования ассимиляции спутниковых данных.

На основе анализа серии спутниковых снимков за 2021–2024 гг. выявлена четкая сезонная динамика границ «цветения» водной среды в районе н.п. Дмитриадовка, связанная с процессами эвтрофикации. Максимальные площади покрытия водорослями наблюдаются в мае–июне каждого года.

Разработанный программный модуль позволяет учитывать специфические для мелководных экосистем биогеохимические процессы и эффективно ассимилировать данные дистанционного зондирования Земли.

Перспективным направлением дальнейших исследований является развитие модели за счет включения более детального описания трофических взаимодействий и учета влияния климатических изменений на динамику эвтрофикации изучаемого гидробиогеценоза.

Исследование выполнено за счет гранта Российского научного фонда № 25-71-20001, https://rscf.ru/project/25-71-20001/.

Литература

1. Кибель И.А. Введение в гидродинамические методы краткосрочного прогноза погоды. М.: Гостехиздат, 1957. 375 с.
2. Марчук Г.И. Численные методы в прогнозе погоды. Л.: Гидрометеоиздат, 1967. 358 с.
3. Монин А.С., Полубаринова-Кочина П.Я., Хлебников В.И. Космология. Гидродинамика. Турбулентность: А.А. Фридман и развитие его научного наследия. М.: Наука, 1989. 328 с.
4. Матишов Г.Г., Шохин И.В., Набоженко М.В., Польшин В.В. Многолетние изменения донных сообществ Азовского моря в связи с характером осадконакопления и гидрологическим режимом // Океанология. 2008. Т. 48, № 3. С. 425–435.
5. Лыкосов В.Н. Современные проблемы математического моделирования климата и прогноза погоды // Известия РАН. Физика атмосферы и океана. 2020. Т. 56, № 3. С. 5–18.
6. Yakushev E., Pogojeva M., Polukhin A., *et al.* Arctic inshore biogeochemical regime influenced by coastal runoff and glacial melting (case study for the Templefjord, Spitsbergen) // Geosciences. 2022. Vol. 12, no. 1. P. 44. DOI: 10.3390/geosciences12010044.
7. Закруткин В.Е., Решетняк О.С., Усова Е.В. О допустимых антропогенных нагрузках на водосборах рек степной зоны Юга России (в пределах Ростовской области) // Степи Северной Евразии: материалы X международного симпозиума (Международного степного форума). Оренбург, 2024. С. 463–468.
8. Чистяков А.Е., Кузнецова И.Ю. Оценка экологических рисков мелководного водоема при проведении дноуглубительных работ // Безопасность техногенных и природных систем. 2024. № 2. С. 37–46. DOI: 10.23947/2541-9129-2024-8-2-37-46.
9. Коваль Л.Г. Зоо и некрозоопланктон Чёрного моря. Киев: Наукова думка, 1984. 127 с.
10. Петипа Т.С., Павлова Е.В. Смертность зоопланктона в Севастопольской бухте // Доповіді НАНУ. 1995. Т. 6. С. 146–148.

11. Гурова Ю.С., Якушев Е.В., Березина А.В. и др. Численное моделирование динамики окислительно-восстановительных условий на границе вода – донные отложения в Южной бухте // Моря России: от теории к практике океанологических исследований: тезисы докладов Всероссийской научной конференции, Севастополь, 25–29 сентября, 2023. Севастополь: ФГБУН ФИЦ МГИ РАН, 2023. С. 246–247. DOI: 10.29039/2413-5577-2023-2-71-90.

12. Пахомова С.В., Ершова А.А., Жданов И.А., Якушев Е.В. Методы исследования загрязнения микропластиком природных вод: современное состояние и рекомендации // Океанологические исследования. 2024. Т. 52, № 1. С. 80–120. DOI: 10.29006/1564-2291.JOR-2024.52(1).5.

13. Сухинов А.И., Чистяков А.Е., Кузнецова И.Ю. и др. Регуляризованная разностная схема для решения задач гидродинамики // Математическое моделирование. 2022. Т. 34, № 2. С. 85–100. DOI: 10.20948/mm-2022-02-07.

14. Лященко Т.В., Чистяков А.Е., Никитина А.В. Моделирование процесса распространения загрязнения водной экосистемы фосфатами // Computational Mathematics and Information Technologies. 2023. Т. 7, № 4. С. 47–53. DOI: 10.23947/2587-8999-2023-7-4-47-53.

15. Сухоруков Б.Л., Решетняк Н.В., Сапрыгин В.В. Информативность мультиспектральных (OLCI) и гиперспектральных данных при оценке состояния пресноводных экосистем на примере Цимлянского водохранилища // Современные проблемы дистанционного зондирования Земли из космоса. 2022. Т. 19, № 1. С. 239–252. DOI: 10.21046/2070-7401-2022-19-1-239-252.

16. Белова Ю.В., Развеева И.Ф., Рахимбаева Е.О. Разработка алгоритма семантической сегментации данных дистанционного зондирования Земли для определения фитопланктонных популяций // Advanced Engineering Research. 2024. Т. 24, № 3. С. 283–292. DOI: 10.23947/2687-1653-2024-24-3-283-292.

17. Белова Ю.В., Никитина А.В. Применение методов усвоения данных наблюдений для моделирования распространения загрязняющих веществ в водоеме и управления устойчивым развитием // Безопасность техногенных и природных систем. 2024. № 3. С. 39–48. DOI: 10.23947/2541-9129-2024-8-3-39-48.

18. Белова Ю.В., Филина А.А., Чистяков А.Е. Прогнозирование динамики летних видов фитопланктона на основе методов усвоения спутниковых данных // Computational Mathematics and Information Technologies. 2024. Т. 8, № 4. С. 27–34. DOI: 10.23947/2587-8999-2024-8-4-27-34.

19. Литвинов В.Н., Руденко Н.Б., Грачева Н.Н. Разработка модели параллельноконвейерного вычислительного процесса для решения системы сеточных уравнений // Advanced Engineering Research. 2023. Т. 23, № 3. С. 329–339. DOI: 10.23947/2687-1653-2023-23-3-329-339.

20. Кудинов Н.В., Филина А.А., Никитина А.В. и др. Моделирование вертикальных движений морской воды в стратифицированных водоемах // Advanced Engineering Research. 2023. Т. 23, № 2. С. 212–224. DOI: 10.23947/2687-1653-2023-23-2-212-224.

21. Горелкина А.К., Яковлев И.В. Математическая модель Гаусса для определения распространения загрязняющих веществ в регионах с повышенной техногенной нагрузкой // Вестник Омского государственного аграрного университета. 2025. № 1(57). С. 156–165.

22. Белихов А.Б., Леготин Д.Л., Сухов А.К. Современные компьютерные модели распространения загрязняющих веществ в атмосфере // Вестник Костромского государственного университета им. Н.А. Некрасова. 2013. Т. 19, № 1. С. 14–19.

23. Сысоева Е.В., Гельманова М.О. Методы расчета рассеивания загрязняющих веществ в городской атмосфере // Вестник МГСУ. 2022. № 8. С. 1027–1045. DOI: 10.22227/1997-0935.2022.8.1027-1045.

24. Сухинов А.И., Кузнецова И.Ю. Математическая модель транспорта трехкомпонентной взвеси // Computational Mathematics and Information Technologies. 2023. Т. 7, № 3. С. 39–48. DOI: 10.23947/2587-8999-2023-7-3-39-48.

25. Katsnelson B.G., Petnikov V.G., Lynch J. Fundamentals of shallow water acoustics. Springer Science & Business Media, 2012. 540 p. DOI: 10.1007/978-1-4419-9777-7.

26. Васильева Е.С., Белякова П.А., Алексюк А.И. и др. Моделирование быстроразвивающихся паводков на малых реках Северного Кавказа с использованием современных данных автоматизированной гидрометеорологической сети // Водные ресурсы. 2021. Т. 48, № 2. С. 135–146. DOI: 10.31857/S0321059621020164.

27. Лукачёв С.В., Бирюк В.В., Горшкалёв А.А. Использование Ansys Fluent для исследования газодинамических и тепловых процессов в малоразмерном двухтактном ДВС // Машиностроение и компьютерные технологии. 2014. № 12. С. 128–135.

28. Дайлюденко В.Ф. Применение программного пакета STARCD для моделирования двухфазного направленного потока в ограниченной области при различных режимах взаимодействия частиц // Информатика. 2010. № 1(25). С. 91–101.

29. Аксенов А.А. FlowVision: индустриальная вычислительная гидродинамика // Компьютерные исследования и моделирование. 2017. Т. 9, № 1. С. 520.

30. Ефимов М.Ю. Разработка и реализация основ компилятора C# с помощью Microsoft Phoenix // Информатика, телекоммуникации и управление. 2010. № 4(103). С. 164–169.

31. Earth Observing System LandViewer Pricing. URL: http://eos.com/landviewer/account/pricing (дата обращения: 11.08.2025).

32. Сухинов А.И., Чистяков А.Е., Проценко С.В., Сидорякина В.В. Решение сеточных уравнений симметризованным адаптивным попеременно-треугольным итерационным методом. 2022. Заявка № 2022680018 от 27.10.2022. Свидетельство о регистрации программы для ЭВМ RU 2022681029.

33. Чикин А.Л. Двухслойная математическая модель ветровых течений в водоемах, имеющих большие площади мелководья // Математическое моделирование. 2009. Т. 21, № 12. С. 152–160.

34. Бердников С.В., Селютин В.В., Сурков Ф.А., Тютюнов Ю.В. Моделирование морских экосистем: опыт, современные подходы, направления развития (обзор). Часть 1. Сквозные модели // Морской гидрофизический журнал. 2022. Т. 38, № 1(223). С. 105–122. DOI: 10.22449/0233-7584-2022-1-105-122.

35. Панасенко Н.Д. Прогноз состояния прибрежных систем с помощью математического моделирования на основе космических снимков // Computational Mathematics and Information Technologies. 2023. Т. 7, № 4. С. 54–65. DOI: 10.23947/2587-8999-2023-7-4-54-65.

Сухинов Александр Иванович, член-корреспондент РАН, доктор физико-математических наук, профессор, заведующий кафедрой «Математика и информатика», Донской государственный технический университет (Ростов-на-Дону, Российская Федерация)

Панасенко Наталья Дмитриевна, кандидат технических наук, доцент кафедры «Математика и информатика», доцент кафедры «Информационная безопасность в вычислительных системах и сетях», Донской государственный технический университет (Ростов-на-Дону, Российская Федерация)

Чистяков Александр Евгеньевич, доктор физико-математических наук, профессор кафедры «Программное обеспечение вычислительной техники и автоматизированных систем», Донской государственный технический университет (Ростов-на-Дону, Российская Федерация)

Никитина Алла Валерьевна, доктор технических наук, доцент, профессор кафедры «Программное обеспечение вычислительной техники и автоматизированных систем», Донской государственный технический университет (Ростов-на-Дону, Российская Федерация)

DOI: 10.14529/cmse260102

USING SATELLITE INDICATORS OF PHYTOPLANKTON BIOMASS IN MATHEMATICAL MODELING OF PHOSPHATE SPREADING IN TAGANROG BAY

© 2026 A.I. Sukhinov, N.D. Panasenko, A.E. Chistyakov, A.V. Nikitina

Don State Technical University

(pl. Gagarina 1, Rostov-on-Don, 344003 Russia)

E-mail: sukhinov@gmail.com, natalija93_93@mail.ru, cheese_05@mail.ru, nikitina.vm@gmail.com

Received: 05.10.2025

A non-stationary three-dimensional mathematical model for the transport of multicomponent impurities in a shallow water body is presented, based on a system of shallow water equations that account for biogeochemical processes. The model allows for the calculation of the spatio-temporal distribution of concentrations of biogenic substances (phosphates, nitrogen compounds), phytoplankton and zooplankton, dissolved oxygen, and hydrogen sulfide. A distinctive feature of the approach is the assimilation of Earth remote sensing data (Sentinel-2, Landsat satellites) through the conversion of the spectral NDCI index into a concentration field for model initialization and verification. A software module in C++ has been developed, implementing a finite-difference approximation of the equations of the proposed mathematical model using an alternating triangular iterative method. Using the example of the Taganrog Bay of the Azov Sea, a series of numerical experiments were conducted to model the spread of phosphates from a point source (deep-water discharge of treatment facilities near the settlement of Dmitriadovka). Model verification was performed by comparing the calculation results with processed satellite data and in-situ measurements. The error estimate according to Theil's criterion was 14%. Based on the analysis of satellite images for the period 2021–2024, seasonal dynamics of the boundaries of "algal bloom" due to eutrophication were revealed. The proposed approach demonstrates an improvement in forecast accuracy compared to models that do not use the assimilation of satellite data.

Keywords: numerical simulation, phosphates, Taganrog Bay, data assimilation, remote sensing, NDCI index, mathematical model, software module.

FOR CITATION

Sukhinov A.I., Panasenko N.D., Chistyakov A.E., Nikitina A.V. Using Satellite Indicators of Phytoplankton Biomass in Mathematical Modeling of Phosphate Spreading in Taganrog Bay. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2026. Vol. 15, no. 1. P. 38–56. (in Russian) DOI: 10.14529/cmse260102.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Kibel I. Introduction to Hydrodynamic Methods of Short-Term Weather Forecasting. Moscow: Gostekhizdat, 1957. 375 p. (in Russian).
2. Marchuk G. Numerical Methods in Weather Forecasting. Leningrad: Gidrometeoizdat, 1967. 358 p. (in Russian).
3. Monin A., Polubarinova-Kochina P., Khlebnikov V. Cosmology. Hydrodynamics. Turbulence: A.A. Friedman and the Development of His Scientific Heritage. Moscow: Nauka, 1989. 328 p. (in Russian).
4. Matishov G., Shokhin I., Nabozhenko M., Pol'shin V. Long-Term Changes in the Bottom Communities of the Azov Sea in Connection with the Nature of Sedimentation and Hydrological Regime. Oceanology. 2008. Vol. 48, no. 3. P. 425–435. (in Russian).
5. Lykosov V. Modern Problems of Mathematical Modeling of Climate and Weather Forecasting. Izvestiya RAN. Fizika atmosfery i okeana. 2020. Vol. 56, no. 3. P. 5–18. (in Russian).
6. Yakushev E., Pogojeva M., Polukhin A., *et al.* Arctic inshore biogeochemical regime influenced by coastal runoff and glacial melting (case study for the Templefjord, Spitsbergen). Geosciences. 2022. Vol. 12, no. 1. P. 44. DOI: 10.3390/geosciences12010044.
7. Zakrutkin V., Reshetnyak O., Usova E. On Permissible Anthropogenic Loads on the Catchments of Rivers in the Steppe Zone of the South of Russia (within the Rostov Region). Steppes of Northern Eurasia: Proceedings of the X International Symposium (International Steppe Forum). Orenburg, 2024. P. 463–468. (in Russian).
8. Chistyakov A., Kuznetsova I. Assessment of Environmental Risks of a Shallow Water Body during Dredging Works. Safety of Technogenic and Natural Systems. 2024. No. 2. P. 37–46. (in Russian) DOI: 10.23947/2541-9129-2024-8-2-37-46.
9. Koval L. Zoo and Necrozooplankton of the Black Sea. Kyiv: Naukova Dumka, 1984. 127 p. (in Russian).
10. Petipa T., Pavlova E. Zooplankton Mortality in Sevastopol Bay. Dopovidi NASU. 1995. Vol. 6. P. 146–148. (in Russian).
11. Gurova Y., Yakushev E., Berezina A., *et al.* Numerical Modeling of the Dynamics of Oxidation-Reduction Conditions at the Water-Bottom Sediment Interface in Yuzhnaya Bay. Seas of Russia: from theory to practice of oceanographic research: Abstracts of reports of the All-Russian scientific conference, Sevastopol, September 25–29, 2023. Sevastopol: FSBSI FRC MHI RAS, 2023. P. 246–247. (in Russian) DOI: 10.29039/2413-5577-2023-2-71-90.

12. Pakhomova S., Ershova A., Zhdanov I., Yakushev E. Methods for Studying Microplastic Pollution of Natural Waters: Current Status and Recommendations. Oceanological Research. 2024. Vol. 52, no. 1. P. 80–120. (in Russian) DOI: 10.29006/1564-2291.JOR-2024.52(1).5.

13. Sukhinov A., Chistyakov A., Kuznetsova I., *et al.* Regularized Difference Scheme for Solving Hydrodynamic Problems. Mathematical Models and Computer Simulations. 2022. Vol. 34, no. 2. P. 85–100. (in Russian) DOI: 10.20948/mm-2022-02-07.

14. Lyashchenko T., Chistyakov A., Nikitina A. Pollutions Spreading Process Modelling in an Aquatic Ecosystem. Computational Mathematics and Information Technologies. 2023. Vol. 7, no. 4. P. 47–53. (in Russian) DOI: 10.23947/2587-8999-2023-7-4-47-53.

15. Sukhorukov B., Reshetnyak N., Saprygin V. Information Content of Multispectral (OLCI) and Hyperspectral Data in Assessing the State of Freshwater Ecosystems Using the Tsimlyansk Reservoir as an Example. Modern Problems of Remote Sensing of the Earth from Space. 2022. Vol. 19, no. 1. P. 239–252. (in Russian) DOI: 10.21046/2070-7401-2022-19-1-239-252.

16. Belova Y., Razveeva I., Rakhimbaeva E. Development of an Algorithm for Semantic Segmentation of Earth Remote Sensing Data to Determine Phytoplankton Populations. Advanced Engineering Research. 2024. Vol. 24, no. 3. P. 283–292. (in Russian) DOI: 10.23947/2687-1653-2024-24-3-283-292.

17. Belova Y., Nikitina A. Application of Methods of Observational Data Assimilation to Model the Spread of Pollutants in a Reservoir and Manage Sustainable Development. Safety of Technogenic and Natural Systems. 2024. No. 3. P. 39–48. (in Russian) DOI: 10.23947/2541-9129-2024-8-3-39-48.

18. Belova Y., Filina A., Chistyakov A. Forecasting the Dynamics of Summer Phytoplankton Species Based on Satellite Data Assimilation Methods. Computational Mathematics and Information Technologies. 2024. Vol. 8, no. 4. P. 27–34. (in Russian) DOI: 10.23947/2587-8999-2024-8-4-27-34.

19. Litvinov V., Rudenko N., Gracheva N. Model of a Parallel-Pipeline Computational Process for Solving a System of Grid Equations. Advanced Engineering Research. 2023. Vol. 23, no. 3. P. 329–339. (in Russian) DOI: 10.23947/2687-1653-2023-23-3-329-339.

20. Kudinov N., Filina A., Nikitina A., *et al.* Simulation of Vertical Movements of Seawater in Stratified Reservoirs. Advanced Engineering Research. 2023. Vol. 23, no. 2. P. 212–224. (in Russian) DOI: 10.23947/2687-1653-2023-23-2-212-224.

21. Gorelkina A., Yakovlev I. Mathematical Gaussian Model for Determining the Spread of Pollutants in Regions with Increased Technogenic Load. Bulletin of Omsk State Agrarian University. 2025. No. 1(57). P. 156–165. (in Russian).

22. Belikhov A., Legotin D., Sukhov A. Modern Computer Models of Pollutant Dispersion in the Atmosphere. Bulletin of the Kostroma State University named after N.A. Nekrasov. 2013. Vol. 19, no. 1. P. 14–19. (in Russian).

23. Sysoeva E., Gel'manova M. Methods for Calculating the Dispersion of Pollutants in the Urban Atmosphere. Bulletin of Moscow State University of Civil Engineering. 2022. No. 8. P. 1027–1045. (in Russian) DOI: 10.22227/1997-0935.2022.8.1027-1045.

24. Sukhinov A., Kuznetsova I. Mathematical Model of Three-Component Suspension Transport. Computational Mathematics and Information Technologies. 2023. Vol. 7, no. 3. P. 39–48. (in Russian) DOI: 10.23947/2587-8999-2023-7-3-39-48.

25. Katsnelson B., Petnikov V., Lynch J. Fundamentals of Shallow Water Acoustics. Springer Science & Business Media, 2012. 540 p. DOI: 10.1007/978-1-4419-9777-7.

26. Vasil'eva E., Belyakova P., Aleksyuk A., *et al.* Simulating Flash Floods in Small Rivers of the Northern Caucasus with the Use of Data of Automated Hydrometeorological Network. Water Resources. 2021. Vol. 48, no. 2. P. 182–193. (in Russian) DOI: 10.1134/S0097807821020160.

27. Lukachev S., Biryuk V., Gorshkalov A. Using Ansys Fluent to Study Gas-Dynamic and Thermal Processes in a Small-Sized Two-Stroke Internal Combustion Engine. Mechanical Engineering and Computer Technologies. 2014. No. 12. P. 128–135. (in Russian).

28. Daylyudenko V. Application of the STARCD Software Package for Modeling a Two-Phase Directed Flow in a Limited Region under Various Particle Interaction Regimes. Computer Science. 2010. No. 1(25). P. 91–101. (in Russian).

29. Aksenov A. FlowVision: Industrial Computational Fluid Dynamics. Computer Research and Modeling. 2017. Vol. 9, no. 1. P. 520. (in Russian).

30. Efimov M. Development and Implementation of the Basics of a C# Compiler Using Microsoft Phoenix. Computer Science, Telecommunications and Management. 2010. No. 4(103). P. 164–169. (in Russian).

31. Earth Observing System LandViewer Pricing. URL: http://eos.com/landviewer/account/pricing (accessed: 11.08.2025).

32. Sukhinov A., Chistyakov A., Protsenko S., Sidoryakina V. Solution of Grid Equations by Symmetrized Adaptive Alternating-Triangular Iterative Method. 2022. Application No. 2022680018 dated 27.10.2022 (in Russian). Certificate of registration of computer program RU 2022681029.

33. Chikin A. The Two-Layer Mathematical Model of the Wind Currents in the Basin with Different Level of Deep. Mathematical Modeling. 2009. Vol. 21, no. 12. P. 152–160. (in Russian).

34. Berdnikov S., Selyutin V., Surkov F., Tyutyunov Y. Modeling of Marine Ecosystems: Experience, Modern Approaches, Directions of Development (Review). Part 1: End-To-End Models. Physical Oceanography. 2022. Vol. 29, no. 1. P. 98–114. (in Russian) DOI: 10.22449/0233-7584-2022-1-105-122.

35. Panasenko N. Forecasting the State of Coastal Systems Using Mathematical Modeling Based on Satellite Images. Computational Mathematics and Information Technologies. 2023. Vol. 7, no. 4. P. 54–65. (in Russian) DOI: 10.23947/2587-8999-2023-7-4-54-65.

УДК 519.7 **DOI: 10.14529/cmse260103**

ПРИЧИННО-СЛЕДСТВЕННАЯ СЕМАНТИКА ДЛЯ НЕПРЕРЫВНО-ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ СО СЛАБОЙ ВРЕМЕННОЙ СТРАТЕГИЕЙ

© 2026 А.Ю. Зубарев

Институт систем информатики им. А.П. Ершова СО РАН

(630090 Новосибирск, пр. Академика Лаврентьева, д. 6)

E-mail: auzubarev@gmail.com

Поступила в редакцию: 14.12.2025

Непрерывно-временные сети Петри (НВСП) являются расширением классических сетей Петри, предназначенным для моделирования и анализа параллельных и распределенных систем с учетом временных характеристик. В НВСП переходы, моделирующие события системы, имеют локальные часы и временные интервалы срабатывания. Для данной модели рассматриваются две стратегии хода времени: сильная, при которой ход времени блокируется необходимостью срабатывания перехода, и слабая, не ограничивающая ход модельного времени. Эти стратегии несравнимы по выразительности, причем многие стандартные задачи анализа разрешимы лишь для слабой стратегии. В общем случае пространство состояний НВСП бесконечно и несчетно, что усложняет анализ поведения модели. Для редукции пространства рассматриваемых состояний применяется причинно-следственная семантика, описывающая поведение системы через отношение частичного порядка на множестве событий. Данный подход позволяет абстрагироваться от избыточного множества всех возможных чередований событий (интерливинга), а также сохраняет информацию о причинно-следственных связях и параллелизме между событиями системы. В статье определяется и исследуется модель временных процессов, представляющая причинно-следственную семантику для НВСП в контексте слабой временной стратегии. Основным результатом работы является установление взаимооднозначного соответствия между пробегами НВСП (представляющими классическую интерливинговую семантику) и линеаризациями временных процессов, что доказывает корректность предложенной причинно-следственной семантики. Представленные в работе вычислительные эксперименты подтверждают значительную редукцию пространства анализируемых состояний НВСП при использовании временных процессов.

Ключевые слова: непрерывно-временная сеть Петри, слабая стратегия хода времени, временной процесс, причинно-следственная семантика.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Зубарев А.Ю. Причинно-следственная семантика для непрерывно-временных сетей Петри со слабой временной стратегией // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2026. Т. 15, № 1. С. 57–79. DOI: 10.14529/cmse260103.

Введение

Сеть Петри — это формализм для моделирования и анализа параллельных и распределенных систем, которые являются критически важными с точки зрения безопасности. Эта модель представлена в виде ориентированного двудольного графа с вершинами из двух множеств: *мест* (условия системы) и *переходов* (события системы). Данная структура наглядно описывает причинную зависимость, параллелизм и конфликт между компонентами системы. Для учета количественных характеристик системы были предложены различные временные расширения сетей Петри. Среди наиболее известных — *непрерывно-временные сети Петри* (НВСП). В то время как в некоторых подходах время описывается дискретно с помощью специальных событий-тактов [1], в НВСП каждому переходу сопоставлен *временной интервал* и вещественнозначные *локальные часы*.

Состояние НВСП определяется *разметкой* — множеством мест с *фишками* — и значениями локальных часов для *допустимых переходов* (имеющих фишки во всех входных местах). Переход *может сработать*, если он допустимый и значение его локальных часов лежит в пределах его временного интервала. Изменения состояния происходят двумя способами: посредством *срабатывания перехода* или в результате *хода времени*. Когда переход срабатывает, разметка обновляется: фишки удаляются из его входных мест и добавляются в его выходные места. Затем локальные часы вновь допустимых переходов сбрасываются в ноль. При ходе времени показания локальных часов всех допустимых переходов равномерно увеличиваются. Таким образом, пространство состояний в общем случае бесконечно и несчетно, что усложняет анализ данной модели.

Для НВСП в литературе рассматриваются две стратегии хода времени: *сильная* и *слабая*. При сильной стратегии допустимый переход должен сработать не позднее, чем его локальные часы достигнут верхней границы его временного интервала. Напротив, при слабой стратегии время не ограничено, и переход не обязан срабатывать. Эти две стратегии несравнимы относительно слабой временной бисимуляции [2]. Для целей оценки производительности систем предлагаются гибридные варианты, такие как «слабо-сильная» стратегия [3]. Многие стандартные задачи разрешимы только при слабой стратегии [4, 5]. Кроме того, недавние исследования показывают, что разрешимость сохраняется даже при расширении модели дополнительными типами дуг, в то время как наличие даже одного перехода с сильной стратегией приводит к неразрешимости [6]. В то время как сильная стратегия является классической и хорошо изучена (см., например, [7, 8]), слабая появилась относительно недавно. В данной работе исследуется слабая стратегия.

Последовательности изменений состояний, вызванные срабатываниями переходов и ходом времени, известные как *пробеги*, определяют *интерливинговую семантику* НВСП. Хотя пробеги могут быть использованы для анализа некоторых свойств системы, они теряют всю информацию о параллелизме переходов. В синхронных моделях эта проблема решается за счет тактовой структуры [9], однако для асинхронных систем предпочтительным подходом является использование причинно-следственных семантик, основанных на отношении частичного порядка. Эти семантики предлагают значительное сокращение пространства рассматриваемых состояний, поскольку позволяют избежать необходимости рассмотрения всех возможных интерливинговых последовательностей. Однако наличие циклов в сетях Петри препятствует прямому определению частичного порядка. Поэтому используется ациклическая *причинно-следственная сеть* и функция (*гомоморфизм*), отображающая элементы этой сети (условия и события) в соответствующие элементы сети Петри (места и переходы). *Причинно-следственный процесс* сети Петри определяется как пара, состоящая из причинно-следственной сети и гомоморфизма. Аналогично, для временных расширений сетей Петри определяются *временные причинно-следственные сети* и соответствующие *временные процессы*. Модель временных процессов для НВСП с сильной стратегией, где временные конструкции сопоставляются с событиями причинно-следственной сети, была введена в [10]. В данной статье обобщается подход из [10] на слабую стратегию.

Статья организована следующим образом. В разделе 1 рассматриваются синтаксис и интерливинговая семантика классических сетей Петри и их расширения до НВСП со слабой стратегией хода времени. В разделе 2 вводится модель временного процесса и его линеаризаций (интерливинговые выполнения). В разделе 3 устанавливается биективное соответствие между данными линеаризациями и пробегами. В разделе 4 приводятся вычислительные экс-

перименты по количественной оценке редукции пространства состояний НВСП при переходе от анализа пробегов к анализу временных процессов. Статья завершается заключением с результатами и направлениями для будущей работы.

1. Непрерывно-временная сеть Петри

В данном разделе вводится модель сети Петри с временными интервалами, представляющими задержки срабатываний переходов. Сначала следует напомнить синтаксис сетей Петри.

Определение 1. (Сеть Петри)

- *(Помеченная на множестве действий Act) сеть Петри* — это набор $\mathcal{N} = (P,\ T,\ F,\ M_0,\ L)$, где P — конечное *множество мест*, а T — конечное множество переходов такое, что $P \cap T = \emptyset$ и $P \cup T \neq \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ — *отношение инцидентности*, $\emptyset \neq M_0 \subseteq P$ — *начальная разметка*, $L : T \to Act$ — *помечающая функция*. Для элемента $x \in P \cup T$ вводится множество ${}^\bullet x = \{y \mid (y, x) \in F\}$ *входных* и множество $x^\bullet = \{y \mid (x, y) \in F\}$ *выходных элементов*. Для $X \subseteq P \cup T$ принимается, что ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ и $X^\bullet = \bigcup_{x \in X} x^\bullet$.
- *Разметка* M сети $\mathcal{N}$ — это любое подмножество множества P. Переход $t \in T$ *допустимый в разметке* M, если ${}^\bullet t \subseteq M$. Пусть $En(M)$ обозначает множество всех переходов, допустимых в M. *Срабатывание* перехода t, допустимого в разметке M, приводит к новой разметке $M' = (M \backslash {}^\bullet t) \cup t^\bullet$ (обозначается $M \xrightarrow{t} M'$).
- Разметка M является *достижимой разметкой* сети $\mathcal{N}$, если существует последовательность переходов $t_1 \dots t_n$ (для $n \geqslant 0$) такая, что $M_0 \xrightarrow{t_1} M_1 \dots M_{n-1} \xrightarrow{t_n} M_n = M$. Сеть Петри $\mathcal{N}$ называется *бесконтактной*, если для каждой достижимой разметки M и каждого перехода $t \in En(M)$ выполняется $(M \backslash {}^\bullet t) \cap t^\bullet = \emptyset$.

Непрерывно-временная сеть Петри (НВСП) состоит из *базовой* сети Петри и *статической временной функции*, которая назначает временной интервал каждому переходу. Предполагается, что каждый переход имеет свои локальные часы, которые начинают отсчет времени с момента, когда переход стал допустимым.

Пусть $Interv = \{[a, b], [a, b) \mid a \in \mathbb{Q},\ b \in (\mathbb{Q} \cup \{\infty\}),\ a \leqslant b\}$ — множество интервалов с границами из множества рациональных чисел.

Определение 2. (Непрерывно-временная сеть Петри)

- *(Помеченная на Act) непрерывно-временная сеть Петри со слабой стратегией хода времени* ($\text{НВСП}_{\text{сл}}$) — это пара $\mathcal{TN} = (\mathcal{N},\ D)$, где $\mathcal{N} = (P,\ T,\ F,\ M_0,\ L)$ — *базовая* (помеченная на *Act*) сеть Петри и $D : T \to Interv$ — *статическая временная функция*.
- *Состояние* $\mathcal{TN}$ — это тройка (M, I, GT), где M — разметка, $I :\ En(M) \to \mathbb{R}_{\geqslant 0}$ — *динамическая временная функция*, а $GT \in \mathbb{R}_{\geqslant 0}$ — *глобальный момент времени*. Начальное состояние $\mathcal{TN}$ — это $S_0 = (M_0, I_0, GT_0)$, где M_0 — начальная разметка, $I_0(t) = 0$ для всех $t \in En(M_0)$ и $GT_0 = 0$.

Переход t, допустимый в разметке M, *может сработать из состояния* $S = (M, I, GT)$ после хода времени $\theta \in \mathbb{R}$, если $(I(t) + \theta) \in D(t)$. *Срабатывание* перехода t из состояния $S = (M, I, GT)$ после задержки времени θ приводит к новому состоянию $S' = (M', I', GT')$ (обозначается $S \xrightarrow{(\theta,t)} S'$ или $S \xrightarrow{(\theta,L(t))} S'$), где:

- $M' = (M \backslash {}^\bullet t) \cup t^\bullet$,

– $\forall t' \in En(M') : I'(t') = \begin{cases} I(t') + \theta, & \text{если } t' \in En(M \setminus {}^\bullet t), \\ 0, & \text{иначе,} \end{cases}$

– $GT' = GT + \theta$.

Используется обозначение $S \xrightarrow{\sigma} S'$, если $\sigma = (\theta_1, t_1) \dots (\theta_n, t_n)$ и $S = S^0 \xrightarrow{(\theta_1,t_1)} S^1 \dots S^{n-1} \xrightarrow{(\theta_n,t_n)} S_n = S'$ для некоторого $n \geqslant 0$. В этом случае σ — *пробег* $\mathcal{TN}$ из S (в S'), а S' — *достижимое состояние* сети $\mathcal{TN}$ из S. Пусть $\mathcal{RUN}(\mathcal{TN}, S)$ обозначает множество всех пробегов из S, $\mathcal{RUN}(\mathcal{TN}) = \mathcal{RUN}(\mathcal{TN}, S_0)$ и $\mathcal{RS}(\mathcal{TN})$ — множество всех достижимых состояний сети $\mathcal{TN}$ из S_0.

НВСП$_{\text{сл}}$ $\mathcal{TN}$ — *T-ограниченна*, если ${}^\bullet t \neq \emptyset$ и $t^\bullet \neq \emptyset$ для всех переходов $t \in T$; *бесконтактная*, если базовая сеть Петри бесконтактная. В дальнейшем будут рассматриваться только T-ограниченные и бесконтактные НВСП$_{\text{сл}}$.

Данная работа исследует *безопасные* НВСП$_{\text{сл}}$, в которых последовательное срабатывание переходов из начального состояния гарантирует, что каждое место содержит не более одной фишки. Это свойство следует из ранее определенного свойства бесконтактности.

Результаты работы [11] позволяют в дальнейшем ограничиться рассмотрением только целочисленного хода времени в НВСП$_{\text{сл}}$.

Пример 1. На рис. 1 показана НВСП$_{\text{сл}}$ $\mathcal{TN}_1$, в которой места представлены окружностями, а переходы — барьерами. Отношение инцидентности представлено направленными дугами. Каждое место, принадлежащее начальной разметке, имеет фишку (помечено жирной точкой). Значения статической временной функции изображены рядом с переходами. Ниже показано, что $\sigma = (1, t_1)(1, t_2)(3, t_3) \in \mathcal{RUN}(\mathcal{TN}_1)$, т.е. $S_0 \xrightarrow{(1,t_1)} S_1 \xrightarrow{(1,t_2)} S_2 \xrightarrow{(3,t_3)} S_3$.

Изначально $S_0 = (M_0, I_0, GT_0)$, где $M_0 = \{p_1, p_2\}$, $I_0(t_1) = I_0(t_2) = 0$, потому что $En(M_0) = \{t_1, t_2\}$, и $GT_0 = 0$. Переход t_1, допустимый в разметке M_0, может сработать из состояния S_0 после хода времени 1, поскольку $I_0(t_1) + 1 = 1 \in D(t_1) = [1, 2]$. Тогда $S_0 \xrightarrow{(1,t_1)} S_1 = (M_1, I_1, GT_1)$, где $M_1 = (M_0 \setminus {}^\bullet t_1) \cup t_1{}^\bullet = (\{p_1, p_2\} \setminus \{p_1\}) \cup \{p_3\} = \{p_3, p_2\}$, $En(M_1) = \{t_2\}$, $I_1(t_2) = 1$ и $GT_1 = GT_0 + 1 = 1$.

Аналогично, $t_2 \in En(M_1)$ может сработать из состояния S_1 после хода времени 1, так как $I_1(t_2) + 1 = 2 \in D(t_2) = [1, 2]$. Следовательно, $S_1 \xrightarrow{(1,t_2)} S_2 = (M_2, I_2, GT_2)$, где $M_2 = (M_1 \setminus {}^\bullet t_2) \cup t_2{}^\bullet = (\{p_3, p_2\} \setminus \{p_2\}) \cup \{p_4\} = \{p_3, p_4\}$, $En(M_2) = \{t_3, t_4\}$, $I_2(t_3) = I_2(t_4) = 0$ и $GT_2 = GT_1 + 1 = 2$.

Наконец, $S_2 \xrightarrow{(3,t_3)} S_3 = (M_3, I_3, GT_3)$, где $M_3 = (M_2 \setminus {}^\bullet t_3) \cup t_3{}^\bullet = (\{p_3, p_4\} \setminus \{p_3, p_4\}) \cup \{p_1, p_2\} = \{p_1, p_2\}$, $En(M_3) = \{t_1, t_2\}$, $I_3(t_1) = I_3(t_2) = 0$ и $GT_3 = GT_2 + 3 = 5$, потому что $I_2(t_3) + 3 = 3 \in D(t_3) = [2, 3]$, т.е. $t_3 \in En(M_2)$ может сработать из состояния S_2.

Стоит заметить, что слабая стратегия позволяет не форсировать срабатывание перехода t_4, когда время на его часах пересекает верхнюю границу его временного интервала $[0, 1]$.

2. Временные процессы

Этот раздел обобщает модель временных процессов, представленных в работе [10], на слабую стратегию хода времени НВСП. Изложение начинается с определения ациклической причинно-следственной сети, которая состоит из двух различных множеств событий и условий, связанных отношением инцидентности. Причинно-следственная сеть удовлетворяет двум структурным ограничениям. Каждое условие должно имеет не более одного вход-

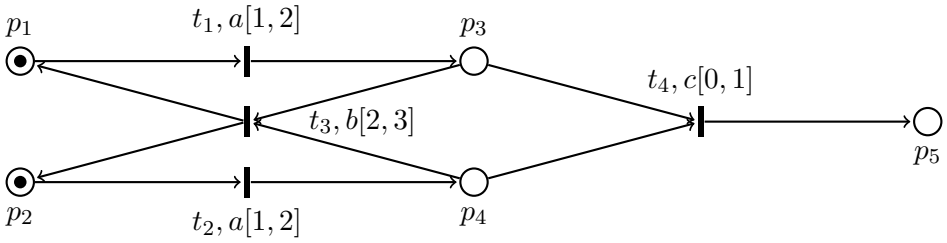


Рис. 1. Непрерывно-временная сеть Петри $\mathcal{TN}_1$

ного события и не более одного выходного события. Кроме того, каждое событие должно иметь как входное, так и выходное условие.

Определение 3. (Причинно-следственная сеть) *(Помеченная над Act) причинно-следственная сеть* (ПСС) — это ациклическая сеть $N = (B,\ E,\ G,\ l)$, где:

- B — конечное *множество условий*;
- E — конечное *множество событий* такое, что $B \cap E = \emptyset$;
- $G \subseteq (B \times E) \cup (E \times B)$ — *отношение инцидентности* такое, что $E = {}^\bullet B = B^\bullet$ и $|\, b^\bullet \,| \leqslant 1$, $|\, {}^\bullet b \,| \leqslant 1$ для всех $b \in B$;
- $l : E \to Act$ — *помечающая функция.*

Ниже приводятся дополнительные понятия и обозначения для ПСС $N = (B,\ E,\ G,\ l)$:

- $x \prec x' \iff x\ G^+\ x'$, и $x \preceq x' \iff x\ G^*\ x'$ — *отношение причинной зависимости* между элементами $x, x' \in B \cup E$.
- $x \smile x' \iff \neg(x \preceq x') \wedge \neg(x' \preceq x)$ — *отношение параллелизма* между элементами $x, x' \in B \cup E$.
- $\smile$*-множество* — непустое подмножество условий $B' \subseteq B$, где $b \smile b'$ для всех различных $b, b' \in B'$.
- *Сечение* — максимальное по включению $\smile$-множество.
- $\mathcal{CUT}(N)$ — множество всех сечений в N.
- ${}^\bullet N = \{b \in B \mid {}^\bullet b = \emptyset\}$ — *начальное сечение.*
- $N^\bullet = \{b \in B \mid b^\bullet = \emptyset\}$ — *конечное сечение.*

Событие $e \in E$ *допустимо* в сечении $C \in \mathcal{CUT}(N)$ (обозначается $e \in En(C)$), если ${}^\bullet e \subseteq C$. В этом случае используется запись $C \xrightarrow{e} C'$ или $C \to C'$, где $C' = (C \backslash {}^\bullet e) \cup e^\bullet$. Отношение $(\to)$ определяет отношения причинной зависимости и параллелизма на сечениях:

- $C \prec C' \iff C\ \to^+\ C'$, и $C \preceq C' \iff C\ \to^*\ C'$ — *отношение причинной зависимости* между сечениями $C, C' \in \mathcal{CUT}(N)$.
- $C \smile C' \iff \neg(C \preceq C') \wedge \neg(C' \preceq C)$ — *отношение параллелизма* между сечениями $C, C' \in \mathcal{CUT}(N)$.
- $\downarrow C = \{e \in E \mid e \preceq e', e' \in {}^\bullet C\}$ — *множество событий, предшествующих* сечению $C \in \mathcal{CUT}(N)$.

Сечение описывает состояние, достигнутое после наступления всех событий, которые ему предшествуют. Следовательно, $C \preceq C' \iff \downarrow C \subseteq \downarrow C'$. Следует отметить, что $\downarrow {}^\bullet N = \emptyset$ и $\downarrow N^\bullet = E$, где E — множество всех событий в ПСС N.

Пример 2. На рис. 2 показана $N_1 = (B,\ E,\ G,\ l)$, где $B = \{b_1, b_2, b_3, b_4, b_5, b_6\}$, $E = \{e_1, e_2, e_3\}$, $G = \{(b_1, e_1), (e_1, b_3), (b_2, e_2), (e_2, b_4), (b_3, e_3), (b_4, e_3), (e_3, b_5), (e_3, b_6)\}$, $l(e_1) =$

$l(e_2) = a$ и $l(e_3) = b$. Видно, что $|b^\bullet| \leqslant 1$ и $|{}^\bullet b| \leqslant 1$ для любого $b \in B$ и $E = {}^\bullet B = B^\bullet$. Поскольку $b_1 \smile b_4$ и не существует $b \in B \setminus \{b_1, b_4\}$ такого, что $b \smile b_1$ и $b \smile b_4$, то $\{b_1, b_4\}$ является сечением в N_1. Легко проверить, что множества ${}^\bullet N_1 = \{b_1, b_2\}$, $\{b_2, b_3\}$, $\{b_3, b_4\}$ и $N_1{}^\bullet = \{b_5, b_6\}$ также являются сечениями в N_1. Поскольку ${}^\bullet e_1 = \{b_1\}$, то событие e_1 допустимо в $\{b_1, b_4\}$. Тогда $\{b_1, b_4\} \xrightarrow{e_1} \{b_3, b_4\}$, потому что $(\{b_1, b_4\} \setminus {}^\bullet e_1) \cup e_1{}^\bullet = \{b_3, b_4\}$.

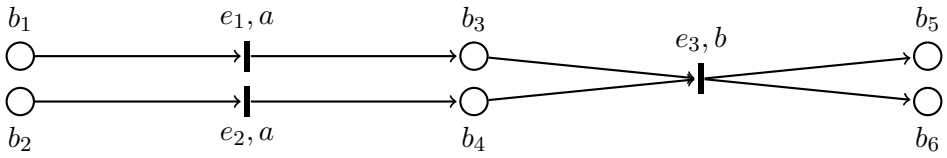


Рис. 2. Причинно-следственная сеть N_1

Для определения отношения между ПСС и сетями Петри используется *гомоморфизм* — отображение, сохраняющее отношение инцидентности и помечающую функцию.

Определение 4. (Гомоморфизм) Пусть $N = (B,\ E,\ G,\ l)$ — ПСС и $\mathcal{N} = (P,\ T,\ F,\ M_0,\ L)$ — сеть Петри с достижимой разметкой M. Отображение $\varphi : (B \cup E) \to (P \cup T)$ является *гомоморфизмом* из N в $\mathcal{N}$ относительно M, если выполняются следующие условия:

- $\varphi(B) \subseteq P$, и $\varphi(E) \subseteq T$,
- сужение φ на ${}^\bullet e$ является биекцией между ${}^\bullet e$ и ${}^\bullet\varphi(e)$ для всех $e \in E$,
- сужение φ на $e^\bullet$ является биекцией между $e^\bullet$ и $\varphi(e)^\bullet$ для всех $e \in E$,
- сужение φ на ${}^\bullet N$ является биекцией между ${}^\bullet N$ и M,
- $l(e) = L(\varphi(e))$ для всех $e \in E$.

Пример 3. Рассматривается базовая сеть Петри $\mathcal{N}_1 = (P, T, F, M_0, L)$, показанная на рис. 1. ПСС $N_1 = (B, E, G, l)$ из примера 2 и отображение φ, определенное следующим образом: $\varphi(b_1) = \varphi(b_5) = p_1$, $\varphi(b_2) = \varphi(b_6) = p_2$, $\varphi(b_3) = p_3$, $\varphi(b_4) = p_4$, $\varphi(e_1) = t_1$, $\varphi(e_2) = t_2$, $\varphi(e_3) = t_3$. Тогда верно следующее: $\varphi(B) \subseteq P$, $\varphi(E) \subseteq T$, $l(e) = L(\varphi(e))$ для всех $e \in E$, и сужение φ на подмножество ${}^\bullet N_1 = \{b_1, b_2\}$ является биекцией между ${}^\bullet N_1$ и $M_0 = \{p_1, p_2\}$. Кроме того, для всех $e \in E$ сужение φ на подмножество ${}^\bullet e$ ($e^\bullet$) является биекцией между ${}^\bullet e$ ($e^\bullet$) и ${}^\bullet\varphi(e)$ ($\varphi(e)^\bullet$). Например, сужение φ на подмножество ${}^\bullet e_3 = \{b_3, b_4\}$ является биекцией между ${}^\bullet e_3$ и ${}^\bullet\varphi(e_3) = {}^\bullet t_3 = \{p_3, p_4\}$. Следовательно, φ является гомоморфизмом из N_1 в $\mathcal{N}_1$.

Ниже приводится доказательство того, что сужение гомоморфизма на произвольное сечение является биекцией между этим сечением и разметкой сети Петри.

Лемма 1. Пусть N — ПСС, $\mathcal{N}$ — бесконтактная сеть Петри, и φ — гомоморфизм из N в $\mathcal{N}$ относительно достижимой разметки M. Тогда для всех $C \in \mathcal{CUT}(N)$ сужение φ на C является биекцией между C и $\varphi(C)$.

Доказательство. По определению ПСС выполняется $\downarrow{}^\bullet N = \emptyset$, откуда следует $\downarrow{}^\bullet N \subseteq \downarrow C$, т.е. ${}^\bullet N \preceq C$. Это означает, что существует последовательность сечений и событий:

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \ldots C_{n-1} \xrightarrow{e_n} (C_n = C),\ \text{для } n \geqslant 0.$$

Утверждение доказывается индукцией по $0 \leqslant i \leqslant n$. Будет показано, что существует последовательность разметок:

$$(M = M^0) \xrightarrow{\varphi(e_1)} M^1 \dots M^{n-1} \xrightarrow{\varphi(e_n)} M^n, \tag{1}$$

такая, что сужение φ на каждое C_i является биекцией между C_i и соответствующей разметкой M^i.

База индукции ($i = 0$)**.** Сужение φ на подмножество $C_0 = {}^\bullet N$ является биекцией между C_0 и разметкой $M = M^0$ по определению гомоморфизма.

Шаг индукции ($i > 0$)**.** По предположению индукции для индекса $i-1$ существует префикс последовательности разметок (1), и сужение φ на C_{i-1} является биекцией между C_{i-1} и M^{i-1}.

Поскольку $C_{i-1} \xrightarrow{e_i} C_i$, то ${}^\bullet e_i \subseteq C_{i-1}$. Из этого следует: $\varphi({}^\bullet e_i) \subseteq \varphi(C_{i-1}) = M^{i-1}$. По определению 4, ${}^\bullet\varphi(e_i) = \varphi({}^\bullet e_i) \subseteq M^{i-1}$, следовательно, переход $\varphi(e_i)$ допустим в разметке M^{i-1}, т.е. $\varphi(e_i) \in En(M^{i-1})$. Согласно определению 1, $M^{i-1} \xrightarrow{\varphi(e_i)} M^i$, что подтверждает существование следующего члена последовательности (1), где:

$$M^i = (M^{i-1} \setminus {}^\bullet\varphi(e_i)) \cup \varphi(e_i)^\bullet.$$

Теперь будет показано, что сужение φ на C_i также является биекцией между C_i и M^i. Из $C_{i-1} \xrightarrow{e_i} C_i$ следует $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i^\bullet$. С учетом свойств гомоморфизма и предположения индукции получаем:

$$\varphi(C_i) = (\varphi(C_{i-1}) \backslash {}^\bullet\varphi(e_i)) \cup \varphi(e_i)^\bullet = M^i.$$

Поскольку $\mathcal{N}$ — бесконтактная сеть Петри и переход $\varphi(e_i)$ допустим в разметке M^{i-1}, то выполняется условие: $(\varphi(C_{i-1}) \backslash {}^\bullet\varphi(e_i)) \cap \varphi(e_i)^\bullet = \emptyset$. Следовательно, сужение φ на C_i является биекцией между C_i и M^i.

□

Вводятся временные ПСС, состоящие из ПСС и временной функции, сопоставляющей вещественное значение каждому событию ПСС.

Определение 5. (Временная причинно-следственная сеть) *(Помеченная на Act) временная ПСС* (ВПСС) — это пара $TN = (N,\ \tau)$, где $N = (B,\ E,\ G,\ l)$ — *базовая* (помеченная на *Act*) ПСС, и $\tau : E \to \mathbb{N}$ — *временная функция* такая, что $e \preceq e' \Rightarrow \tau(e) \leqslant \tau(e')$ для всех $e, e' \in E$.

ВПСС $TN = ((B,\ E,\ G,\ l),\ \tau)$ и $TN' = ((B',\ E',\ G',\ l'),\ \tau')$ *изоморфны* (обозначается $TN \simeq TN'$ или $f : TN \simeq TN'$), если существует биективное отображение $f : B \cup E \to B' \cup E'$ (*изоморфизм*) такое, что:

- $f(B) = B'$, и $f(E) = E'$,
- $x\ G\ y \iff f(x)\ G'\ f(y)$ для всех $x, y \in B \cup E$,
- $l(e) = l'(f(e))$, и $\tau(e) = \tau'(f(e))$ для всех $e \in E$.

Для ВПСС $TN = (N, \tau)$, НВСП$_{\text{сл}}$ $\mathcal{TN} = (\mathcal{N}, D)$ и состояния $S = (M, I) \in \mathcal{RS}(\mathcal{TN})$ гомоморфизм из N в $\mathcal{N}$ относительно разметки M называется также *гомоморфизмом из TN в $\mathcal{TN}$ относительно состояния S*.

Далее рассматривается метод линеаризации событий ВПСС с учетом временных характеристик модели.

Определение 6. (Линеаризация) Пусть $TN = (N, \tau)$ — ВПСС. Последовательность событий $\omega = e_1 \dots e_n$ $(n \geqslant 0)$ называется *линеаризацией* TN, если $\tau(e_i) < \tau(e_j) \Rightarrow i < j$ для всех $1 \leqslant i, j \leqslant n$, и существует цепь сечений:

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = N^\bullet).$$

Пусть $\mathcal{LIN}(TN)$ обозначает множество всех линеаризаций TN. Заметим, что поскольку $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i^\bullet$ для всех $1 \leqslant i \leqslant n$, $C_0 = {}^\bullet N$ и $C_n = N^\bullet$, то каждое событие из TN появляется в линеаризации ровно один раз.

Пример 4. В данном примере рассматривается $TN_1 = (N_1, \tau)$, где N_1 — ПСС из примера 2 и функция τ определена следующим образом: $\tau(e_1) = 1$, $\tau(e_2) = 2$, $\tau(e_3) = 5$. Тогда $e \preceq e' \Rightarrow \tau(e) \leqslant \tau(e')$, для всех $e, e' \in E$. Следовательно, TN_1 является ВПСС. Далее рассмотрим последовательность событий $\omega = e_1\ e_2\ e_3$ с цепью сечений $C_0 \xrightarrow{e_1} C_1 \xrightarrow{e_2} C_2 \xrightarrow{e_3} C_3$, где $C_0 = {}^\bullet N_1 = \{b_1, b_2\}$, $C_1 = \{b_3, b_2\}$, $C_2 = \{b_3, b_4\}$, $C_3 = N_1^\bullet = \{b_5, b_6\}$. Видно, что $\tau(e_i) < \tau(e_j) \Rightarrow i < j$ для всех $1 \leqslant i, j \leqslant 3$, т.е. ω является линеаризацией TN_1.

Ниже показано, что для каждой ВПСС существует линеаризация.

Утверждение 1. Пусть $TN = (N = (B,\ E,\ G,\ l),\ \tau)$. Тогда $\mathcal{LIN}(TN) \neq \emptyset$.

Доказательство. Поскольку множество событий конечно, то $|E| = n$ для некоторого $n \geqslant 0$. Если $n = 0$, утверждение тривиально. Индукцией по $1 \leqslant i \leqslant n$ строится последовательность событий $\omega = e_1 \dots e_n$ с цепью сечений:

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = N^\bullet) \tag{2}$$

такая, что $\tau(e_j) \leqslant \tau(e)$ для всех $e \in E \setminus {\downarrow}C_j$ и $1 \leqslant j \leqslant n$.

База индукции $(i = 1)$**.** Так как ${\downarrow}C_0 \subset {\downarrow}N^\bullet = E$, то $C_0 \to^+ N^\bullet$ и $En(C_0) \neq \emptyset$. Пусть $e_1 \in En(C_0)$ — событие с минимальным значением функции τ. Тогда $C_0 \xrightarrow{e_1} C_1$, где $C_1 = (C_0 \setminus {}^\bullet e_1) \cup e_1^\bullet$.

Рассмотрим $e \in E \setminus {\downarrow}C_1$. Поскольку $e_1 \in En({}^\bullet N)$, то ${}^\bullet e_1 \subseteq {}^\bullet N$. Возможны два случая:
(а) $e_1 \preceq e$. Тогда $\tau(e_1) \leqslant \tau(e)$ по определению 5
(б) $e_1 \smile e$. Тогда $e \in En(C_0)$ и $\tau(e_1) \leqslant \tau(e)$ согласно выбору e_1.
Следовательно, $\tau(e_1) \leqslant \tau(e)$ для всех $e \in E \setminus {\downarrow}C_1$.

Шаг индукции $(i > 1)$**.** По предположению индукции построена последовательность (2) до C_{i-1}, причем $\tau(e_j) \leqslant \tau(e)$ для $e \in E \setminus {\downarrow}C_j$ при $1 \leqslant j \leqslant i-1$. Так как ${\downarrow}C_{i-1} \subset E$, то $En(C_{i-1}) \neq \emptyset$. Пусть $e_i \in En(C_{i-1})$ — событие с минимальным значением τ. Тогда $C_{i-1} \xrightarrow{e_i} C_i$, где $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i^\bullet$.

Пусть

$$e \in E \setminus {\downarrow}C_i = (E \setminus {\downarrow}C_{i-1}) \setminus \{e_i\}. \tag{3}$$

По предположению индукции $\tau(e_j) \leqslant \tau(e)$ для $1 \leqslant j \leqslant i-1$. Покажем, что $\tau(e_i) \leqslant \tau(e)$. Если $e \in En(C_{i-1})$, это следует из выбора e_i. Для $e \notin En(C_{i-1})$, рассмотрим случаи:
(а) $e \smile e'$ для всех $e' \in En(C_{i-1})$. Это противоречит максимальности C_{i-1} как сечения, так как $C_{i-1} \cup {}^\bullet e$ было бы $\smile$-множеством.

(б) $e \prec e'$ для некоторого $e' \in En(C_{i-1})$. Тогда $e \in {\downarrow}C_{i-1}$, что противоречит (3).
(в) $e' \prec e$ для некоторого $e' \in En(C_{i-1})$. По определению 5 и выбору e_i:

$$\tau(e_i) \leqslant \tau(e') \leqslant \tau(e).$$

В результате построенная последовательность $\omega = e_1 \dots e_n$ имеет цепь сечений (2). Докажем, что ω — линеаризация. Рассмотрим произвольные e_i, e_j из последовательности ω такие, что $\tau(e_i) < \tau(e_j)$. Предположим обратное, т.е. $i > j$. Тогда $e_i \in E \setminus {\downarrow}C_j$ из (2). По построению $\tau(e_j) \leqslant \tau(e_i)$, что противоречит $\tau(e_i) < \tau(e_j)$. Следовательно, $i < j$ и $\omega \in \mathcal{LIN}(TN)$ согласно определению 6. □

Чтобы связать временные конструкции ВПСС с НВСП$_{\text{сл}}$, используется функция TOE. Эта функция определяется в контексте достижимого состояния S и гомоморфизма φ относительно S. Функция принимает два аргумента: подмножество условий B' и переход t, допустимый в разметке $\varphi(B')$. TOE для t представляет глобальный момент модельного времени, когда фишки появляются во всех входных местах t. Используя эту функцию, можно определить временной процесс для НВСП$_{\text{сл}}$ как пару, состоящую из ВПСС и гомоморфизма.

Определение 7. (Временной процесс) Пусть $\mathcal{TN} = (\mathcal{N},\ D)$ — НВСП$_{\text{сл}}$, $TN = (N = (B,\ E,\ G,\ l),\ \tau)$ — ПСС, $S = (M, I, GT) \in \mathcal{RS}(\mathcal{TN})$ и φ — гомоморфизм из TN в $\mathcal{TN}$ относительно S.

- $\mathbf{TOE}_{\varphi,S}(B', t) = \begin{cases} \max(\{\tau(^\bullet b) \mid b \in [B']_t \setminus {}^\bullet N\}), & \text{если } [B']_t \setminus {}^\bullet N \neq \emptyset, \\ GT - I(t), & \text{иначе,} \end{cases}$
  где $[B']_t = \{b \in B' \mid \varphi(b) \in {}^\bullet t\}$, и $t \in En(\varphi(B'))$.
- Пара $\pi = (TN, \varphi)$ является *временным процессом* НВСП$_{\text{сл}}$ $\mathcal{TN}$ относительно состояния S, если для всех $e \in E$ выполняется: $\tau(e) - \mathbf{TOE}_{\varphi,S}(^\bullet e, \varphi(e)) \in D(\varphi(e))$ и $\tau(e) \geqslant GT$.

Следует отметить, что согласно лемме 1, если $e \in En(C)$ для сечения C, то $\mathbf{TOE}_{\varphi,S}(^\bullet e, \varphi(e)) = \mathbf{TOE}_{\varphi,S}(C, \varphi(e))$. Пусть $\mathcal{PRC}(\mathcal{TN}, S)$ обозначает множество всех временных процессов $\mathcal{TN}$ относительно S.

Временные процессы $\pi = (TN = ((B,\ E,\ G,\ l), \tau),\ \varphi)$ и $\pi' = (TN',\ \varphi')$ *изоморфны* (обозначается $\pi \simeq \pi'$ или $f : \pi \simeq \pi'$), если существует изоморфизм $f : TN \simeq TN'$ такой, что $\varphi(x) = \varphi'(f(x))$ для всех $x \in B \cup E$.

Пример 5. Рассмотрим НВСП$_{\text{сл}}$ $\mathcal{TN}_1$ из примера 1, временной процесс TN_1 из примера 4 и гомоморфизм φ из примера 3. Покажем, что $\pi = (TN_1, \varphi) \in \mathcal{PRC}(\mathcal{TN}_1, S_0)$, где $S_0 = (M_0, I_0, GT_0)$ — начальное состояние. Видно, что $\tau(e) \geqslant GT_0 = 0$ для всех $e \in \{e_1, e_2, e_3\}$. Вычислим функцию **TOE**:

- $\mathbf{TOE}_{\varphi,S_0}(^\bullet e_1, \varphi(e_1)) = \mathbf{TOE}_{\varphi,S_0}(\{b_1\}, t_1) = GT_0 - I_0(t_1) = 0$,
- $\mathbf{TOE}_{\varphi,S_0}(^\bullet e_2, \varphi(e_2)) = \mathbf{TOE}_{\varphi,S_0}(\{b_2\}, t_2) = GT_0 - I_0(t_2) = 0$,
- $\mathbf{TOE}_{\varphi,S_0}(^\bullet e_3, \varphi(e_3)) = \mathbf{TOE}_{\varphi,S_0}(\{b_3, b_4\}, t_2) = \max(\{\tau(^\bullet b) \mid b \in \{b_3, b_4\}\}) = \max(\{\tau(e_1), \tau(e_2)\}) = \max(\{1, 2\}) = 2$.

Тогда:

- $\tau(e_1) - \mathbf{TOE}_{\varphi,S_0}(^\bullet e_1, \varphi(e_1)) = 1 - 0 = 1 \in D(\varphi(e_1)) = D(t_1) = [1, 2]$,
- $\tau(e_2) - \mathbf{TOE}_{\varphi,S_0}(^\bullet e_2, \varphi(e_2)) = 2 - 0 = 2 \in D(\varphi(e_2)) = D(t_2) = [1, 2]$,
- $\tau(e_3) - \mathbf{TOE}_{\varphi,S_0}(^\bullet e_3, \varphi(e_3)) = 5 - 2 = 3 \in D(\varphi(e_3)) = D(t_3) = [2, 3]$.

Следовательно, $\pi = (TN_1, \varphi)$ является временным процессом $\mathcal{TN}_1$ относительно начального состояния S_0.

3. Временные процессы и пробеги

В этом разделе исследуется взаимосвязь между линеаризациями предложенных временных процессов и пробегами НВСП$_{\text{сл}}$, которые определяют ее классическую интерливинговую семантику. Изложение начинается с доказательства вспомогательного факта, который связывает сечения с разметками, а функцию TOE с динамической временной функцией НВСП$_{\text{сл}}$.

Утверждение 2. Пусть $\mathcal{TN}$ — НВСП$_{\text{сл}}$, $S = (M^0, I^0, GT^0) \in \mathcal{RS}(\mathcal{TN})$, $TN = (N = (B,\ E,\ G,\ l),\ \tau)$ — ВПСС, а φ — гомоморфизм из TN в $\mathcal{TN}$ относительно S. Пусть $\omega = e_1 \dots e_n \in \mathcal{LIN}(TN)$ для $n \geqslant 0$ с цепью сечений:

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = N^\bullet). \tag{4}$$

Если

$$(M^0, I^0, GT^0) \overset{(\tau(e_1)-GT^0,\varphi(e_1))}{\longrightarrow} \dots \overset{(\tau(e_k)-GT^{k-1},\varphi(e_k))}{\longrightarrow} (M^k, I^k, GT^k) \tag{5}$$

для $0 \leqslant k \leqslant n$, то выполняется:

(а) сужение φ на C_k является биекцией между C_k и M^k;

(б) $\mathbf{TOE}_{\varphi,S}(C_k, t) = GT^k - I^k(t)$ для всех $t \in En(M^k)$ и $GT^k = \tau(e_k)$ для $k > 0$.

Доказательство. Утверждение доказывается индукцией по $0 \leqslant i \leqslant k$.

База индукции ($i = 0$):

(а) Сужение φ на $C_0 = {}^\bullet N$ является биекцией между C_0 и M^0 согласно определению 4

(б) Рассмотрим произвольный $t \in En(M^0)$. Тогда $t \in En(\varphi(C_0))$ согласно пункту (а). Следовательно:

$$\mathbf{TOE}_{\varphi,S}(C_0, t) = \begin{cases} \max(\{\tau({}^\bullet b) \mid b \in [C_0]_t \setminus {}^\bullet N\}), & \text{если } [C_0]_t \setminus {}^\bullet N \neq \emptyset, \\ GT^0 - I^0(t), & \text{иначе.} \end{cases}$$

Так как $[C_0]_t \subseteq (C_0 = {}^\bullet N)$, то $[C_0]_t \setminus {}^\bullet N = \emptyset$, откуда получаем $\mathbf{TOE}_{\varphi,S}(C_0, t) = GT^0 - I^0(t)$.

Шаг индукции ($i > 0$): По предположению индукции сужение φ на C_{i-1} является биекцией между C_{i-1} и M^{i-1}, и выполняется $\mathbf{TOE}_{\varphi,S}(C_{i-1}, t) = GT^{i-1} - I^{i-1}(t)$ для всех $t \in En(M^{i-1})$.

Поскольку $C_{i-1} \xrightarrow{e_i} C_i$ из (4), то $e_i \in En(C_{i-1})$, т.е. ${}^\bullet e_i \subseteq C_{i-1}$ и $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i{}^\bullet$. Согласно условию (5) и определению 2

$$GT^i = GT^{i-1} + (\tau(e_i) - GT^{i-1}) = \tau(e_i).$$

(а) По определению 2 и (5), $M^i = (M^{i-1} \setminus {}^\bullet\varphi(e_i)) \cup \varphi(e_i)^\bullet$. Используя свойства гомоморфизма из определения 4

$$M^i = (M^{i-1} \setminus \varphi({}^\bullet e_i)) \cup \varphi(e_i{}^\bullet).$$

Так как ${}^\bullet e_i \subseteq C_{i-1}$, по предположению индукции имеем:

$$M^{i-1} \setminus \varphi({}^\bullet e_i) = \varphi(C_{i-1}) \setminus \varphi({}^\bullet e_i) = \varphi(C_{i-1} \setminus {}^\bullet e_i).$$

Отсюда $M^i = \varphi((C_{i-1} \backslash {}^\bullet e_i) \cup e_i{}^\bullet) = \varphi(C_i)$. По лемме 1 сужение φ на C_i — биекция между C_i и M^i.

(б) Рассмотрим произвольный $t \in En(M^i)$, т.е. $t \in En(\varphi(C_i))$ по пункту (а). Поскольку $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i{}^\bullet$ и $C_{i-1} \cap e_i{}^\bullet = \emptyset$, то:

$$\varphi(C_{i-1} \cap C_i) = \varphi(C_{i-1} \cap ((C_{i-1} \setminus {}^\bullet e_i) \cup e_i{}^\bullet)) = \varphi(C_{i-1} \setminus {}^\bullet e_i).$$

Более того, $\varphi(C_{i-1} \setminus {}^\bullet e_i) = M^{i-1} \backslash \varphi({}^\bullet e_i)$ согласно ${}^\bullet e_i \subseteq C_{i-1}$ и предположению индукции. Следовательно,

$$\varphi(C_{i-1} \cap C_i) = M^{i-1} \backslash \varphi({}^\bullet e_i) = M^{i-1} \backslash {}^\bullet \varphi(e_i) \tag{6}$$

в силу определения 4.

Предположим, что $t \in En(\varphi(C_{i-1} \cap C_i))$, т.е. $t \in En(M^{i-1} \backslash {}^\bullet \varphi(e_i))$ из (6). Тогда по определению 2

$$I^i(t) = I^{i-1}(t) + \tau(e_i) - GT^{i-1} = I^{i-1}(t) + GT^i - GT^{i-1}.$$

Благодаря лемме 1

$$[C_i]_t = \{b \in C_i \mid \varphi(b) \in {}^\bullet t\} = \{b \in C_{i-1} \cap C_i \mid \varphi(b) \in {}^\bullet t\} = \{b \in C_{i-1} \mid \varphi(b) \in {}^\bullet t\}.$$

Это означает, что $\mathbf{TOE}_{\varphi,S}(C_i, t) = \mathbf{TOE}_{\varphi,S}(C_{i-1}, t)$. Следовательно:

$$\begin{aligned}\mathbf{TOE}_{\varphi,S}(C_i, t) &= \mathbf{TOE}_{\varphi,S}(C_{i-1}, t) = GT^{i-1} - I^{i-1}(t) = \\ &= GT^{i-1} - I^{i-1}(t) + GT^i - GT^i = GT^i - (I^{i-1}(t) + GT^i - GT^{i-1}) = GT^i - I^i(t).\end{aligned}$$

Далее предположим, что $t \notin En(\varphi(C_{i-1} \cap C_i))$, т.е. $t \notin En(M^{i-1} \backslash {}^\bullet \varphi(e_i))$ согласно (6). Тогда $I^i(t) = 0$ в силу определения 2. Так как $t \in En(\varphi(C_i))$, $t \notin En(\varphi(C_{i-1} \cap C_i))$ и $C_i = (C_{i-1} \setminus {}^\bullet e_i) \cup e_i{}^\bullet$, существует $b \in [C_i]_t = \{b \in C_i \mid \varphi(b) \in {}^\bullet t\}$ такой, что $b \in e_i{}^\bullet$. Следовательно, ${}^\bullet b = e_i$, т.е. $b \notin {}^\bullet N$ и $[C_i]_t \setminus {}^\bullet N \neq \emptyset$. Значит, согласно конструкции линеаризации:

$$\mathbf{TOE}_{\varphi,S}(C_i, t) = \max(\{\tau({}^\bullet b) \mid b \in [C_i]_t \setminus {}^\bullet N\}) = \tau(e_i) = \tau(e_i) - 0 = GT^i - I^i(t).$$

□

Далее для $\omega = e_1 \dots e_n \in \mathcal{LIN}(TN)$ вводится функция *Run* следующим образом: $Run_{\varphi,S}(TN, \omega) = (\tau(e_1) - GT^0, \varphi(e_1))(\tau(e_2) - \tau(e_1), \varphi(e_2)) \dots (\tau(e_n) - \tau(e_{n-1}), \varphi(e_n))$. Ниже показано, что функция *Run* отображает любую линеаризацию временного процесса НВСП$_{\text{сл}}$ $\mathcal{TN}$ в пробег $\mathcal{TN}$.

Теорема 1. Пусть $\mathcal{TN}$ — НВСП$_{\text{сл}}$, $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}, S)$ для $S = (M, I, GT) \in \mathcal{RS}(\mathcal{TN})$, и $\omega \in \mathcal{LIN}(TN)$. Тогда $Run_{\varphi,S}(TN, \omega) \in \mathcal{RUN}(\mathcal{TN}, S)$.

Доказательство. Пусть $\omega = e_1 \dots e_n$ для $n \geqslant 0$ с цепью сечений:

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = N^\bullet).$$

Для $n = 0$ доказывать нечего. Предположим, что $n > 0$. Индукцией по $1 \leqslant i \leqslant n$ доказывается существование последовательности срабатываний:

$$S = (M^0, I^0, GT^0) \xrightarrow{(\tau(e_1) - GT^0, \varphi(e_1))} \dots \xrightarrow{(\tau(e_n) - GT^{n-1}, \varphi(e_n))} (M^n, I^n, GT^n). \tag{7}$$

База индукции ($i = 1$)**.** Согласно утверждению 2(а), сужение φ на C_0 является биекцией между C_0 и M^0. Из $C_0 \xrightarrow{e_1} C_1$ следует ${}^\bullet e_1 \subseteq C_0$, откуда: $\varphi({}^\bullet e_1) \subseteq \varphi(C_0) = M^0$. По определению 4, ${}^\bullet\varphi(e_1) \subseteq M^0$, получаем:

$$\varphi(e_1) \in En(M^0).$$

Кроме того, по утверждению 2(б), $\mathbf{TOE}_{\varphi,S}(C_0, t) = GT^0 - I^0(t)$ для всех $t \in En(M^0)$. Тогда:

$$I^0(\varphi(e_1)) + (\tau(e_1) - GT^0) = \tau(e_1) - (GT^0 - I^0(\varphi(e_1))) = \\ = \tau(e_1) - \mathbf{TOE}_{\varphi,S}(C_0, \varphi(e_1)) = \tau(e_1) - \mathbf{TOE}_{\varphi,S}({}^\bullet e_1, \varphi(e_1)) \in D(\varphi(e_1)).$$

Следовательно, по определению 2, переход $\varphi(e_1)$ может сработать из (M^0, I^0, GT^0) после хода времени $\tau(e_1) - GT^0$. Новое состояние (M^1, I^1, GT^1) последовательности (7) определяется как $M^1 = (M^0 \setminus {}^\bullet\varphi(e_1)) \cup \varphi(e_1)^\bullet$, $GT^1 = GT^0 + \tau(e_1) - GT^0 = \tau(e_1)$ и:

$$\forall t \in En(M^1) : I^1(t) = \begin{cases} I^0(t) + (\tau(e_1) - GT^0), & \text{если } t \in En(M^0 \setminus {}^\bullet\varphi(e_1)), \\ 0, & \text{иначе.} \end{cases}$$

Шаг индукции ($i > 1$)**.** По предположению индукции выполнена часть последовательности (7) до состояния $(M^{i-1}, I^{i-1}, GT^{i-1})$. Согласно утверждению 2 (аналогично рассуждениям выше), переход $\varphi(e_i)$ допустим и может сработать после хода времени $\tau(e_i) - GT^{i-1}$:

$$(M^{i-1}, I^{i-1}, GT^{i-1}) \xrightarrow{(\tau(e_i) - GT^{i-1}, \varphi(e_i))} (M^i, I^i, GT^i),$$

$M^i = (M^{i-1} \setminus {}^\bullet\varphi(e_i)) \cup \varphi(e_i)^\bullet$, $GT^i = GT^{i-1} + \tau(e_i) - GT^{i-1} = \tau(e_i)$ и:

$$\forall t \in En(M^i) : I^i(t) = \begin{cases} I^{i-1}(t) + (\tau(e_i) - GT^{i-1}), & \text{если } t \in En(M^{i-1} \setminus {}^\bullet\varphi(e_i)), \\ 0, & \text{иначе.} \end{cases}$$

По утверждению 2(б), $GT^k = \tau(e_k)$ для всех $1 \leqslant k \leqslant n$. Подставляя эти значения в (7), получаем последовательность:

$$(\tau(e_1) - GT^0, \varphi(e_1))(\tau(e_2) - \tau(e_1), \varphi(e_2)) \dots (\tau(e_n) - \tau(e_{n-1}), \varphi(e_n)) = Run_{\varphi,S}(TN, \omega).$$

Следовательно, $Run_{\varphi,S}(TN, \omega) \in \mathcal{RUN}(\mathcal{TN}, S)$. □

Далее доказывается, что для каждого пробега σ НВСП$_{\text{сл}}$ $\mathcal{TN}$ существует временной процесс $\mathcal{TN}$ такой, что Run отображает его линеаризацию в σ.

Теорема 2. Пусть $\mathcal{TN}$ — НВСП$_{\text{сл}}$, $\sigma \in \mathcal{RUN}(\mathcal{TN}, S)$ для $S \in \mathcal{RS}(\mathcal{TN})$. Тогда существуют временной процесс $\pi^* = (TN^*, \varphi^*) \in \mathcal{PRC}(\mathcal{TN}, S)$ и линеаризация $\omega^* \in \mathcal{LIN}(TN^*)$ такие, что $Run_{\varphi^*,S}(TN^*, \omega^*) = \sigma$.

Доказательство. Предположим, что $\mathcal{TN} = (\mathcal{N},\ D)$ с $\mathcal{N} = (P,\ T,\ F,\ M_0,\ L)$ (помеченной на Act), и задан пробег $\sigma = (\theta_1,\ t_1) \dots (\theta_n,\ t_n) \in \mathcal{RUN}(\mathcal{TN}, S)$ с последовательностью состояний:

$$S = (M^0, I^0, GT^0) \xrightarrow{(\theta_1,\ t_1)} \dots \xrightarrow{(\theta_n,\ t_n)} (M^n, I^n, GT^n), \tag{8}$$

для $n \geqslant 0$.

Сначала построим ПСС $N^* = (B^*,\ E^*,\ G^*,\ l^*)$ следующим образом:

– $B^* = \{b^*_{0,p} \mid p \in M^0\} \cup \{b^*_{i,p} \mid p \in t_i{}^\bullet, 1 \leqslant i \leqslant n\}$;
– $E^* = \{e^*_i \mid 1 \leqslant i \leqslant n\}$;
– $G^* = \{(e^*_i, b^*_{i,p}) \mid p \in t_i{}^\bullet, 1 \leqslant i \leqslant n\} \cup$
$\cup \{(b^*_{j,p}, e^*_i) \mid p \in {}^\bullet t_i, j = \max\{k < i \mid b^*_{k,p} \in B^*\}, 1 \leqslant i \leqslant n\}$;
– $l^* : E^* \to Act$ такая, что $l^*(e^*_i) = L(t_i)$ для $1 \leqslant i \leqslant n$.

По определению 1 P и T — конечные множества, и $P \cap T = \emptyset$. Тогда B^* и E^* также конечны и $B^* \cap E^* = \emptyset$. Легко видеть, что $G^* \subseteq (B^* \times E^*) \cup (E^* \times B^*)$, $|b^*{}^\bullet| \leqslant 1$ и $|{}^\bullet b^*| \leqslant 1$ для каждого $b^* \in B^*$. Поскольку ${}^\bullet t \neq \emptyset$ и $t^\bullet \neq \emptyset$ для всех $t \in T$, то $E^* = {}^\bullet B^* = B^{*\bullet}$. Следовательно, $N^* = (B^*, E^*, G^*, l^*)$ является причинно-следственной сетью, по определению 3.

Далее определим $\varphi^* : (B^* \cup E^*) \to (P \cup T)$ так, что:

$$\varphi^*(b^*_{i,p}) = p \text{ и } \varphi^*(e^*_i) = t_i$$

для всех $b^*_{i,p} \in B^*$ и $e^*_i \in E^*$. Согласно конструкции N^* и определению 4 φ^* является гомоморфизмом из N^* в $\mathcal{N}$ относительно M^0.

Пусть $C^*_0 = {}^\bullet N^*$ и $C^*_i = (C^*_{i-1} \backslash {}^\bullet e^*_i) \cup e^{*\bullet}_i$ для $1 \leqslant i \leqslant n$. Рассмотрим произвольный $1 \leqslant i \leqslant n$. Поскольку $e^*_i \in En(C^*_{i-1})$ и $C^*_i = (C^*_{i-1} \backslash {}^\bullet e^*_i) \cup e^{*\bullet}_i$, то $C^*_{i-1} \xrightarrow{e^*_i} C^*_i$. Следовательно, имеет место цепь сечений:

$$(C^*_0 = {}^\bullet N^*) \xrightarrow{e^*_1} C^*_1 \dots C^*_{n-1} \xrightarrow{e^*_n} (C^*_n = N^{*\bullet}). \tag{9}$$

Пусть $\tau^* : E^* \to \mathbb{N}$ — функция такая, что:

$$\tau^*(e^*_1) = GT^0 + \theta_1 \text{ и } \tau^*(e^*_i) = \tau^*(e^*_{i-1}) + \theta_i \text{ для } 1 < i \leqslant n.$$

Следовательно, $e^*_i \preceq e^*_j \Rightarrow i \leqslant j \Rightarrow \tau^*(e^*_i) \leqslant \tau^*(e^*_j)$ для $1 \leqslant i, j \leqslant n$, т.е. $TN^* = (N^*, \tau^*)$ является временной причинно-следственной сетью в силу определения 5. Более того, $\omega^* = e^*_1 \dots e^*_n$ является линеаризацией, по определению 6, потому что $\tau^*(e^*_i) < \tau^*(e^*_j) \Rightarrow i < j$ для всех $1 \leqslant i \leqslant j \leqslant n$ и (9). Отсюда следует:

$$Run_{\varphi^*,S}(TN^*, \omega^*) = (\tau^*(e^*_1) - GT^0, \varphi^*(e^*_1))\ (\tau^*(e^*_2) - \tau^*(e^*_1), \varphi^*(e^*_2)) \dots$$
$$\dots (\tau^*(e^*_n) - \tau^*(e^*_{n-1}), \varphi^*(e^*_n)) = (\theta_1, t_1) \dots (\theta_n, t_n) = \sigma.$$

Остается проверить, что $\pi^* = (TN^*, \varphi^*) \in \mathcal{PRC}(\mathcal{TN}, S)$.

Рассмотрим произвольный $e^*_i \in E^*$. Видно что, $\tau^*(e^*_i) \geqslant GT^0$. Теперь докажем, что $\tau^*(e^*_i) - \mathbf{TOE}_{\varphi^*,S}({}^\bullet e^*_i, \varphi^*(e^*_i)) \in D(t_i)$. Поскольку $(M^{i-1}, I^{i-1}, GT^{i-1}) \xrightarrow{(\theta_i,\ \varphi^*(e^*_i))} (M^i, I^i, GT^i)$ из (8), то $\varphi^*(e^*_i) \in En(M^{i-1})$ и $I^{i-1}(\varphi^*(e^*_i)) + \theta_i \in D(\varphi^*(e^*_i))$, по определению 2. На основании утверждения 2(б) получаем цепочку равенств:

$$\tau^*(e^*_i) - \mathbf{TOE}_{\varphi^*,S}({}^\bullet e^*_i, \varphi^*(e^*_i)) = \tau^*(e^*_i) - \mathbf{TOE}_{\varphi^*,S}(C^*_{i-1}, \varphi^*(e^*_i)) =$$
$$= \tau^*(e^*_i) - GT^{i-1} + I^{i-1}(\varphi^*(e^*_i)) = \theta_i + I^{i-1}(\varphi^*(e^*_i)) \in D(\varphi^*(e^*_i)).$$

Следовательно, $\pi^* = (TN^*, \varphi^*) \in \mathcal{PRC}(\mathcal{TN}, S)$, в силу произвольности выбора $e^*_i \in E^*$. $\square$

Наконец, устанавливается, что временной процесс НВСП$_{\text{сл}}$ $\mathcal{TN}$ в теореме 2 является единственным с точностью до изоморфизма.

Теорема 3. Пусть $\mathcal{TN}$ — НВСП$_{\text{сл}}$, $\sigma \in \mathcal{RUN}(\mathcal{TN}, S)$ для $S = (M, I, GT) \in \mathcal{RS}(\mathcal{TN})$. Тогда временной процесс $\pi^* = (TN^*, \varphi^*) \in \mathcal{PRC}(\mathcal{TN}, S)$ такой, что $Run_{\varphi^*,S}(TN^*, \omega^*) = \sigma$ для $\omega^* \in \mathcal{LIN}(TN^*)$, является единственным с точностью до изоморфизма.

Доказательство. Пусть $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}, S)$ с $TN = (N = (B,\ E,\ G,\ l),\ \tau)$ и линеаризацией $\omega = e_1 \dots e_n \in \mathcal{LIN}(TN)$ с цепью сечений

$$({}^\bullet N = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = N^\bullet)$$

для $n \geqslant 0$ такие, что:

$$Run_{\varphi,S}(TN, \omega) = (\tau(e_1) - GT, \varphi(e_1))\ (\tau(e_2) - \tau(e_1), \varphi(e_2)) \dots \\ \dots (\tau(e_n) - \tau(e_{n-1}), \varphi(e_n)) = (\theta_1, t_1) \dots (\theta_n, t_n) = \sigma \in \mathcal{RUN}(\mathcal{TN}, S).$$

Поскольку каждое событие из TN появляется в линеаризации ровно один раз, то $E = \{e_1, \dots, e_n\}$. Рассмотрим множество позиций B. По определению 3, $|{}^\bullet b| \leqslant 1$ для всех $b \in B$ и $E = {}^\bullet B$. Тогда либо $b \in {}^\bullet N$, либо существует единственный $e \in E$ такой, что $\{e\} = {}^\bullet b$, т.е. $b \in e^\bullet$.

Согласно определению 4, сужение φ на ${}^\bullet N$ является биекцией между ${}^\bullet N$ и M. Кроме того, сужение φ на ${e_i}^\bullet$ является биекцией между ${e_i}^\bullet$ и $\varphi(e_i)^\bullet = {t_i}^\bullet$ для всех $e_i \in E$. Без потери общности предположим:

$$B = \{b_{0,p} \mid p \in M\} \cup \{b_{i,p} \mid p \in {t_i}^\bullet, 1 \leqslant i \leqslant n\},$$

где $\varphi(b_{i,p}) = p$ для каждого $b_{i,p} \in B$.

Рассмотрим множество G. Для произвольного $e_i \in E$, благодаря конструкции B, ${e_i}^\bullet = \{b_{i,p} \in B \mid p \in {t_i}^\bullet\}$. Из $C_{i-1} \xrightarrow{e_i} C_i$ следует ${}^\bullet e_i \subseteq C_{i-1}$. Согласно лемме 1 и определению 4, сужение φ на ${}^\bullet e_i$ является биекцией между ${}^\bullet e_i$ и ${}^\bullet t_i$. Тогда:

$${}^\bullet e_i = \{b_{j,p} \in B \mid p \in {}^\bullet t_i, b_{j,p} \in C_{i-1}\}.$$

Поскольку $C_0 = {}^\bullet N$ и $C_k = (C_{k-1} \setminus {}^\bullet e_k) \cup {e_k}^\bullet$ для $1 \leqslant k \leqslant n$, то $j = \max\{k < i \mid b_{k,p} \in B\}$. Следовательно, получаем:

$$G = \{(e_i, b_{i,p}) \in E \times B \mid p \in {t_i}^\bullet, 1 \leqslant i \leqslant n\} \cup \\ \cup \{(b_{j,p}, e_i) \in B \times E \mid p \in {}^\bullet t_i, j = \max\{k < i \mid b_{k,p} \in B\}, 1 \leqslant i \leqslant n\}.$$

По определению гомоморфизма $l(e_i) = L(t_i)$. Из равенства пробегов следует:

$$\tau(e_1) = GT + \theta_1, \quad \tau(e_i) = \tau(e_{i-1}) + \theta_i \text{ для } 1 < i \leqslant n.$$

Сравним π с процессом π^* из доказательства теоремы 2. Из равенства линеаризаций $|E^*| = |E| = n$. Определим функцию $f : B \cup E \to B^* \cup E^*$ как $f(b_{i,p}) = b^*_{i,p}$ и $f(e_i) = e^*_i$. Тогда:

(а) $f(B) = B^*$ и $f(E) = E^*$;

(б) $x\ G\ y \iff f(x)\ G^*\ f(y)$ для всех $x, y \in B \cup E$;

(в) $l(e) = l^*(f(e))$ и $\tau(e) = \tau^*(f(e))$ для всех $e \in E$.
Следовательно, f является изоморфизмом $f : TN \simeq TN^*$. Так как $\varphi(x) = \varphi^*(f(x))$ для всех $x, y \in B \cup E$, то $f : \pi \simeq \pi^*$. □

4. Реализация и вычислительные эксперименты

В последние десятилетия аппарат сетей Петри активно применяется в биоинформатике [12]. Способность данного формализма отображать параллелизм, причинно-следственные связи и недетерминизм процессов позволяет адекватно моделировать динамику молекулярных взаимодействий. Растущий объем и структурная сложность накопленных моделей требуют разработки новых вычислительных методов анализа, так как традиционные подходы сталкиваются с вычислительными ограничениями при увеличении числа компонентов сети [13]. Использование причинно-следственной семантики в этих условиях позволяет сократить анализируемое пространство состояний модели за счет исключения избыточного перебора последовательностей срабатываний параллельных переходов.

Учет временных характеристик, таких как длительность реакций и время жизни молекулярных соединений, требует применения расширений сетей Петри [14, 15]. Для подобных систем предпочтительно использование НВСП со слабой временной стратегией. В отличие от сильной стратегии, наличие ресурсов и выполнение временных условий являются необходимыми, но не достаточными условиями для срабатывания перехода. Данный подход позволяет описывать биохимические процессы, в которых реализация реакции может произойти в любой момент внутри заданного временного интервала или не произойти вовсе.

На рис. 3 представлен фрагмент модели сигнального каскада RKIP/ERK (подсистема пути MAPK) [16]. В структуре данной НВСП$_{\text{сл}}$ места соответствуют молекулярным соединениям и белковым комплексам, а переходы — биохимическим реакциям. В зависимости от типа реакции переходам сопоставлены временные интервалы срабатывания: для реакций связывания (переходы t_1, t_4, t_7, t_{10}) задан интервал $[0, 1]$, для реакций преобразования (t_3, t_9) — $[0, 3]$, а для реакций распада $(t_2, t_5, t_6, t_8, t_{11})$ — $[2, 3]$.

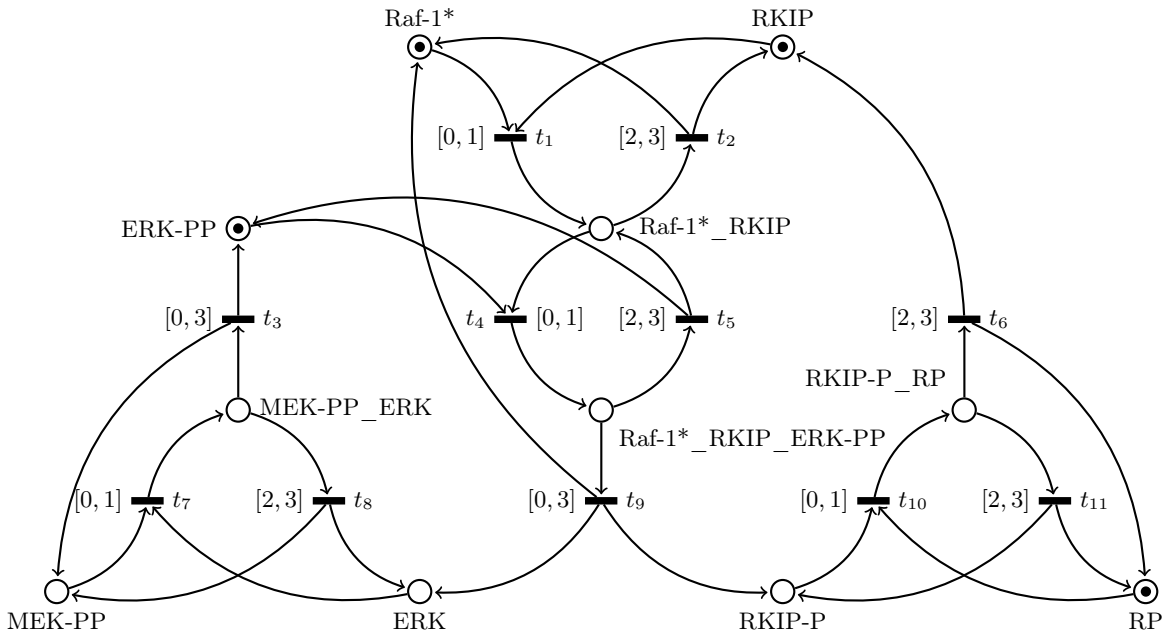


Рис. 3. Непрерывно-временная сеть Петри RKIP/ERK

Начальное состояние НВСП$_{\text{сл}}$ определяется разметкой $\{ERK\text{-}PP, Raf\text{-}1^*, RKIP, RP\}$ и нулевым значением локальных часов единственного допустимого перехода t_1. Данный переход готов сработать без временной задержки. Срабатывание t_1 приводит к новой разметке $\{ERK\text{-}PP, Raf\text{-}1^*_RKIP, RP\}$, в которой допустимы два конфликтных перехода — t_2 и t_4.

В терминах сильной временной стратегии переход t_4 обязан сработать не позднее момента времени 1. Это исключает возможность срабатывания перехода t_2, так как нижняя граница его временного интервала строго больше верхней границы интервала перехода t_4. В отличие от этого, слабая временная стратегия не требует обязательного срабатывания t_4, что допускает дальнейшее увеличение значений локальных часов переходов. При достижении момента времени 2 переход t_2 также получает возможность сработать. Таким образом, рассматриваемая модель позволяет исследовать сценарии, в которых реализуются «медленные» реакции при отсутствии выполнения более «быстрых».

Структура исследуемого каскада допускает параллельные пути передачи сигнала. Из достижимых состояний в одни и те же моменты времени возможны срабатывания параллельных переходов, не имеющих общих входных мест: $\{t_7, t_{10}\}$, $\{t_8, t_{11}\}$, $\{t_8, t_6\}$, $\{t_3, t_6\}$ и других. В классической интерливингивой семантике анализ модели потребовал бы рассмотрения всех возможных перестановок срабатываний параллельных переходов, что ведет к экспоненциальному росту числа исследуемых пробегов. Напротив, разработанная причинно-следственная семантика позволяет представить срабатывания таких переходов в виде единого причинно-следственного временного процесса.

Таблица 1. Результаты эксперимента для $n \leq 16$

Модель EGFr			
n	N_{states}	N_{cuts}	K_{red}
1	14	14	**1.00**
2	185	153	**1.21**
3	2569	1573	**1.63**
4	36 865	15 486	**2.38**
5	530 098	143 465	**3.69**
6	7 475 994	1 225 982	**6.10**
7	101 848 894	9 510 052	**10.71**

Модель MAPKbis			
1	1	1	**1.00**
2	2	2	**1.00**
3	8	8	**1.00**
4	36	33	**1.09**
5	169	128	**1.32**
6	848	486	**1.74**
7	4572	1959	**2.33**
8	27 534	9072	**3.04**
9	190 360	47 995	**3.97**
10	1 489 035	271 746	**5.48**
11	12 775 006	1 562 320	**8.18**
12	116 033 541	8 882 586	**13.06**

Модель simpleLBS			
n	N_{states}	N_{cuts}	K_{red}
1	3	3	**1.00**
2	8	7	**1.14**
3	18	12	**1.50**
4	25	14	**1.79**
5	73	44	**1.66**
6	264	132	**2.00**
7	974	372	**2.62**
8	3487	920	**3.79**
9	10 771	1804	**5.97**
10	33 262	4096	**8.12**
11	125 622	9880	**12.71**
12	473 935	21 288	**22.26**
13	1 931 822	50 392	**38.34**
14	7 672 227	115 072	**66.67**
15	28 063 891	236 800	**118.51**
16	98 573 820	483 984	**203.67**

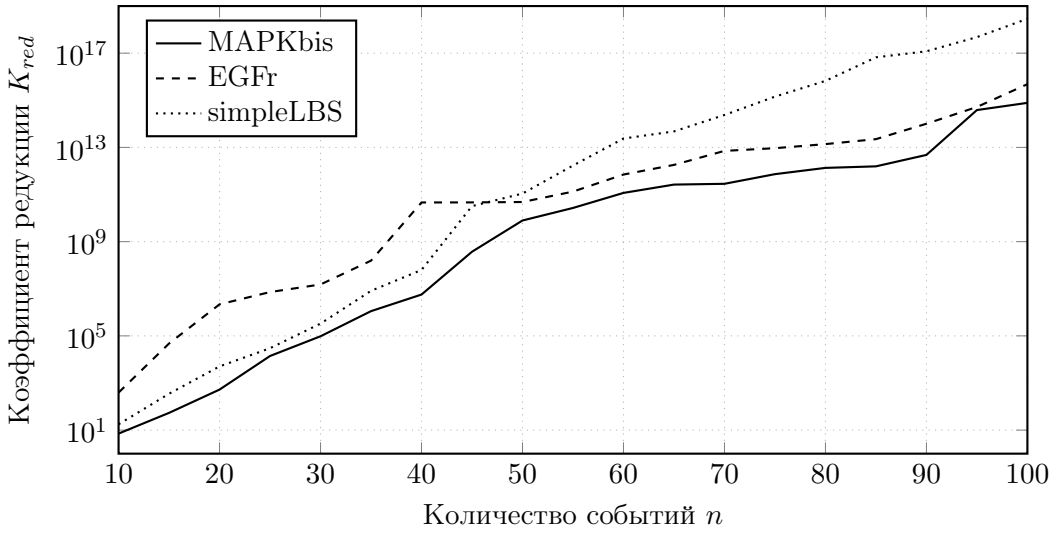


Рис. 4. Результаты эксперимента для $n \leq 100$ (логарифмическая шкала)

Далее представлены результаты апробации алгоритма построения временных процессов, предложенного в доказательстве теоремы 2. Целью вычислительных экспериментов являлась количественная оценка редукции пространства состояний $\text{НВСП}_{\text{сл}}$ при переходе от анализа пробегов к анализу временных процессов. Объектами исследования послужили биологические модели сигнальных каскадов EGFr (Egfr-104-2-0), MAPK (MAPKbis-53-1-0) [17, 18], а также модель системы балансировки нагрузки simpleLBS (simple_lbs-2) из репозитория Model Checking Contest [19]. Для трансформации исходных сетей Петри в $\text{НВСП}_{\text{сл}}$ каждому переходу был сопоставлен временной интервал, выбранный случайным образом из набора $\{[0,0],[0,1],[1,1]\}$.

Разработанный экспериментальный инструмент был реализован на языке Python (исходный код доступен по адресу: https://gitverse.ru/auzubarev/time-process.git). Программа включает следующие модули:

- *parser.py*: считывание сети Петри из формата pnml и ее трансформация в $\text{НВСП}_{\text{сл}}$;
- *interleaving.py*: реализация классической интерливинговой семантики;
- *process.py*: реализация предложенной причинно-следственной семантики;
- *view.py*: вспомогательный модуль визуализации $\text{НВСП}_{\text{сл}}$ и временных процессов.

Управляющая логика инструмента реализована в модуле *main.py*, где для каждого пробега итеративно строится соответствующий ему временной процесс. Для подсчета временных процессов проводится проверка на изоморфизм согласно представленным в работе теоремам. Эффективность редукции оценивалась коэффициентом:

$$K_{red} = \frac{N_{states}}{N_{cuts}},$$

где N_{states} — число анализируемых состояний, а N_{cuts} — число анализируемых сечений временных процессов $\text{НВСП}_{\text{сл}}$.

Вычислительные эксперименты проводились на персональном компьютере (Intel Core Ultra 5 228V 2.10 ГГц, 32 ГБ RAM). Результаты тестирования представлены в табл. 1, где n — число событий во временном процессе. При малых n число анализируемых состояний и сечений сопоставимо, однако с увеличением n наблюдается рост коэффициента редукции. Для оценки коэффициента редукции на большем количестве событий была проанализиро-

вана случайная выборка из 500 попарно неизоморфных временных процессов, каждый из которых включает до 100 событий. Зависимость на рис. 4 демонстрирует экспоненциальный рост коэффициента редукции при увеличении числа событий.

Заключение

В статье определена причинно-следственная семантика для НВСП$_{\text{сл}}$, основанная на модели временных процессов. Конструкция временного процесса состоит из временной причинно-следственной сети и гомоморфизма, отображающего ее элементы в соответствующие элементы НВСП$_{\text{сл}}$. Корректность предложенной семантической модели подтверждается установлением взаимооднозначного соответствия между линеаризациями временных процессов и пробегами НВСП$_{\text{сл}}$. Анализ данных, полученных в ходе экспериментов с моделями MAPK, EGFr и simpleLBS, подтверждает значительную редукцию пространства состояний НВСП$_{\text{сл}}$ и возможность использования предлагаемой причинно-следственной семантики в инструментах верификации сигнальных каскадов.

В качестве будущих исследований планируется определить различные поведенческие эквивалентности [20] и исследовать взаимосвязи между ними на базе предложенной модели временных процессов.

Обозначения

НВСП	непрерывно-временная сеть Петри
НВСП$_{\text{сл}}$	НВСП со слабой стратегией хода времени
ПСС	причинно-следственная сеть
ВПСС	временная причинно-следственная сеть
Act	множество действий
$\mathcal{N}$	сеть Петри
P, T, F	места, переходы и отношение инцидентности сети Петри
M_0	начальная разметка
L	помечающая функция НВСП
${}^\bullet x, x^\bullet$	множества входных и выходных элементов
M	разметка
$En(M)$	множество всех переходов, допустимых в разметке M
$Interv$	множество временных интервалов с рациональными границами
$\mathcal{TN}$	непрерывно-временная сеть Петри
D	статическая временная функция
S	состояние НВСП
I	динамическая временная функция
GT	глобальный момент времени
$S \xrightarrow{(\theta,t)} S'$	срабатывание перехода t после задержки θ в НВСП
σ	пробег
$\mathcal{RUN}(\mathcal{TN}, S)$	множество всех пробегов из состояния S
$\mathcal{RS}(\mathcal{TN})$	множество всех достижимых состояний сети
N	причинно-следственная сеть

B, E, G	условия, события и отношение инцидентности ПСС
l	помечающая функция ПСС
$\prec, \preceq, \smile$	отношения строгой и нестрогой причинной зависимости, параллелизма
$CUT(N)$	множество всех сечений ПСС N
${}^\bullet N, N^\bullet$	начальное и конечное сечения ПСС N
$En(C)$	множество событий, допустимых в сечении C
$C \xrightarrow{e} C'$	изменение сечения в результате события e
$\downarrow C$	множество событий, предшествующих сечению C
φ	гомоморфизм
TN	временная причинно-следственная сеть
τ	временная функция ВПСС
ω	линеаризация
$\mathcal{LIN}(TN)$	множество всех линеаризаций ВПСС TN
π	временный процесс
$\mathcal{PRC}(\mathcal{TN}, S)$	множество всех временных процессов $\mathcal{TN}$ относительно S

Литература

1. Komenda J., Lefebvre D. On tick automata for distributed timed DESs with synchronisations and minimal time constraints // IFAC-PapersOnLine. 2023. Vol. 56, no. 2. P. 8635–8640. DOI: 10.1016/j.ifacol.2023.10.039.

2. Boyer M., Roux O.H. Comparison of the expressiveness of arc, place and transition time Petri nets // 28th International Conference on Applications and Theory of Petri Nets, ICATPN 2007, Siedlce, Poland, June 25–29, 2007. Vol. 4546. Berlin, Heidelberg: Springer, 2007. P. 63–82. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-73094-1_7.

3. Komenda J., Lahaye S., Parrot R., Roux O.H. Weakly strong semantics of time Petri nets for performance evaluations // IFAC-PapersOnLine. 2024. Vol. 58, no. 1. P. 66–71. DOI: 10.1016/j.ifacol.2024.07.012.

4. Bérard B., Cassez F., Haddad S., *et al.* Comparison of the expressiveness of timed automata and time Petri nets // 3rd International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2005, Uppsala, Sweden, September 26–28, 2005. Vol. 3829. Berlin, Heidelberg: Springer, 2005. P. 211–225. Lecture Notes in Computer Science. DOI: 10.1007/11603009_17.

5. Reynier P.A., Sangnier A. Weak time Petri nets strike back! // 20th International Conference on Concurrency Theory, CONCUR 2009, Bologna, Italy, September 1–4, 2009. Vol. 5710. Berlin, Heidelberg: Springer, 2009. P. 557–571. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-04081-8_37.

6. Lime D., Parrot R., Roux O.H. Decidability problems for weak time Petri nets with read, reset and transfer arcs // 46th International Conference on Application and Theory of Petri Nets and Concurrency, Petri Nets 2025, Paris, France, June 23–27, 2025. Vol. 15714. Cham: Springer, 2025. P. 289–308. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-94634-9_16.

7. Leclercq L., Lime D., Roux O.H. A state class based controller synthesis approach for time Petri nets // 44th International Conference on Application and Theory of Petri Nets and Concurrency, Petri Nets 2023, Lisbon, Portugal, June 25–30, 2023. Vol. 13929. Cham: Springer, 2023. P. 409–429. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-33620-1_21.

8. Hadjidj R. An efficient approach for model-checking zeno behaviors in real-time system models based on the time Petri net formalism // IEEE Transactions on Automation Science and Engineering. 2024. Vol. 21, no. 4. P. 6628–6642. DOI: 10.1109/TASE.2023.3328895.

9. Godary-Dejean K., Leroux H., Andreu D. Interpreted synchronous extension of time Petri nets // Discrete Event Dynamic Systems. 2022. Vol. 32. P. 27–64. DOI: 10.1007/s10626-021-00347-z.

10. Aura T., Lilius J. A causal semantics for time Petri nets // Theoretical Computer Science. 2000. Vol. 243. P. 409–447. DOI: 10.1016/S0304-3975(99)00114-0.

11. Zubarev A. State space reduction for time Petri nets with weak semantics // Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2019. No. 43. P. 39–52. DOI: 10.31144/bncc.cs.2542-1972.2019.n43.p39-52.

12. Heiner M., Gilbert D., Donaldson R. Petri nets for systems and synthetic biology // 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008, Bertinoro, Italy, June 2–7, 2008. Vol. 5016. Berlin, Heidelberg: Springer, 2008. P. 215–264. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-68894-5_7.

13. Szawulak B., Radom M., Formanowicz P. Comparison by partition—Finding Petri nets similarity on the basis of subnets // Computers in Biology and Medicine. 2025. Vol. 187. Art. 109775. DOI: 10.1016/j.compbiomed.2025.109775.

14. Olszak J., Radom M., Formanowicz P. Some aspects of modeling and analysis of complex biological systems using time Petri nets // Bulletin of the Polish Academy of Sciences. Technical Sciences. 2018. Vol. 66, no. 1. P. 67–78. DOI: 10.24425/119060.

15. Gilin V., Laauwen S., Xia Y., *et al.* Modeling HIF-ILK interaction using continuous Petri nets // 18th Biomedical Engineering Systems and Technologies, BIOSTEC 2025, Porto, Portugal, February 20–22, 2025, Proceedings of the 18th International Joint Conference. Setúbal: SCITEPRESS, 2025. P. 537–544. DOI: 10.5220/0013102600003911.

16. Gilbert D., Heiner M. From Petri Nets to Differential Equations — An Integrative Approach for Biochemical Network Analysis // 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2006, Turku, Finland, June 26–30, 2006. Vol. 4024. Berlin, Heidelberg: Springer, 2006. P. 200–219. Lecture Notes in Computer Science. DOI: 10.1007/11767589_11.

17. Grieco L., Calzone L., Bernard-Pierrot I., *et al.* Integrative modelling of the influence of MAPK network on cancer cell fate decision // PLoS Computational Biology. 2013. Vol. 9, no. 10. DOI: 10.1371/journal.pcbi.1003286.

18. Sahin O., Frohlich H., Lobke C., *et al.* Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance // BMC Systems Biology. 2009. Vol. 3, no. 1. P. 1–20. DOI: 10.1186/1752-0509-3-1.

19. Kordon F., Hulin-Hubard F., Jezequel L., *et al.* Complete results for the 2025 edition of the Model Checking Contest. 2025. URL: https://mcc.lip6.fr/2025/results.php (дата обращения: 14.12.2025).

20. Virbitskaite I.B., Bushin D., Best E. True concurrent equivalences in time Petri nets // Fundamenta Informaticae. 2016. Vol. 149. P. 401–418. DOI: 10.3233/FI-2016-1454.

Зубарев Алексей Юрьевич, младший научный сотрудник, Институт систем информатики им. А.П. Ершова СО РАН (Новосибирск, Российская Федерация)

DOI: 10.14529/cmse260103

A CAUSAL SEMANTICS FOR TIME PETRI NETS WITH WEAK TIME POLICY

© 2026 A.Y. Zubarev

A.P. Ershov Institute of Informatics Systems, SB RAS
(pr. Acad. Lavrentjev 6, Novosibirsk 630090, Russia)
E-mail: auzubarev@gmail.com

Received: 14.12.2025

Time Petri Nets (TPNs) are an extension of classical Petri Nets designed for modeling and analyzing concurrent and distributed systems with respect to real-time characteristics. In TPNs, transitions representing system events have local clocks and firing time intervals. Two time-elapsing policies are considered for this model: the strong policy, where the progress of time is blocked by the necessity of a transition firing, and the weak policy, which does not restrict the progress of model time. These policies are incomparable in expressive power, and many standard verification problems are decidable only for the weak policy. In the general case, the state space of TPNs is infinite and uncountable, which complicates the analysis of model behavior. To reduce the state space, causal semantics is applied, describing system behavior through a partial order relation on the set of events. This approach allows for abstraction from the redundant set of all possible event interleavings while preserving information about causal dependencies and concurrency between system events. The paper introduces a model of time processes that represents causal semantics for TPNs in the context of the weak time policy. The main result of the work is the establishment of a bijective correspondence between TPN runs (representing classical interleaving semantics) and linearizations of time processes, which proves the correctness of the proposed causal semantics. Computational experiments presented in the paper confirm a significant reduction in the analyzed state space of TPNs when using time processes.

Keywords: time Petri net, weak time elapsing policy, time process, causal semantics.

FOR CITATION

Zubarev A.Y. A Causal Semantics for Time Petri Nets with Weak Time Policy. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2026. Vol. 15, no. 1. P. 57–79. (in Russian) DOI: 10.14529/cmse260103.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Komenda J., Lefebvre D. On tick automata for distributed timed DESs with synchronisations and minimal time constraints. IFAC-PapersOnLine. 2023. Vol. 56, no. 2. P. 8635–8640.

DOI: 10.1016/j.ifacol.2023.10.039.

2. Boyer M., Roux O.H. Comparison of the expressiveness of arc, place and transition time Petri nets. 28th International Conference on Applications and Theory of Petri Nets, ICATPN 2007, Siedlce, Poland, June 25–29, 2007. Vol. 4546. Berlin, Heidelberg, Springer, 2007. P. 63–82. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-73094-1_7.

3. Komenda J., Lahaye S., Parrot R., Roux O.H. Weakly strong semantics of time Petri nets for performance evaluations. IFAC-PapersOnLine. 2024. Vol. 58, no. 1. P. 66–71. DOI: 10.1016/j.ifacol.2024.07.012.

4. Bérard B., Cassez F., Haddad S., *et al.* Comparison of the expressiveness of timed automata and time Petri nets. 3rd International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2005, Uppsala, Sweden, September 26–28, 2005. Vol. 3829. Berlin, Heidelberg, Springer, 2005. P. 211–225. Lecture Notes in Computer Science. DOI: 10.1007/11603009_17.

5. Reynier P.A., Sangnier A. Weak time Petri nets strike back! 20th International Conference on Concurrency Theory, CONCUR 2009, Bologna, Italy, September 1–4, 2009. Vol. 5710. Berlin, Heidelberg, Springer, 2009. P. 557–571. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-04081-8_37.

6. Lime D., Parrot R., Roux O.H. Decidability problems for weak time Petri nets with read, reset and transfer arcs. 46th International Conference on Application and Theory of Petri Nets and Concurrency, Petri Nets 2025, Paris, France, June 23–27, 2025. Vol. 15714. Cham, Springer, 2025. P. 289–308. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-94634-9_16.

7. Leclercq L., Lime D., Roux O.H. A state class based controller synthesis approach for time Petri nets. 44th International Conference on Application and Theory of Petri Nets and Concurrency, Petri Nets 2023, Lisbon, Portugal, June 25–30, 2023. Vol. 13929. Cham, Springer, 2023. P. 409–429. Lecture Notes in Computer Science. DOI: 10.1007/978-3-031-33620-1_21.

8. Hadjidj R. An efficient approach for model-checking zeno behaviors in real-time system models based on the time Petri net formalism. IEEE Transactions on Automation Science and Engineering. 2024. Vol. 21, no. 4. P. 6628–6642. DOI: 10.1109/TASE.2023.3328895.

9. Godary-Dejean K., Leroux H., Andreu D. Interpreted synchronous extension of time Petri nets. Discrete Event Dynamic Systems. 2022. Vol. 32. P. 27–64. DOI: 10.1007/s10626-021-00347-z.

10. Aura T., Lilius J. A causal semantics for time Petri nets. Theoretical Computer Science. 2000. Vol. 243. P. 409–447. DOI: 10.1016/S0304-3975(99)00114-0.

11. Zubarev A. State space reduction for time Petri nets with weak semantics. Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2019. No. 43. P. 39–52. DOI: 10.31144/bncc.cs.2542-1972.2019.n43.p39-52.

12. Heiner M., Gilbert D., Donaldson R. Petri nets for systems and synthetic biology. 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008, Bertinoro, Italy, June 2–7, 2008. Vol. 5016. Berlin, Heidelberg, Springer, 2008. P. 215–264. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-68894-5_7.

13. Szawulak B., Radom M., Formanowicz P. Comparison by partition—Finding Petri nets similarity on the basis of subnets. Computers in Biology and Medicine. 2025. Vol. 187. Art. 109775. DOI: 10.1016/j.compbiomed.2025.109775.

14. Olszak J., Radom M., Formanowicz P. Some aspects of modeling and analysis of complex biological systems using time Petri nets. Bulletin of the Polish Academy of Sciences. Technical Sciences. 2018. Vol. 66, no. 1. P. 67–78. DOI: 10.24425/119060.

15. Gilin V., Laauwen S., Xia Y., *et al.* Modeling HIF-ILK interaction using continuous Petri nets. 18th Biomedical Engineering Systems and Technologies, BIOSTEC 2025, Porto, Portugal, February 20–22, 2025, Proceedings of the 18th International Joint Conference. Setúbal, SCITEPRESS, 2025. P. 537–544. DOI: 10.5220/0013102600003911.

16. Gilbert D., Heiner M. From Petri Nets to Differential Equations — An Integrative Approach for Biochemical Network Analysis. 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2006, Turku, Finland, June 26–30, 2006. Vol. 4024. Berlin, Heidelberg, Springer, 2006. P. 200–219. Lecture Notes in Computer Science. DOI: 10.1007/11767589_11.

17. Grieco L., Calzone L., Bernard-Pierrot I., *et al.* Integrative modelling of the influence of MAPK network on cancer cell fate decision. PLoS Computational Biology. 2013. Vol. 9, no. 10. DOI: 10.1371/journal.pcbi.1003286.

18. Sahin O., Frohlich H., Lobke C., *et al.* Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. BMC Systems Biology. 2009. Vol. 3, no. 1. P. 1–20. DOI: 10.1186/1752-0509-3-1.

19. Kordon F., Hulin-Hubard F., Jezequel L., *et al.* Complete results for the 2025 edition of the Model Checking Contest. 2025. URL: https://mcc.lip6.fr/2025/results.php (accessed: 14.12.2025).

20. Virbitskaite I.B., Bushin D., Best E. True concurrent equivalences in time Petri nets. Fundamenta Informaticae. 2016. Vol. 149. P. 401–418. DOI: 10.3233/FI-2016-1454.

УДК 004.42:519.766.[24+26]+343.983 DOI: 10.14529/cmse260104

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И АЛГОРИТМЫ ПОСТРОЕНИЯ ЭТОГРАММЫ ОЛЬФАКТОРНОГО ИССЛЕДОВАНИЯ ОДНОГО ОБЪЕКТА И ОДНОГО ПРОВЕРЯЕМОГО ЛИЦА

© 2026 А.А. Солдатенко, Д.В. Семенова

Сибирский федеральный университет

(660041 Красноярск, пр. Свободный, д. 79)

E-mail: ASoldatenko@sfu-kras.ru, DVSemonova@sfu-kras.ru

Поступила в редакцию: 30.12.2025

Целью исследования является разработка алгоритмов и комплекса программ составления сравнительных рядов для запаховых следов при ольфакторной экспертизе. Предложена математическая модель, основанная на теории формальных языков, для задачи построения сравнительных рядов и формирования этограммы для идентификационной ольфакторной экспертизы при проверке одного объекта и одного проверяемого лица. Этограмма — рабочая карта исследования, в которой отражаются сведения о расположении объектов сравнительных рядов и сигнальном поведении собак-детекторов. Для каждого из четырех сравнительных рядов рассматриваемой этограммы построен свой формальный язык. Формирование десятибуквенных слов каждого языка происходит согласно регламентированным требованиям, что позволяет выделить шаблоны. Шаблон определяет порядок расстановки значащих запаховых проб в каждом ряду. Математическая задача составления этограммы сводится к выбору четырех слов из соответствующих языков. Данная задача имеет комбинаторный характер с множественными ограничениями, продиктованными практической областью. Представлено описание алгоритмов и комплекса программ для решения поставленной задачи. Выполнен анализ мощности языков для каждого сравнительного ряда и проанализированы генеративные возможности алгоритма построения этограммы.

Ключевые слова: ольфакторный анализ, этограмма, формальный язык, комбинаторика, алгоритмизация.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Солдатенко А.А., Семенова Д.В. Математическая модель и алгоритмы построения этограммы ольфакторного исследования одного объекта и одного проверяемого лица // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2026. Т. 15, № 1. С. 80–95. DOI: 10.14529/cmse260104.

Введение

Органами внутренних дел России проводится множество различных видов экспертиз. Одной из них является экспертиза запаховых следов человека (ольфакторная) [1]. Данный вид экспертизы использует в качестве детектора орган обоняния специально обученной собаки [2, 3]. Как следствие, основным методом ольфакторного исследования служит известный из зоопсихологии метод выбора из множества по образцу на основе анализа рефлексии животных [4]. Ольфакторный анализ является хорошо изученным и считается высокоточным методом проведения криминалистических экспертиз [5, 6]. Одним из основных этапов проведения ольфакторного исследования является составление экспертами этограммы. К этограмме предъявляются требования, которые строго регламентированы, что гарантирует достоверность и высокую точность экспертизы. Стоит отметить, что в каждой стране перечень данных требований и особенности проведения экспертизы различны [2, 7]. В данной работе рассматриваются требования предъявляемые к составлению этограммы в Россий-

ской Федерации для проведения ольфакторной экспертизы для одного объекта и одного проверяемого лица.

Большое число публикаций по теме ольфакторного анализа, как в отечественных журналах, так и в зарубежных, посвящено исследованию вопросов юриспруденции, криминалистики [7], кинологии [4, 8] и медицины [5, 9]. Предъявляемые к построению этограммы требования порождают трудную математическую задачу размещения исследуемых объектов в нескольких рядах с перекрестными условиями, которая даже с учетом невысокой размерности на практике занимает достаточное время у экспертов на ее составление. В научных исследованиях этот аспект ольфакторного анализа опущен, что будет восполнено в настоящей работе.

Предлагается строить этограмму как слово специфического формального языка над конечным алфавитом. Слова данного языка обладают следующими свойствами: одинаковая длина слов, ограничения и запреты на внутреннюю структуру слов, определяемые требованиями к проведению экспертизы. Языки, обладающие подобными свойствами, и их комбинаторные характеристики рассматривались в следующих работах [10–17]. В настоящее время активно исследуются языки, избегающие степени, в которых накладываются ограничения на внутренние повторы элементов [13]. Другим актуальным направлением является вычисление комбинаторных характеристик формальных языков, а именно комбинаторная сложность языков, минимизация антисловаря, алгоритмизация построения [14, 15, 17].

Раздел 1 посвящен необходимым понятиям, определениям и требованиям предъявляемым к ольфакторной идентификационной экспертизе при проверке одного объекта и одного проверяемого лица. В разделе 2 предлагается математическая постановка задачи составления этограммы на основе теории формальных языков. Раздел 3 содержит описание алгоритмов для построения деревьев шаблонов для каждого сравнительного ряда, а также алгоритм заполнения шаблонов для получения итоговой этограммы, соответствующей всем предъявляемым требованиям. Обсуждение результатов работы и перспективные направления исследования представлены в разделе Заключение.

1. Об особенностях ольфакторной экспертизы

С криминалистической точки зрения основная задача в ольфакторном исследовании состоит в установлении наличия пахучих следов конкретных лиц на объектах, изымаемых по расследуемым делам [18]. Результатом ольфакторного исследования является заключение экспертов, которое содержит один из четырех выводов:

- выявлены пахучие следы проверяемого человека (категорически положительный);
- не выявлены пахучие следы проверяемого человека (категорически отрицательный);
- вероятное присутствие пахучих следов;
- не представляется возможным.

Для разграничения вероятного и категорически положительного результата используют имеющиеся вероятностно-статистические расчеты ученых Российского федерального центра судебной экспертизы при МЮ РФ [5] и Экспертно-криминалистического центра МВД России [6].

Процедура ольфакторного исследования зачастую проводится по одному алгоритму [3, 7]. Специалист проводит собаку-детектора вдоль станций сравнительного ряда из открытых емкостей для поочередного обнюхивания помещенных в них запаховых проб. При наличии

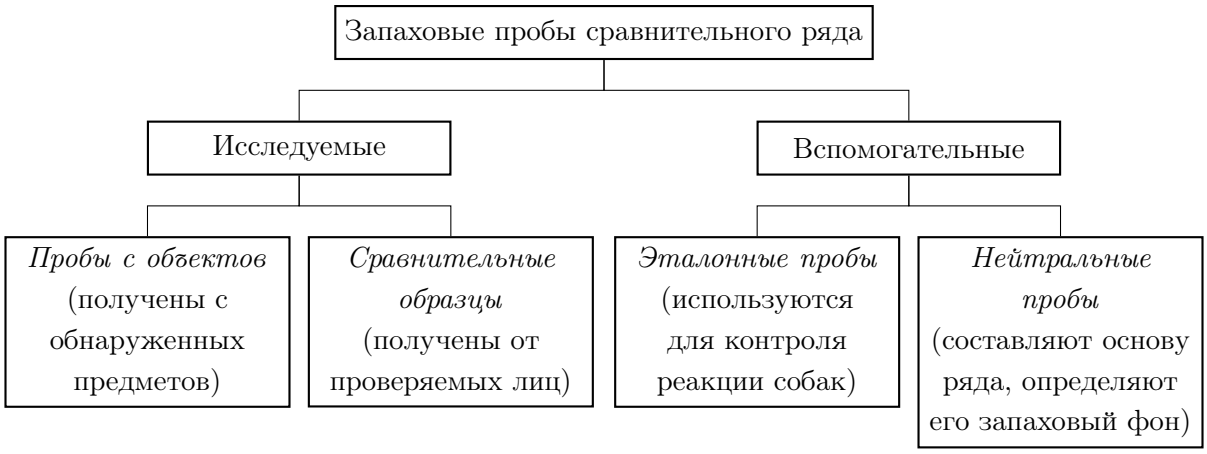


Рис. 1. Разновидности запаховых проб, используемых в сравнительных рядах, с позиции их функционального назначения

Таблица 1. Пробы, участвующие в идентификационной экспертизе

Название пробы	**Тип пробы**	**Идентификатор пробы**
Исследуемый объект	Вещь	И.
Лицо контрольной группы	Вещь	1-5
Эталон	Кровь	Э.
Нейтральный Эталон	Кровь	Н.Э.
Лицо контрольной группы	Кровь	6-7
Лицо контрольной группы (замена)	Кровь	8

среди них пробы с заданной к поиску ольфакторной характеристикой, воспринявшая этот запах собака-детектор принимает выработанную дрессировкой сигнальную позу — садится у этого объекта. Данное поведение сопровождается пищевым подкреплением собаки-детектора и фиксируется в протоколах работы (этограммах).

Предварительное составление этограммы, как рабочей карты исследования, позволяет эксперту до начала ольфакторной экспертизы продумать варианты расстановки запаховых проб. Сравнительный ряд этограммы наполняют различными пробами, каждая из которых выполняет специальную функцию [2]. Виды проб, применяемые МВД РФ при проведении экспертизы, представлены на рис. 1

Согласно регламенту для проведения идентификационной экспертизы проверки одного объекта и одного проверяемого лица требуется составить четыре различных сравнительных ряда, которые должны удовлетворять установленным требованиям [2]. Запаховые пробы подразделяются на два типа: проба, взятая с вещи; проба, взятая из крови. В данной экспертизе используется шесть проб вещей и четыре пробы крови. В табл. 1 представлены все пробы, которые могут появиться в сравнительных рядах, при этом в первых двух рядах не используется проба замены (проба номер 8), а в третьем и четвертом рядах удаляется нейтральный эталон (проба Н.Э.).

На основе многолетней экспертной практики сформулированы методические рекомендации и требования относительно процедуры идентификационного ольфакторного исследования с использованием обоняния собак-детекторов [2]. Данные требования представлены ниже в виде ограничений, которым должны удовлетворять сравнительные ряды в ольфак-

торной идентификационной экспертизе при проверке одного объекта и одного проверяемого лица. Буквой «О» обозначены обязательные требования, буквой «Р» — рекомендации.

О1 Длина проводок должна быть разной в каждом ряду.
О2 Движение должно быть в различном направлении (по часовой стрелке или против) и с различной станции в каждом ряду.
О3 Первые две проводки должны заканчиваться на станции с пробой Н.Э., а третья и четвертая проводки должны заканчиваться на станции с пробой Э.
О4 Первая проводка должна содержать пробу со схожей ольфакторной характеристикой, взятой с вещи или из крови, перед пробами И. и Э., соответственно.
О5 Все пробы, кроме пробы номер 8, должны присутствовать в первых двух проводках.
О6 Пробы не должны встречаться на тех же станциях, что и в предыдущих рядах. Все пробы расположенные на стациях вне проводки не влияют на выполнение этого требования.
О7 Пробы И., Э., Н.Э. не должны располагаться на соседних станциях.
О8 В каждой из двух первых проводок должны содержаться пробы И., Э., Н.Э.
О9 Недопустимо размещать более двух проб в третьей и четвертой проводках с одинаковой ольфакторной характеристикой, и более трех проб в первых двух проводках.
О10 Недопустимо размещать пробу, взятую из крови, перед пробой И.
О11 Недопустимо размещать пробы И., Э., Н.Э. на станциях, где собака получала пищевое подкрепление.

Дополнительно предъявляются необязательные рекомендации к сравнительным рядам.
Р1 Одинаковые пробы не должны размещаться перед пробами И., Э., Н.Э. в соседних рядах.
Р2 В третьей и четвертой проводках должно быть различное число станций между пробами И. и Э.
Р3 Проба номер 8 должна находиться в начале проводки.
Р4 Пробы И. и Э. не должны быть первыми в ряду.

Перечисленные требования накладывают ограничения не только на расположение проб внутри сравнительных рядов, но и на их размещение между рядами. Так, первый ряд определяет дополнительные ограничения на второй, второй на третий, а третий на четвертый. Данные свойства каждого требования отражены в табл. 2, где символ «+» означает влияние требования на внутреннюю структуру ряда, а символ «⋆» означает влияние требования на расположение пробы в зависимости от предыдущих рядов.

Таблица 2. Влияние ограничений на каждый ряд этограммы

	О1	О2	О3	О4	О5	О6	О7	О8	О9	О10	О11	Р1	Р2	Р3	Р4
Ряд 1			+	+			+	+	+	+					+
Ряд 2	⋆	⋆	+		⋆	⋆	+	+	+	+	⋆	⋆			+
Ряд 3	⋆	⋆	+			⋆	+		+	+	⋆	⋆		+	+
Ряд 4	⋆	⋆	+			⋆	+		+	+	⋆	⋆	⋆	+	+

Исходя из высказанных наблюдений очевидно, что задача составления сравнительных рядов для запаховых следов имеет комбинаторный характер. В качестве иллюстрирующего примера, в табл. 3 представлена этограмма, удовлетворяющая всем требованиям и ограничениям. Синим цветом выделена стартовая позиция ряда (станция), зеленым — предпола-

гаемое завершение проводки при положительной экспертизе, красным — пробы, которые не должны быть пронюханы во время проводки. Стрелками указаны стартовая позиция и направление проводки. Символ «*» означает предполагаемое место посадки собаки-детектора, буква «п» означает место пищевого подкрепления. Так, в первом ряду длина проводки составляет девять проб, а в третьем — семь проб, однако собака-детектор должна распознать связь между пробой И. и Э., поэтому выделено несколько мест посадки. Всё что находится в ряду вне границ проводки не обязано удовлетворять условиям.

Таблица 3. Пример корректной этограммы

	Позиция пробы в ряду									
Заданная проба	1	2	3	4	5	6	7	8	9	10
Н.Э.	1 →	И.	6	3	Э.	7	5	2	Н.Э. *п	4
Н.Э.	Н.Э. *п	3	И.	1	6	Э.	2	4 ←	5	7
Э.	5	2 →	6	7	4	И. *	8	Э. *п	1	3
Э.	4	2	7	И. *п	3 ←	5	1	8	6	Э. *п

2. Математическая постановка задачи составления этограммы

Для дальнейшего изложения используются классические определения и обозначения теории формальных языков [10–12].

Определим несколько множеств объектов, из которых состоят сравнительные ряды. Пусть множество $V = \{v_1, \ldots, v_5, v_0\}$ определяет запаховые пробы, взятые с вещей, где v_0 — исследуемый объект. Множество $K = \{k_6, k_7, k_8, k_e, k_n\}$ задает запаховые пробы, полученные из крови, где k_e — эталонный элемент, а k_n — нейтральный эталон. Таким образом, общий набор элементов для построения каждого сравнительного ряда определяется множеством $\Sigma = V \cup K$, при этом первые два ряда используют алфавит $\Sigma^{1,2} = \Sigma \setminus \{k_8\}$, а последние алфавит $\Sigma^{3,4} = \Sigma \setminus \{k_n\}$. Тогда языки для каждого сравнительного ряда состоят из слов, содержащих десять неповторяющихся символов над алфавитом Σ, и удовлетворяют ограничениям на внутреннюю структуру ряда.

Определим языки L_1, L_2, L_3 и L_4 для каждого сравнительного ряда [19, 20]. Процедура построения слов для каждого языка базируется на понятии шаблона сравнительного ряда. Шаблон сравнительного ряда β учитывает ограничения на внутреннее размещение символов в слове языка в рамках проводки. Для построения шаблонов каждого языка будем использовать следующее допущение: любой элемент из $\{v_1, \ldots, v_5\}$ будем обозначать как v, а элемент из $\{k_6, k_7, k_8\}$ как k, соответственно. Тогда язык шаблонов B_i для i-го сравнительного ряда определяется над алфавитом $\Sigma^\beta = \{k, v, v_0, k_e, k_n\}$ и удовлетворяет ограничениям на внутреннюю структуру ряда. Согласно положительному результату экспертизы собака-детектор выполняет обход проб от стартовой позиции до предполагаемого места посадки, при этом пробы, размещаемые в оставшейся части ряда, не будут пронюханы. Тогда слово для i-го ряда $w \in L_i$ является конкатенацией двух слов $w = \hat{\beta} \cdot \hat{\gamma}$, где $\hat{\beta}$ —

значащая часть ряда, которая определяет пробы от стартовой позиции до предполагаемого места посадки, а $\hat{\gamma}$ — символы алфавита Σ^i не вошедшие в $\hat{\beta}$, т. е. оставшиеся пробы, которые не будут пронюханы. Будем говорить, что $\hat{\beta}$ удовлетворяет шаблону $\beta \in B_i$ если для символа v в шаблоне β на соответствующей позиции в $\hat{\beta}$ расположен любой символ из $\{v_1, v_2, v_3, v_4, v_5\}$, а для символа k — любой символ из $\{k_6, k_7, k_8\}$, и записывать как $\hat{\beta} \sim \beta$. Заметим, что размер всех слов из B_i ограничен сверху и снизу. Далее, словом ζ будем обозначать последовательность символов в слове β, на которую накладывается ограничение. Пусть слово β конкатенация слов δ, η, и, возможно, ζ, где δ — префикс, а η — суффикс. Обозначение $\{z\}^m, m \in \mathrm{Z}^+$ означает последовательность из m символов z некоторого алфавита. В табл. 4 приведены ограничения, которые влияют на шаблоны рядов согласно табл. 2. В P3 не указано, что понимается под началом ряда, поэтому в рамках модели началом ряда будут считаться первые три элемента.

Определим операцию формирования языка остаточной части $\Gamma\big(\hat{\beta}, \Sigma'\big)$. Данная операция порождает язык, который содержит все слова длины $10 - |\hat{\beta}|$, состоящие из неповторяющихся символов алфавита $\Sigma' \setminus \{z\colon z \in \hat{\beta}\}$. Таким образом, словами являются все возможные перестановки символов нового алфавита.

Будем считать, что шаблон первого ряда всегда соответствует проводке, которая начинается с первой позиции и выполняет обход по часовой стрелке, т. е. выполнение требований ряда проверяется в направлении слева направо. Данное условие не противоречит общему способу построения, но значительно упрощает дальнейшее изложение. Поскольку первый ряд никак не зависит от других, то согласно табл. 4 определим язык шаблонов B_1:

$$B_1 = \{\beta\colon 6 \le |\beta| \le 10, R_1\},$$

где R_1 — набор ограничений для первого ряда, состоящий из O3 (O3 в ослабленном виде), O4, O7, O8, O9 (O9 в ослабленном виде), O10, и рекомендации P4. Таким образом, язык первого ряда можно записать следующим образом:

$$L_1 = \Big\{w\colon w = \hat{\beta}\cdot\hat{\gamma}, \hat{\beta} \sim \beta, \beta \in B_1, \hat{\gamma} \in \Gamma\big(\hat{\beta}, \Sigma^{1,2}\big)\Big\}.$$

По результатам работы алгоритма построения дерева шаблонов, который будет представлен далее в разделе 3, установлены следующие характеристики языка L_1. Число шаблонов $\beta \in B_1$ составляет 99 шаблонов, при этом число уникальных заполнений $\hat{\beta}$ символами из алфавита $\Sigma^{1,2}$ равно 12 964. С учетом произвольной последовательности символов в $\hat{\gamma}$ мощность языка L_1 составляет 510 720 слов.

Начиная со второго ряда добавляются ограничения, которые влияют не только на шаблон ряда, но и на его наполнение. Особенности того, как учитывать данные ограничения, будут рассмотрены далее. Приведем вид оставшихся языков и их характеристики.

Аналогично первому ряду, определим язык шаблонов B_2:

$$B_2 = \{\beta\colon 5 \le |\beta| \le 10, R_2\},$$

где набор ограничений R_2 состоит из O3 (O3 в ослабленном виде), O7, O8, O9 (O9 в ослабленном виде), O10, и рекомендации P4. Благодаря отсутствию ограничения O4, минимальная длина шаблона β уменьшилась до 5. Тогда язык второго ряда имеет вид

$$L_2 = \Big\{w\colon w = \hat{\beta}\cdot\hat{\gamma}, \hat{\beta} \sim \beta, \beta \in B_2, \hat{\gamma} \in \Gamma\big(\hat{\beta}, \Sigma^{1,2}\big)\Big\}. \quad (1)$$

Таблица 4. Математическое описание ограничений на шаблон ряда

Ограничение		Математическое описание
О3	Первый и второй ряды должны оканчиваться пробой Н.Э.	$\forall\beta\colon \beta \sqsupseteq k_n$
О3′	Второй и третий ряды должны оканчиваться пробой Э.	$\forall\beta\colon \beta \sqsupseteq k_e$
О4	Первый ряд должен содержать пробу со схожей ольфакторной характеристикой, перед пробами И., Э.	$\beta = \delta\cdot\{v_0\}\cdot\eta\colon \delta = \delta_1\cdot\{v\}\cdot\delta_2,$ $\beta = \delta\cdot\{k_e\}\cdot\eta\colon \delta = \delta_1\cdot\{k\}\cdot\delta_2$
О5	В первых двух рядах должны присутствовать все пробы, кроме пробы 8	$\hat\beta_1\cdot\hat\beta_2 = \langle\delta, N, \eta\rangle,$ где $N \in \Sigma^{1,2}, \hat\beta_1\cdot\hat\gamma_1 \in L_1, \hat\beta_2\cdot\hat\gamma_2 \in L_2.$
О7	Пробы И., Э., Н.Э. не должны находиться рядом	$\beta = \langle\delta,\zeta,\eta\rangle$, где $\zeta \notin \{v_0k_e, v_0k_n, k_ev_0, k_ek_n, k_nv_0, k_nv_e\}$
О8	Первые два ряда должны содержать пробы И., Э., Н.Э.	$\beta = \langle\delta,\{e\},\eta\rangle$, где $e \in \{v_0, k_e, k_n\}$
О9	Запрещено размещать четыре и более пробы с одинаковой ольфакторной характеристикой	$\zeta \neq \{z^4\}, z \in \{k, k_e, k_n\},$ $\zeta \neq \{z^4\}, z \in \{v, v_0\}$, где $\zeta\colon \beta = \delta\cdot\zeta\cdot\eta$
О9′	Запрещено размещать три и более пробы с одинаковой ольфакторной характеристикой	$\zeta \neq \{z^3\}, z \in \{k, k_e, k_n\},$ $\zeta \neq \{z^3\}, z \in z \in \{v, v_0\}$, где $\zeta\colon \beta = \delta\cdot\zeta\cdot\eta$
О10	Запрещено размещать пробу полученную из крови перед пробой И.	$\zeta \neq \{kv_0\}$, где $k \in K$, $\zeta\colon \beta = \delta\cdot\zeta\cdot\eta$
Р3	Проба номер 8 должна находиться в начале проводки.	$\beta = \delta\cdot\eta\colon k_8 \in \delta$ и $\lvert\delta\rvert = 3$
Р4	Пробы И. и Э. не должны быть первыми в ряду.	$\beta = \delta\cdot\eta, \delta \notin \{k_e, k_n, v_0\}$ и $\lvert\delta\rvert = 1$

Число шаблонов $\beta \in B_2$ составляет 103 шаблона, при этом число уникальных заполнений $\hat\beta$ символами из алфавита $\Sigma^{1,2}$ равно 13 069. Учитывая произвольную последовательность символов в γ, мощность языка L_2 составляет 870 720 слов. Это наиболее многочисленный язык, среди всех остальных.

Третий и четвертый ряды обладают одинаковым набором ограничений на шаблон ряда, и, как следствие, одинаковым набором слов в языке. Определим языки шаблонов третьего

и четвертого рядов:

$$B_{3,4} = \{\beta\colon 4 \le |\beta| \le 10, R_{3,4}\},$$

где набор ограничений $R_{3,4}$ состоит из О3, О7, О9, О10 и рекомендаций Р3, Р4. Минимальная длина шаблона β составляет 4 символа. Теперь определим языки L_3 и L_4:

$$L_{3,4} = \left\{w\colon w = \hat{\beta}\cdot\hat{\gamma}, \hat{\beta} \sim \beta, \beta \in B_{3,4}, \hat{\gamma} \in \Gamma\left(\hat{\beta}, \Sigma^{3,4}\right)\right\}. \tag{2}$$

Число шаблонов $\beta \in B_{3,4}$ составляет 35 шаблонов, при этом число уникальных заполнений $\hat{\beta}$ символами из алфавита $\Sigma^{3,4}$ равно 1 340. Учитывая произвольную последовательность символов в γ, мощность каждого языка L_3 и L_4 составляет 21 360 слов.

Сравнение всех характеристик языков $L_1, \ldots, L_4$ представлены в табл. 5

Таблица 5. Характеристики формальных языков для сравнительных рядов ольфакторной экспертизы

Номер ряда	Язык L_i	Число шаблонов $\lvert B_i\rvert$	Число уникальных заполнений $\hat{\beta}_i$	Мощность языка
1	L_1	99	12 964	510 720
2	L_2	103	13 069	870 720
3	L_3	35	1 340	21 360
4	L_4	35	1 340	21 360

На основе введенных определений можно сформулировать задачу построения сравнительных рядов ольфакторного анализа следующим образом:

Дано: алфавит Σ, языки L_1, L_2, L_3 и L_4.
Требуется: найти слова $w_1 \in L_1$, $w_2 \in L_2$, $w_3 \in L_3$ и $w_4 \in L_4$, удовлетворяющие требованиям О1–О11 и рекомендациям Р1–Р4.

Для решения данной задачи предлагается ряд алгоритмов, представленных в следующем разделе.

3. Алгоритмы построения сравнительных рядов ольфакторного анализа

3.1. Алгоритм построения дерева шаблонов

В предыдущем разделе упоминался алгоритм построения дерева шаблонов для каждого сравнительного ряда. В данном случае под деревом понимается ориентированный граф $G = (V, E)$, где V — множество вершин, а E — упорядоченное множество пар вершин $e = (x, y)$, где $x, y \in V$. Зададим функцию $\varphi\colon E \to \{k_e, k_n, v_0, k, v\}$. Пусть функция φ ставит каждому ребру специальную метку, которая определяет тип запаховой пробы. Таким образом, граф формируется следующим образом. Из корневой вершины исходят все ребра e с метками проб, которые могут стоять на первой позиции соответствующего ряда. Так для первого ряда это будут два ребра с метками v и k. Продолжая спускаться вглубь дерева, формируется шаблон ряда, при этом ребро, которое приводит в листовую вершину содержит метку k_n для первых двух рядов и метку k_e для третьего и четвертого рядов.

Основная идея алгоритма заключается в рекурсивном построении дерева шаблонов, путем проверки всех ограничений на шаблон ряда на каждом шаге. Так рекурсивный вызов происходит для каждой метки, которая не противоречит ограничению:

- проба k может быть выставлена, если
  - текущая последовательность проб, взятых из крови, меньше трех (двух для заключительных рядов);
  - количество проб k меньше двух в ряду (трех для заключительных);
- проба v может быть выставлена, если
  - текущая последовательность проб, взятых с вещей, меньше трех (двух для заключительных рядов);
  - количество проб v меньше пяти в ряду;
- проба v_0 может быть выставлена, если
  - текущая последовательность проб, взятых с вещей, меньше трех (двух для заключительных рядов);
  - прошлая проба была взята не из крови;
  - проба v_0 не представлена в ряде;
  - ранее была выставлена проба, взятая с вещи (для первого ряда);
- проба k_e может быть выставлена, если
  - текущая последовательность проб, взятых из крови, меньше трех (двух для заключительных рядов);
  - ранее была выставлена проба, взятая из крови (для первого ряда);
  - прошлая проба не является v_0 или k_n;
  - проба k_e не представлена в ряде;
- проба k_n может быть выставлена, если
  - текущая последовательность проб, взятых из крови, меньше трех (двух для заключительных рядов);
  - ранее была выставлена проба, взятая из крови (для первого ряда);
  - прошлая проба не является v_0 или k_e;
  - в ряде представлены v_0 и k_e (для первого и второго ряда);
  - проба k_n не представлена в ряде.

Общее правило для всех проб — длина ряда не превосходит 10 проб, что гарантирует завершение рекурсивного алгоритма построения. При этом, после установки пробы k_n текущая ветвь рекурсии всегда заканчивается. В алгоритме может возникнуть ситуация, когда невозможно выставить пробу, например, для последовательности $vv_0vkvvkvk_e$. В данном примере остается проба k_n, но она не должна соседствовать с k_e. В этом случае рекурсия также заканчивает свою работу, однако в листовую вершину ведет ребро не с меткой k_n. Поэтому используются процедура очистки висячих вершин, в которые не ведет ребро с меткой k_n для первых двух рядов и меткой k_e в заключительных рядах.

Суть процедуры очистки заключается в определении списка висячих вершин, в которые ведет ребро с некорректной меткой и их удалении. Алгоритм повторяется до тех пор, пока во все висячие вершины не будут вести ребра с заключительной меткой соответствующей ряду.

После завершения работы алгоритма формируется дерево шаблонов, которое используется далее для формирования ряда. Стоит отметить, что для экономии занимаемой памяти все листовые вершины дерева шаблонов можно заменить одной вершиной, в которую ве-

дут все ребра с заключительной меткой, получив при этом ориентированный ациклический граф. Примеры получаемых графов представлены в работах [19–21].

3.2. Алгоритм построения сравнительных рядов

Как было сказано ранее, каждый сравнительный ряд состоит из проводки и оставшейся части. Проводка формируется на основе шаблона ряда, который получается из дерева шаблонов, описанного выше. Шаблон формируется при обходе дерева шаблонов от корневой вершины до листовой. Каждый переход по ребру добавляет метку к шаблону ряда. Согласно выбранному шаблону, проводка заполняется конкретными пробами, учитывая ограничения [O1] [O2] [O5] [O6] [O11] и рекомендации [P1] [P2]. Оставшиеся пробы могут быть размещены в случайном порядке в окончании ряда, поскольку при положительном результате экспертизы они не влияют на результат. Если k — количество оставшихся символов, то в алгоритме для построения оставшейся части равномерно выбирается одна из возможных перестановок этих символов с вероятностью $1/k!$.

Учитывая особенности сравнительных рядов, описанных ранее, в алгоритме полагается, что движение в нечетных рядах совершается вправо, а в четных — влево. Первый ряд всегда начинается с первой позиции, последующие ряды начинаются с произвольной позиции, отличной от предыдущих рядов. Очевидно, что все ряды в совокупности можно линейным образом сдвигать. В процессе построение рядов вычисляются индексы, учитывающие относительные сдвиги между положением проб, которые позволяют проверять ограничения [O1] [O2] [O5] [O6] [O11] и рекомендации [P1] [P2].

Стоит отметить, что при определенном выборе шаблонов и их заполнении может возникнуть ситуация, при которой невозможно удовлетворить всем ограничениям. В этом случае алгоритм отказывается от текущих выбранных рядов и начинает генерацию с первого ряда.

4. Анализ работы программного комплекса

Разработанные алгоритмы представлены в виде комплекса программ. Комплекс программ реализован на языке `C++` стандарта `C++17` и графической библиотеки `Dear ImGui` с применением средств автоматизации `CMake`. В ходе работы комплекса программ выполняется построение четырех деревьев шаблонов для каждого ряда L_1–L_4, на основе которых формируется готовая этограмма исследования. Далее программа строит таблицу с шаблонами, где каждому ряду ставится в соответствие один шаблон, выбранный случайным образом из дерева. На основе данной таблицы происходит конструктивное заполнение шаблонов и их слияние в готовую этограмму согласно алгоритму из подраздела 3.2. Таким образом, результатом работы комплекса являются две таблицы: таблица, содержащая один шаблон для каждого из четырех рядов, и построенная этограмма исследования. Комплекс программ работает исключительно для рядов длины 10, поскольку это регламентировано порядком проведения экспертизы. Полученные деревья шаблонов, коды для визуализации и оценки размерностей представлены в открытом доступе [21]. Далее приводятся несколько экспериментов по заполнению этограммы.

Таблицы 6 и 7 представляют результат работы комплекса программ. В табл. 6 указаны выбранные шаблоны для каждого ряда, а в табл. 7 — заполненные ряды согласно выбранным шаблонам. Цветами в таблицах обозначены старт и окончание проводки, а также неиспользуемые в проводке станции, аналогично табл. 3. Символом «⊘» обозначены

позиции, которые не используются в шаблоне, т. е. фактически принадлежащие слову γ из конкатенации $\alpha = \beta \cdot \gamma$.

Таблица 6. Шаблоны для каждого сравнительного ряда

	Позиция ряда									
	1	2	3	4	5	6	7	8	9	10
Ряд 1	k $\rightarrow$	v	k	v	v_0	v	k_e	v	v	k_n
Ряд 2	v	k_e	v	v $\leftarrow$	$\oslash$	k_n	k	v_0	v	k
Ряд 3	$\oslash$	$\oslash$	v $\rightarrow$	v	k	v	v_0	k	k_e	$\oslash$
Ряд 4	k_e	v	k	v_0	v	k	v	v $\leftarrow$	$\oslash$	$\oslash$

Таблица 7. Заполненные сравнительные ряды

	Позиция ряда									
	1	2	3	4	5	6	7	8	9	10
Ряд 1	k_6 $\rightarrow$	v_1	k_7	v_4	v_0	v_2	k_e	v_3	v_5	k_n
Ряд 2	v_5	k_e	v_4	v_2 $\leftarrow$	v_3	k_n	k_6	v_0	v_1	k_7
Ряд 3	v_1	v_2	v_3 $\rightarrow$	v_5	k_7	v_4	v_0	k_8	k_e	k_6
Ряд 4	k_e	v_5	k_7	v_0	v_1	k_6	v_2	v_3 $\leftarrow$	v_4	k_8

В этом примере получаются следующие шаблоны для каждого ряда: шаблон первого ряда имеет вид $\beta_1 = kvkvv_0vk_evvk_n$, для второго ряда $\beta_2 = vvk_evkvv_0kk_n$, для третьего ряда $\beta_3 = vvkvv_0kk_e$, и для заключительного четвертого ряда $\beta_4 = vvkvv_0kvk_e$.

Также рассмотрим пример, когда невозможно построить шаблон четвертого ряда. В табл. 8 представлены шаблоны первых трех рядов. При таких шаблонах не существует ни одного такого шаблона для четвертого ряда, который мог бы удовлетворить ограничениям О3 О7 О9 О10 и рекомендациям Р3 Р4

Заключение

В работе предложен конструктивный алгоритм формирования этограммы для идентификационной ольфакторной экспертизы при проверке одного объекта и одного проверяемого лица. Постановка задачи построения этограммы представлена в виде математической модели, использующей понятия и обозначения теории формальных языков. Решение задачи можно свести к пяти этапам. Первые четыре этапа связаны с построением новых языков, соответствующих четырем сравнительным рядам этограммы, над алфавитом проб Σ. Построенные языки состоят из слов, удовлетворяющих ограничениям О1–О11 и рекомендациям Р1–Р4. Также были разработаны алгоритмы построения всех возможных шаблонов

Таблица 8. Пример шаблонов первых трех рядов, при которых не существует ни одного шаблона для четвертого ряда, удовлетворяющего ограничениям

	Позиция ряда									
	1	2	3	4	5	6	7	8	9	10
Ряд 1	k $\rightarrow$	v	v	k	v	k_e	v	v_0	v	k_n
Ряд 2	$\oslash$	k_n	k	v_0	v	k	k_e	v	v	v $\leftarrow$
Ряд 3	k_e	$\oslash$	$\oslash$	$\oslash$	$\oslash$	v $\rightarrow$	k	v	v_0	k

для каждого из четырех сравнительных рядов и выполнен анализ мощности полученных языков. На пятом этапе непосредственно формируется этограмма. Разработан алгоритм, который позволяет строить все возможные слова языков для каждого ряда и их комбинацию, удовлетворяющую всем требованиям. Проанализированы генеративные возможности алгоритма построения этограммы. Следует заметить, что конкатенация четырех языков, соответствующих рядам этограммы, не позволяет построить общий язык этограмм. Это связно с необходимостью учета сдвигов и произвольности начала и направления проводки.

Интерес представляет решение задачи построения языка этограмм средствами смежных разделов теории формальных языков — языков, избегающих степени (с дополнительными ограничениями), и универсальных слов (последовательностей де Брюина) [13, 17]. Также актуален вопрос анализа комбинаторных характеристик и сложности полученной модели. Дальнейшие направления исследований в первую очередь связаны с построением общего языка этограммы, а также разработкой алгоритма верификации произвольной этограммы; алгоритма, присваивающего каждой допустимой этограмме уникальный идентификатор, основанный на ее структурных свойствах, что позволит создать библиотеку готовых этограмм для криминалистических лабораторий.

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ (Соглашение № 075-02-2026-1314).

Авторы выражают благодарность за постановку задачи, консультации и помощь в исследовании заместителю начальника отдела ЭКЦ ГУ МВД России по Красноярскому краю подполковнику Коршуновой Наталье Васильевне.

Авторы благодарят рецензентов за их время и ценные рекомендации, которые позволили значительно улучшить первоначальную версию статьи и стимулировали дальнейшие исследования.

Литература

1. Министерство внутренних дел Российской Федерации (МВД России) Приказ МВД России от 29.06.2005 № 511 «Вопросы организации производства судебных экспертиз в экспертно-криминалистических подразделениях органов внутренних дел Российской Федерации», зарегистрирован в Минюсте России 15.08.2005 № 6903 / Онлайн-версия СПС «КонсультантПлюс». URL: https://www.consultant.ru/document/cons_doc_LAW_55315/ (дата обращения: 18.12.2025).

2. Сулимов К.Т. Криминалистическая одорология : кинологический подход. Москва: Институт Наследия, 2019. 312 с.

3. Образцов В.А. Криминалистика: учебник. Москва: Юристъ, 1997. 760 с.

4. Зорина З.А., Полетаева И.И. Зоопсихология. Элементарное мышление животных: Учебное пособие. Москва: Аспект Пресс, 2002. 102 с.

5. Собко Г.М. Вероятностно-статистическое обоснование достоверности одорологической идентификации // Вопросы теории судебной экспертизы. Сборник науч. трудов. 1977. № 31. С. 142–177.

6. Панфилов П.Б. Вероятностно-статистическое обоснование достоверности ольфакторных исследований запаховых следов человека в судебной экспертизе // Юридические науки. 2006. Т. 17, № 1. С. 172–186.

7. Ferry B., Ensminger J.J., Schoon A., *et al.* Scent lineups compared across eleven countries: Looking for the future of a controversial forensic technique // Forensic Science International. 2019. Vol. 302. P. 109895. DOI: 10.1016/j.forsciint.2019.109895.

8. Пименов М.Г., Саламатин А.В., Панфилов П.Б. и др. Подготовка собак-детекторов пахучих следов человека: Методические рекомендации. Москва: ЭКЦ МВД России, 2006. 56 С.

9. Панфилов П.Б. Биодетекция и интерпретация поведения собак-детекторов в индентификационном ольфакторном исследовании // Судебная экспертиза. Научно-практический журнал. Волгоград, 2008. Т. 15, № 3. С. 104–112.

10. Ахо А., Ульман Д. Теория синтаксического анализа, перевода и компиляции. Т. 1. Москва: Мир, 1978. 612 С.

11. Lothaire M. Algebraic Combinatorics on Words. Cambridge University Press, 2002. 455 p. Encyclopedia of Mathematics and its Applications. DOI: 10.1017/CBO9781107326019.

12. Substitutions in Dynamics, Arithmetics and Combinatorics. Vol. 1794 / ed. by N.P. Fogg, V. Berthé, S. Ferenczi, *et al.* Springer Berlin, Heidelberg, 2002. 402 p. Lecture Notes in Mathematics. DOI: 10.1007/b13861.

13. Petrova E.A., Shur A.M. Branching Densities of Cube-Free and Square-Free Words // Algorithms. 2021. Vol. 14, no. 4. P. 1–19. DOI: 10.3390/a14040126.

14. Шур А.М. Комбинаторная сложность рациональных языков // Дискретный анализ и исследование операций. Серия 1. 2005. Т. 12, № 2. С. 78–99.

15. Белаззуги Д., Колпаков Р., Раффино М. Индексация и выявление символьных множеств в одномерных и двумерных словах // Фундаментальная и прикладная математика. 2015. Т. 20, № 6. С. 3–16.

16. MacFie A. Enumerative properties of restricted words and compositions. Carleton University, 2019. 121 p. DOI: 10.22215/etd/2019-13532.

17. Chen H.Z., Kitaev S., Mütze T., Sun B.Y. On universal partial words // The European Conference on Combinatorics, Graph Theory and Applications, EUROCOMB'17, Vienna, August 28 – September 1, 2017. Vol. 61. 2017. P. 231–237. DOI: 10.1016/j.endm.2017.06.043.

18. Старовойтов В.И. Судебная ольфакторная экспертиза следов пахнущих веществ из крови и пота человека // Законы России: опыт, анализ, практика. Москва, 2011. № 12. С. 70–79.

19. Soldatenko A., Semenova D., Goldenok E., *et al.* Research on the Problem of Constructing Scent Lineups of Olfactory Analysis // 2023 IEEE 17th International Conference on Application of Information and Communication Technologies, AICT, Baku, October 18–20, 2023). IEEE, 2023. P. 1–4. DOI: 10.1109/AICT59525.2023.10313149.

20. Солдатенко А.А., Семенова Д.В. Построение сравнительных рядов для задачи ольфакторного анализа // Информационные технологии и математическое моделирование (ИТММ-2024): материалы XXIII Международной конференции имени А.Ф. Терпугова, Томск, 20–26 октября 2024. Томск: Национальный исследовательский Томский государственный университет, 2024. С. 263–268.

21. Glincka/dogtree-data Деревья шаблонов для задачи составления этограммы в Российской Федерации для проведения ольфакторной экспертизы для одного объекта и одного проверяемого лица. URL: https://github.com/Glincka/dogtree-data (дата обращения: 02.04.2026).

Солдатенко Александр Александрович, к.ф.-м.н., кафедра высшей и прикладной математики, Сибирский федеральный университет (Красноярск, Российская Федерация)

Семенова Дарья Владиславовна, к.ф.-м.н., доцент, кафедра высшей и прикладной математики, Сибирский федеральный университет (Красноярск, Российская Федерация)

DOI: 10.14529/cmse260104

MATHEMATICAL MODELING AND ALGORITHMS FOR PROBLEM OF CONSTRUCTING SCENT LINEUPS OF OLFACTORY ANALYSIS FOR SINGLE OBJECT AND SINGLE TEST SUBJECT

© 2026 A.A. Soldatenko, D.V. Semenova

Siberian Federal University (pr. Svobodny 79, Krasnoyarsk, 660041 Russia)

E-mail: ASoldatenko@sfu-kras.ru, DVSemonova@sfu-kras.ru

Received: 30.12.2025

The aim of this study is to develop algorithms and a software package for generating scent lineups for odor traces in olfactory forensic examination. A mathematical model based on formal language theory is proposed for the problem of constructing scent lineups and forming an ethogram for identification-oriented olfactory examination involving a single object and a single test subject. An ethogram serves as a working research map, documenting the placement of objects in lineups, the deployment, and the signaling behavior of detector dogs. For each of the four comparative sequences within the ethogram of the examination under consideration, a distinct formal language is constructed. The generation of ten-letter words for each language follows standardized requirements, which allowed for the identification of specific templates. A template defines the order of placement for meaningful odor samples within each lineup. Thus, the mathematical problem of ethogram compilation is reduced to selecting four words from their corresponding languages. This problem is combinatorial nature, subject to multiple constraints dictated by the practical field. The paper presents a description of the algorithms and the software package developed to solve the stated problem. An analysis of the language cardinality for each lineup is performed, and the generative capabilities of the ethogram construction algorithm are examined.

Keywords: olfactory analysis, ethogram, formal language, combinatorics, algorithmization.

FOR CITATION

Soldatenko A.A., Semenova D.V. Mathematical Modeling and Algorithms for Problem of Constructing Scent Lineups of Olfactory Analysis for Single Object and Single Test Subject. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2026. Vol. 15, no. 1. P. 80–95. (in Russian) DOI: 10.14529/cmse260104.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Rossiyskoy Federatsii (MVD Rossii) M. vnutrennikh del Prikaz MVD Rossii ot 29.06.2005 No. 511 "Voprosy organizatsii proizvodstva sudebnykh ekspertiz v ekspertno-kriminalisticheskikh podrazdeleniyakh organov vnutrennikh del Rossiyskoy Federatsii", Registered with the Ministry of Justice of Russia on 15.08.2005 No. 6903. / Online version of the legal reference system "ConsultantPlus". URL: https://www.consultant.ru/document/cons_doc_LAW_55315/ (accessed: 18.12.2025) (in Russian).
2. Sulimov K.T. Kriminalisticheskaya odorologiya: kinologicheskij podhod. Moscow: Russian Heritage Institute, 2019. 312 p. (in Russian).
3. Obrazcov V.A. Kriminalistika: uchebnik. Moscow: YUrist, 1997. 760 p. (in Russian).
4. Zorina Z.A., Poletaeva I.I. Zoopsihologiya. Elementarnoe myshlenie zhivotnyh: Uchebnoe posobie. Moscow: Aspect Press Ltd, 2002. 102 p. (in Russian).
5. Sobko G.M. Veroyatnostno-statisticheskoe obosnovanie dostovernosti odorologicheskoj identifikacii. Voprosy teorii sudebnoj ekspertizy. Sbornik nauch. trudov. 1977. No. 31. P. 142–177. (in Russian).
6. Panfilov P.B. Veroyatnostno-statisticheskoe obosnovanie dostovernosti ol'faktornyh issledovanij zapahovyh sledov cheloveka v sudebnoj ekspertize. YUridicheskie nauki. 2006. Vol. 17, no. 1. P. 172–186. (in Russian).
7. Ferry B., Ensminger J.J., Schoon A., *et al.* Scent lineups compared across eleven countries: Looking for the future of a controversial forensic technique. Forensic Science International. 2019. Vol. 302. P. 109895. DOI: 10.1016/j.forsciint.2019.109895.
8. Pimenov M.G., Salamatin A.V., Panfilov P.B., *et al.* Podgotovka sobak-detektorov pahuchih sledov cheloveka: Metodicheskie rekomendacii. Moscow: EKC MVD Rossii, 2006. 56 p. (in Russian).
9. Panfilov P.B. Biodetekciya i interpretaciya povedeniya sobak-detektorov v indentifikacionnom ol'faktornom issledovanii. Forensic Examination. Volgograd, 2008. Vol. 15, no. 3. P. 104–112. (in Russian).
10. Aho A.V., Ullman J.D. The theory of parsing, translation, and compiling. Vol. 1. NJ: Prentice-Hall Englewood Cliffs, 1972. 560 p.
11. Lothaire M. Algebraic Combinatorics on Words. Cambridge University Press, 2002. 455 p. Encyclopedia of Mathematics and its Applications. DOI: 10.1017/CBO9781107326019.

12. Substitutions in Dynamics, Arithmetics and Combinatorics. Vol. 1794 / ed. by N.P. Fogg, V. Berthé, S. Ferenczi, *et al.* Springer Berlin, Heidelberg, 2002. 402 p. Lecture Notes in Mathematics. DOI: 10.1007/b13861.

13. Petrova E.A., Shur A.M. Branching Densities of Cube-Free and Square-Free Words. Algorithms. 2021. Vol. 14, no. 4. P. 1–19. DOI: 10.3390/a14040126.

14. Shur A.M. Kombinatornaya slozhnost' ratsional'nykh yazykov [Combinatorial complexity of regular languages]. Diskretnyi Analiz i Issledovanie Operatsii. Seriya 1. 2005. Vol. 12, no. 2. P. 78–99. (in Russian).

15. Belazzougi J., Kolpakov R., Raffinot M. Indexing and Detecting Symbol Sets in One-Dimensional and Two-Dimensional Words. Journal of Mathematical Sciences. 2018. Vol. 233, no. 1. P. 1–9. DOI: 10.1007/s10958-018-3921-y.

16. MacFie A. Enumerative properties of restricted words and compositions. Carleton University, 2019. 121 p. DOI: 10.22215/etd/2019-13532.

17. Chen H.Z., Kitaev S., Mütze T., Sun B.Y. On universal partial words. The European Conference on Combinatorics, Graph Theory and Applications, EUROCOMB'17, Vienna, August 28 – September 1, 2017. Vol. 61. 2017. P. 231–237. DOI: 10.1016/j.endm.2017.06.043.

18. Starovoytov V.I. Forensic Olfactory Expertise Traces of Smelling Substances from the Person Blood and Sweat. Zakony Rossii: opyt, analiz. praktika. Moscow, 2011. No. 12. P. 70–79. (in Russian).

19. Soldatenko A., Semenova D., Goldenok E., *et al.* Research on the Problem of Constructing Scent Lineups of Olfactory Analysis. 2023 IEEE 17th International Conference on Application of Information and Communication Technologies, AICT. Baku, October 18–20, 2023. IEEE, 2023. P. 1–4. DOI: 10.1109/AICT59525.2023.10313149.

20. Soldatenko A.A., Semenova D.V. Postroenie sravnitel'nyh ryadov dlya zadachi ol'faktornogo analiza. Informational Technologies and Mathematical Modelling (ITMM-2024): Proceedings of the 23rd International Conference named after A.F. Terpugov, Tomsk, October 20–26, 2024. Tomsk: Tomsk State University Publishing, 2024. P. 263–268. (in Russian).

21. Glincka/dogtree-data Templates trees for problem of constructing scent lineups of olfactory analysis for single object and single test subject. URL: https://github.com/Glincka/dogtree-data (accessed: 02.04.2026).

УДК 004.021, 519.116 DOI: 10.14529/cmse260105

СООТНОШЕНИЯ МЕЖДУ КЛАССАМИ РАЗБИЕНИЙ С ОГРАНИЧЕНИЕМ НА ЧИСЛО ЧАСТЕЙ

© 2026 А.А. Самойлов

Южно-Уральский государственный университет
(454080 Челябинск, пр. им. В.И. Ленина, д. 76)
E-mail: samoilovaa@susu.ru

Поступила в редакцию: 27.01.2026

Работа посвящена исследованию связей между классами разбиений натурального числа n с ограничениями на число частей. Основное внимание уделяется разбиениям на ровно k попарно различных нечетных слагаемых. Для установления связи данного класса разбиений с классом разбиений с ограничением на величину частей в работе применяется операция сопряжения диаграмм Юнга. Это позволяет построить явную биекцию и доказать, что количество таких разбиений $r(n,k)$ в точности совпадает с $P((n-k^2)/2,k)$ для числа $n \equiv k \pmod 2$ и $n \geq k^2$, части которых не превосходят k. На основе полученного соотношения, проводится сравнение асимптотик функций разбиений $r(n,k)$, $p(n,k)$ — количество разбиений числа n на ровно k частей, и $q(n,k)$ — количество разбиений числа n на k различных частей. При $k = O(n^{1/3})$ доказано, что отношения $q(n,k)/r(n,k)$ и $p(n,k)/r(n,k)$ асимптотически стремятся к 2^{k-1} при $n \to \infty$. Для численной верификации полученных теорем реализованы два алгоритма на основе динамического программирования. Первый алгоритм основан на рекуррентном соотношении для подсчета разбиений на различные нечетные части $r(n,k)$, второй — на доказанной биекции и рекуррентной формуле для числа разбиений $P(m,k)$, части которых не превосходят k. Вычислительные эксперименты показывают, что алгоритм, использующий установленную биекцию, обеспечивает существенный выигрыш в производительности при увеличении n. Разработанные алгоритмы также применены для проверки гипотезы Каргаполова о рангах группы центральных единиц целочисленного группового кольца знакопеременной группы A_n, связанных с разбиениями на различные нечетные части при дополнительных ограничениях. Вычисления, проведенные до $n = 100000$, подтверждают асимптотическое поведения величин с относительной погрешностью менее 0.1%.

Ключевые слова: разбиения натуральных чисел, разбиение на различные нечетные части, комбинаторная биекция, асимптотический анализ, динамическое программирование.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Самойлов А.А. Соотношения между классами разбиений с ограничением на число частей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2026. Т. 15, № 1. С. 96–109. DOI: 10.14529/cmse260105.

Введение

Нахождение количества разбиений натурального числа n — классическая задача комбинаторики и теории чисел. Разбиения находят практическое применение не только в этих областях, но и служат фундаментальным инструментом, в частности, при описании неприводимых представлений симметрических и знакопеременных групп, а также в алгебраической комбинаторике при изучении симметрических функций [1, 2].

Центральной задачей теории разбиений, имеющей фундаментальное значение как для чистой комбинаторики, так и для приложений, является перечислительная проблема: для заданного класса разбиений найти количество всех разбиений числа n, удовлетворяющих определенным условиям. Наиболее изученным объектом является функция разбиения $p(n)$. Она определяется как количество всех разбиений натурального числа n (т.е. представлений n в виде суммы положительных целых чисел, где порядок слагаемых не учитывается).

В данной работе исследуются разбиения натуральных чисел на ровно k попарно различных нечетных слагаемых. Этот класс разбиений лежит на пересечении двух классических классов — разбиений на различные части и разбиений на нечетные части, которые связаны тождеством Эйлера $p_{\text{distinct}}(n) = p_{\text{odd}}(n)$, и характеризуется рядом свойств. Наиболее известным из них является комбинаторное тождество, устанавливающее, что количество разбиений числа n на различные нечетные части (без ограничений на их число k) равно количеству самосопряженных разбиений числа n.

Практическая значимость разбиений на различные нечетные части подтверждается их приложениями в теории представлений симметрической группы через подсчет крюков в самосопряженных разбиениях [3, 4], в аналитической теории чисел при выводе формул для чисел Гурвица и модулярных форм [5], а также в рамках единого подхода, основанного на связи двухстрочных матриц и решетчатых путей [6]. Таким образом, рассматриваемый объект служит связующим звеном между комбинаторикой, теорией чисел и теорией представлений.

Ранее автор исследовал вычислительные аспекты, связанные с задачей вычисления ранга группы центральных единиц целочисленного группового кольца знакопеременной группы A_n, требующей значительных вычислительных ресурсов. В работе [7] был предложен параллельный алгоритм в общей памяти с использованием вложенного параллелизма, что позволило ускорить расчеты для небольших значений n. В настоящей статье на основе разработанных здесь алгоритмов динамического программирования мы проверяем гипотезы, сформулированные в [8], для существенно больших значений n, что ранее было недоступно из-за вычислительной сложности.

Статья организована следующим образом. Раздел 1 содержит формальные определения и обозначения. Формулировки и доказательства основных результатов приведены в разделах 2 и 3. Раздел 4 описывает алгоритмы подсчета на основе динамического программирования. В разделе 5 приведены результаты вычислительных экспериментов и верификации полученных теорем. Раздел 6 содержит проверку гипотез о рангах. Заключение посвящено сводке результатов.

1. Формальные определения и обозначения

Разбиение натурального числа n — представление n в виде суммы невозрастающей последовательности натуральных чисел, то есть:

$$n = a_1 + a_2 + \cdots + a_k, \quad a_1 \geq a_2 \geq \cdots \geq a_k \geq 1.$$

Числа a_i называются *частями*, k — *длиной* разбиения. Следует отметить, что часто рассматриваются неупорядоченные разбиения. Тогда считается, что разбиения равны или эквивалентны, если упорядоченные наборы их частей совпадают с точностью до перестановки.

Различное нечетное разбиение числа n — разбиение с попарно различными нечетными частями:

$$n = (2m_1 - 1) + (2m_2 - 1) + \cdots + (2m_k - 1), \quad m_1 > m_2 > \cdots > m_k \geq 1.$$

Так как для любого (необязательно различного) разбиения $n = 2(m_1 + m_2 + \cdots + m_k) - k$ или, что эквивалентно, $n + k = 2(m_1 + m_2 + \cdots + m_k)$, то n и k одинаковой четности, то есть $n \equiv k \pmod 2$. Это будет часто использоваться в дальнейшем.

Введем обозначения для некоторых функций разбиений чисел.

1. $r(n,k)$ — количество различных нечетных разбиений n длины k, здесь $n \equiv k \pmod 2$.
2. $p_{\text{odd}}(n,k)$ — количество разбиений n на ровно k нечетных слагаемых (с возможными повторениями), снова $n \equiv k \pmod 2$.
3. $q(n,k)$ — количество разбиений n на ровно k различных частей.
4. $p(n,k)$ — количество разбиений n на ровно k частей (с возможными повторениями).
5. $P(n,k)$ — количество разбиений n с частями, не превышающими k:

$$n = a_1 + \cdots + a_m, \quad k \geq a_1 \geq \cdots \geq a_m \geq 1.$$

Между этими функциями справедливы следующие комбинаторные тождества:

$$p(n,k) = P(n-k,k), \tag{1}$$

$$q(n,k) = P\left(n - \binom{k+1}{2}, k\right). \tag{2}$$

Эти соотношения отражают известные биекции между классами разбиений и позволяют сводить одни задачи к другим. Например, их использовал Секереш в своей работе [9].

Важно отметить, что известное тождество Эйлера $p_{\text{distinct}}(n) = p_{\text{odd}}(n)$, верное для общего числа разбиений без учета длины k, не выполняется для функций с фиксированным числом частей: $q(n,k) \neq p_{\text{odd}}(n,k)$. Однако суммирование по всем допустимым длинам k восстанавливает тождество:

$$\sum_{k=1}^{n} q(n,k) = \sum_{\substack{1 \leq k \leq n \\ n \equiv k \pmod 2}} p_{\text{odd}}(n,k).$$

Наконец, для $k = O(n^{1/3})$ справедлива следующая асимптотическая оценка [10]:

$$P(n,k) = \frac{1}{k!}\binom{n-1}{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right). \tag{3}$$

Рассмотрим известные подходы к нахождению количества разбиений на различные нечетные части. Первым из них является аналитический метод. Для оценки числа таких разбиений для больших n применяется следующая асимптотическая формула [8] при $n \to \infty$:

$$r(n) \sim \frac{1}{2 \cdot (24n^3)^{1/4}} \exp\left(R\sqrt{n}\right), \quad \text{где } R = \frac{\pi}{\sqrt{6}}.$$

Данное соотношение дает приближенное значение и не является точным.

Вторым подходом является рекурсивный алгоритм, основанный на следующих двух операциях над разбиениями на различные нечетные части:

1. Добавление 2 к каждой части разбиения.
2. Добавление 2 к каждой части разбиения и введение нового дополнительного слагаемого.

Рассматривая обратные операции (удаление точек вместо добавления), получаем следующее рекуррентное соотношение:

$$r(n,k) = \begin{cases} 1, & \text{при } n = 1, k = 1, \\ 0, & \text{при } n < k, \\ r(n-2k,k) + r(n-(2k-1),k-1), & \text{иначе.} \end{cases} \tag{4}$$

Алгоритм 1. NrRecDistinctOdd (in n, k; out result)

```
1: if n < k или n < 1 или k < 1 then
2:     return 0
3: end if
4: if n = 1 и k = 1 then
5:     return 1
6: end if
7: return NrRecDistinctOdd(n − 2k, k) + NrRecDistinctOdd(n − (2k − 1), k − 1)
```

Алгоритм 1 является простым способом для вычисления количества разбиений на различные нечетные части. Тогда число таких разбиений можно получить путем суммирования по всем допустимым k:

$$r(n) = \sum_{\substack{k=1 \\ n \equiv k \pmod 2}}^{n} \text{NrRecDistinctOdd}(n, k).$$

2. Биекция между нечетными и ограниченными разбиениями

Теорема 1. Пусть n и k — натуральные числа, такие, что $n \equiv k \pmod 2$ и $n \geq k^2$. Тогда справедливо следующее равенство:

$$r(n, k) = P\left(\frac{n - k^2}{2}, k\right). \tag{5}$$

Доказательство. Рассмотрим произвольное различное нечетное разбиение числа n на k частей. Поскольку все части нечетны и попарно различны, его можно записать в виде:

$$n = (2m_1 - 1) + (2m_2 - 1) + \cdots + (2m_k - 1),$$

где $m_1 > m_2 > \cdots > m_k \geq 1$ — строго убывающая последовательность натуральных чисел.

Упростим выражение:

$$n = 2\sum_{i=1}^{k} m_i - k \quad \Longrightarrow \quad \sum_{i=1}^{k} m_i = \frac{n + k}{2}. \tag{6}$$

При $n = k^2$ существует единственное минимальное различное нечетное разбиение числа n длины k, достигаемое при значениях:

$$m_i^{\min} = k - i + 1, \quad i = 1, 2, \ldots, k.$$

Сумма $m_i^{\min}$ в этом минимальном случае равна:

$$\sum_{i=1}^{k} m_i^{\min} = \sum_{i=1}^{k} (k - i + 1) = \sum_{j=1}^{k} j = \frac{k(k+1)}{2}. \tag{7}$$

Обозначим разницу между текущей (6) и минимальной (7) суммой как:

$$S = \frac{n + k}{2} - \frac{k(k+1)}{2} = \frac{n - k^2}{2}.$$

Число S характеризует «избыток» значений m_i по сравнению с минимальным возможным набором. Теперь представим значения m_i в виде:

$$m_i = m_i^{\min} + a_i,$$

где $a_i \geq 0$ — целые числа, которые описывают «отклонение» от минимального случая и их сумма $S = a_1 + a_2 + \cdots + a_k = (n - k^2)/2$.

Кроме того,

$$\begin{aligned} &m_i = k - i + 1 + a_i, \\ &m_{i-1} = k - (i-1) + 1 + a_{i-1} = k - i + 2 + a_{i-1}, \\ &m_{i-1} - m_i = 1 + a_{i-1} - a_i \geq 1 \quad \Longrightarrow \quad a_{i-1} - a_i \geq 0. \end{aligned}$$

Таким образом, набор $(a_1, a_2, \ldots, a_k)$, где $a_1 \geq a_2 \geq \cdots \geq a_k \geq 0$, задает разбиение числа S на не более чем k частей. В силу сопряжения диаграмм Юнга, количество таких разбиений равно количеству разбиений числа S на части, не превосходящие k. Следовательно, $r(n,k) = P(S,k)$, причем $S = (n-k^2)/2$ является неотрицательным целым числом, то есть $n \geq k^2$ и $n \equiv k \pmod 2$. □

3. Асимптотическое сравнение классов разбиений

Теорема 2. Пусть n и k — натуральные числа, такие, что $n \equiv k \pmod 2$, $n \geq k^2$ и $k = O(n^{1/3})$. Тогда при $n \to \infty$ справедливы следующие асимптотические соотношения:

$$\frac{q(n,k)}{r(n,k)} = 2^{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right) \quad \text{и} \quad \frac{p(n,k)}{r(n,k)} = 2^{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right).$$

Доказательство. Условие $k = O(n^{1/3})$ соответствует области применимости асимптотики (3) для числа ограниченных разбиений $P(n,k)$. Подставляя данную асимптотику с учетом биекций (1), (2), (5) и условий $n \equiv k \pmod 2$, $n \geq k^2$ в выражения для $q(n,k)$ и $r(n,k)$, получаем:

$$\begin{aligned} q(n,k) &= \frac{1}{k!}\binom{t-1}{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right), \\ r(n,k) &= \frac{1}{k!}\binom{s-1}{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right), \end{aligned}$$

где

$$t = n - \frac{k(k+1)}{2} \quad \text{и} \quad s = \frac{n-k^2}{2}.$$

Отношение $q(n,k)/r(n,k)$ принимает вид:

$$\begin{aligned} \frac{q(n,k)}{r(n,k)} &= \frac{(t-1)!}{(k-1)!(t-k)!} \cdot \frac{(k-1)!(s-k)!}{(s-1)!} \exp\left(O\left(\frac{k^3}{n}\right)\right) \\ &= \frac{(t-1)!(s-k)!}{(t-k)!(s-1)!} \exp\left(O\left(\frac{k^3}{n}\right)\right). \end{aligned} \tag{8}$$

Тогда при $n \to \infty$ и $k = O(n^{1/3})$:

$$t = n - \frac{k^2}{2} - \frac{k}{2} = n + O(n^{2/3}),$$
$$s = \frac{n}{2} - \frac{k^2}{2} = \frac{n}{2} + O(n^{2/3}),$$
$$t - k = n - \frac{k^2}{2} - \frac{k}{2} - k = n + O(n^{2/3}),$$
$$s - k = \frac{n}{2} - \frac{k^2}{2} - k = \frac{n}{2} + O(n^{2/3}).$$

Воспользуемся формулой Стирлинга:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \exp\left(\frac{1}{12n + \theta_n}\right),$$

где $0 < \theta_n < 1$.

Тогда

$$\frac{(t-1)!}{(t-k)!} = \frac{\sqrt{2\pi(t-1)}\left(\frac{t-1}{e}\right)^{t-1}\exp\left(\frac{1}{12(t-1)+\theta_{t-1}}\right)}{\sqrt{2\pi(t-k)}\left(\frac{t-k}{e}\right)^{t-k}\exp\left(\frac{1}{12(t-k)+\theta_{t-k}}\right)} = \sqrt{\frac{t-1}{t-k}} \cdot \frac{(t-1)^{t-1}}{(t-k)^{t-k}} \cdot e^{1-k} \cdot \exp(\Delta_t),$$

где при $n \to \infty$ и $k = O(n^{1/3})$:

$$\Delta_t = \frac{1}{12(t-1) + \theta_{t-1}} - \frac{1}{12(t-k) + \theta_{t-k}} = \frac{12(1-k) + (\theta_{t-k} - \theta_{t-1})}{(12(t-1) + \theta_{t-1})(12(t-k) + \theta_{t-k})}$$
$$= \frac{O(k)}{144n^2 + O(n^{5/3})} = \frac{O(k)}{O(n^2)} = O\left(\frac{k}{n^2}\right) = O(n^{-5/3}).$$

Аналогично,

$$\frac{(s-k)!}{(s-1)!} = \frac{\sqrt{2\pi(s-k)}\left(\frac{s-k}{e}\right)^{s-k}\exp\left(\frac{1}{12(s-k)+\theta_{s-k}}\right)}{\sqrt{2\pi(s-1)}\left(\frac{s-1}{e}\right)^{s-1}\exp\left(\frac{1}{12(s-1)+\theta_{s-1}}\right)} = \sqrt{\frac{s-k}{s-1}} \cdot \frac{(s-k)^{s-k}}{(s-1)^{s-1}} \cdot e^{k-1} \cdot \exp(\Delta_s),$$

где при $n \to \infty$ и $k = O(n^{1/3})$:

$$\Delta_s = \frac{1}{12(s-k) + \theta_{s-k}} - \frac{1}{12(s-1) + \theta_{s-1}} = O(n^{-5/3}).$$

Далее рассмотрим выражение под знаком корня при $n \to \infty$ и $k = O(n^{1/3})$:

$$\frac{t-1}{t-k} \cdot \frac{s-k}{s-1} = \frac{n + O(n^{2/3})}{n + O(n^{2/3})} \cdot \frac{\frac{n}{2} + O(n^{2/3})}{\frac{n}{2} + O(n^{2/3})} = \left(1 + O(n^{-1/3})\right) \cdot \left(1 + O(n^{-1/3})\right)$$
$$= 1 + O(n^{-1/3}) = \exp\left(\ln[1 + O(n^{-1/3})]\right) = \exp\left(O(n^{-1/3})\right).$$

Рассмотрим:

$$A = \frac{(t-1)^{t-1}}{(t-k)^{t-k}}.$$

Логарифмируем:

$$\ln A = (t-1)\ln(t-1) - (t-k)\ln(t-k).$$

Далее $\ln(t-1) > \ln(t-k)$ при $n \to \infty$ и $k = O(n^{1/3})$, тогда:

$$\ln(t-1) - \ln(t-k) = \ln\left(\frac{t-1}{t-k}\right) = \ln\left(1 + O(n^{-1/3})\right) = O(n^{-1/3}).$$

Получаем оценку сверху:

$$\ln A < (t-1)\ln(t-1) - (t-k)\ln(t-1) = (k-1)\ln(t-1).$$

Получаем оценку снизу:

$$\ln A > (t-1)\ln(t-k) - (t-k)\ln(t-k) = (k-1)\ln(t-k).$$

Объединяя две оценки, имеем:

$$(k-1)\ln(t-1) > \ln A > (k-1)\ln(t-k).$$

Разность границ равна при $n \to \infty$ и $k = O(n^{1/3})$ равна:

$$(k-1)[\ln(t-1) - \ln(t-k)] = (k-1) \cdot O(n^{-1/3}) = O(1).$$

Следовательно,

$$\ln A = (k-1)\ln(t-1) + O(1),$$

что дает при $n \to \infty$ и $k = O(n^{1/3})$:

$$A = e^{\ln A} = e^{(k-1)\ln(t-1)+O(1)} = (t-1)^{k-1}\exp\left(O(1)\right).$$

Аналогично для:

$$B = \frac{(s-1)^{s-1}}{(s-k)^{s-k}},$$

при $n \to \infty$ и $k = O(n^{1/3})$:

$$B = e^{\ln B} = e^{(k-1)\ln(s-1)+O(1)} = (s-1)^{k-1}\exp\left(O(1)\right).$$

Все полученные промежуточные результаты подставляем в (8):

$$\begin{aligned}\frac{q(n,k)}{r(n,k)} &= \exp\left(O(n^{-1/3})\right) \cdot \frac{(t-1)^{k-1}\exp\left(O(1)\right)}{(s-1)^{k-1}\exp\left(O(1)\right)} \cdot \exp(\Delta_t) \cdot \exp(\Delta_s) \cdot \exp\left(O\left(\frac{k^3}{n}\right)\right) \\ &= \left(\frac{t-1}{s-1}\right)^{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right).\end{aligned}$$

Рассмотрим дробь:

$$\begin{aligned}\left(\frac{t-1}{s-1}\right)^{k-1} &= \left(\frac{n + O(n^{2/3})}{\frac{n}{2} + O(n^{2/3})}\right)^{k-1} = \left(2 \cdot \frac{1 + O(n^{-1/3})}{1 + O(n^{-1/3})}\right)^{k-1} = \left(2 \cdot \left(1 + O(n^{-1/3})\right)\right)^{k-1} \\ &= 2^{k-1}\left(1 + O(n^{-1/3})\right)^{k-1} = 2^{k-1}\exp\left((k-1)\ln(1 + O(n^{-1/3}))\right) \\ &= 2^{k-1}\exp\left(O(k \cdot n^{-1/3})\right) = 2^{k-1}\exp(O(1)).\end{aligned}$$

Тогда:

$$\frac{q(n,k)}{r(n,k)} = 2^{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right).$$

Аналогичные рассуждения можно привести для $p(n,k)/r(n,k)$. Используя обозначения $t = n - k$ и $s = \frac{n-k^2}{2}$, получаем при $n \to \infty$ и $k = O(n^{1/3})$, что:

$$\frac{p(n,k)}{r(n,k)} = 2^{k-1} \exp\left(O\left(\frac{k^3}{n}\right)\right).$$

□

4. Алгоритмы подсчета различных нечетных разбиений

Рассмотрим два алгоритма вычисления числа разбиений на различные нечетные части. Наивный из них, алг. 1 представляет собой прямой рекурсивный вызов. Очевидно, что такой подход неэффективен и приводит к экспоненциальному времени работы, что делает его неприменимым для больших n из-за переполнения стека вызовов.

Алгоритм 2. NrDistinctOddParts (in n, k; out result)

```
1: Инициализировать матрицу P[1..n][1..k] нулями
2: P[1][1] ← 1
3: for i ← 1 to n do
4:     for j ← 1 to min(k, i) do
5:         if i > 2j then
6:             P[i][j] ← P[i][j] + P[i − 2j][j]
7:         end if
8:         if i > 2j − 1 и j > 1 then
9:             P[i][j] ← P[i][j] + P[i − (2j − 1)][j − 1]
10:        end if
11:    end for
12: end for
13: return P[n][k]
```

Алгоритм 2 вычисляет количество разбиений $r(n,k)$ на различные нечетные части, реализуя формулу (4) методом динамического программирования. Промежуточные результаты сохраняются в матрице P, благодаря чему временная и пространственная сложность составляют $O(n \cdot k)$. Количество всех разбиений числа n на различные нечетные части получается суммированием по всем допустимым k, для которых выполнены условия $n \equiv k \pmod 2$ и (согласно теореме 1) $n \geq k^2$:

$$r(n) = \sum_{\substack{k=1 \\ n \equiv k \pmod 2}}^{\lfloor \sqrt{n} \rfloor} \text{NrDistinctOddParts}(n,k). \tag{9}$$

Второй алгоритм основан на доказанной биекции (5). Для вычисления $P(n,k)$ воспользуемся рекуррентной формулой:

$$P(n,k) = \begin{cases} 1, & \text{при } n = 0, k \geq 0, \\ 0, & \text{при } n > 0, k = 0, \\ P(n,n), & \text{при } k > n, \\ P(n,k-1) + P(n-k,k), & \text{при } 1 \leq k \leq n. \end{cases}$$

Алгоритм 3. NrRestrictParts (in n, k; out result)

```
1: Инициализировать массив P[1..n+1] нулями
2: P[1] ← 1
3: for m ← 1 to k do
4:     for i ← m + 1 to n + 1 do
5:         P[i] ← P[i] + P[i − m]
6:     end for
7: end for
8: return P[n + 1]
```

Алгоритм 3 вычисляет $P(n,k)$ за время $O(n \cdot k)$ и использует $O(n)$ памяти. Следовательно, общее количество разбиений числа n на различные нечетные части можно получить суммированием по всем допустимым k:

$$r(n) = \sum_{\substack{k=1 \\ n \equiv k \pmod 2}}^{\lfloor \sqrt{n} \rfloor} \text{NrRestrictParts}\left(\frac{n-k^2}{2}, k\right). \tag{10}$$

5. Вычислительные эксперименты

Описанные в разделе 4 алгоритмы были реализованы на языке программирования Python. Для оценки производительности проведено сравнение двух методов вычисления количества разбиений натурального числа n на различные нечетные части, основанных на формулах (9) и (10) соответственно. Вычисления проводились на компьютере с процессором Intel Core i7-13700K (3.40 ГГц) и 32 ГБ ОЗУ (частота 4000 МГц) под управлением Windows 10 x64. Временные замеры выполнены с использованием встроенного модуля `timeit` при числе повторений 10. Результаты (среднее время исполнения в секундах) представлены в табл. 1. Корректность реализации проверена совпадением результатов с известными последовательностями из базы данных OEIS [11]. Исходные коды свободно доступны в репозитории GitHub [12].

Анализ данных показывает, что второй алгоритм демонстрирует существенно более высокую производительность, особенно с ростом n. При этом оба алгоритма дают идентичные результаты, что подтверждает их корректность. Таким образом, для практических вычислений количества разбиений на различные нечетные части целесообразно использовать подход, основанный на доказанной биекции (5), который обеспечивает значительный выигрыш в скорости.

Численная верификация асимптотики теоремы 2 представлена в табл. 2. Видно, что с ростом n отношение $q(n,k)/r(n,k)$ стремится к 2^{k-1}, а для малых k совпадение почти

Таблица 1. Время выполнения, с.

n	Алг. 2	Алг. 3	$r(n)$
100	< 0.001	< 0.001	2.6×10^{3}
200	0.002	< 0.001	3.1×10^{5}
500	0.010	0.001	6.0×10^{9}
1000	0.037	0.003	5.2×10^{14}
3333	0.439	0.037	7.4×10^{28}
5000	0.993	0.086	9.2×10^{35}
7777	2.402	0.216	3.6×10^{45}
9999	4.065	0.363	1.1×10^{52}
15000	9.258	0.833	2.8×10^{64}
25000	26.469	2.381	1.3×10^{84}
49999	110.745	9.699	2.4×10^{120}

полное уже при небольших n. Заметное влияние $O(k^3/n)$ проявляется при $k \sim n^{1/3}$, но общая тенденция сохраняется, что подтверждает справедливость теоремы.

Таблица 2. Численные значения отношения $q(n,k)/r(n,k)$

$n = 5000$		$n = 20000$		$n = 11111$		$n = 33333$	
k	ratio	k	ratio	k	ratio	k	ratio
2	1.9992	2	1.9998	1	1	1	1
4	8.0048	4	8.0012	3	4	3	3.9999
6	32.1449	6	32.0360	5	16.0144	5	16.004
8	129.8186	8	128.4496	7	64.2430	7	64.0807
10	528.5810	10	516.0597	9	258.5113	9	256.8319
12	2175.5090	12	2078.7677	11	1044.6084	11	1030.7951

6. Проверка гипотез о рангах

Как отмечалось ранее, разбиения на различные нечетные части естественно возникают при вычислении ранга группы центральных единиц целочисленного группового кольца знакопеременной группы A_n. Для решения данной задачи рассматриваются разбиения $\lambda = (a_1, \ldots, a_k)$ натурального числа n, удовлетворяющие следующему набору условий [13]:

1. a_i нечетно при $1 \le i \le k$.
2. $a_i \neq a_j$ при $i \neq j$.
3. $n \equiv k \pmod 4$.
4. $\prod_{i=1}^{k} a_i$ *не является квадратом* натурального числа.

Число разбиений, удовлетворяющих первым двум условиям, обозначается $r(n)$ — разбиения на различные нечетные части. Количество разбиений, удовлетворяющих первым трем условиям, обозначается $r_4(n)$. Наконец, число разбиений, для которых выполнены все четыре условия, есть $\mathrm{rank}(n)$.

В работе [8] была выдвинута гипотеза о поведении этих величин, а в [7] предпринята попытка ее численной проверки для небольших значений n. Полученные в предыдущих раз-

делах алгоритмы позволяют распространить анализ на значительно большие n. Гипотеза Каргаполова при $n \to \infty$ утверждает, что:

$$\operatorname{rank}(n) \sim r_4(n) \sim \frac{r(n)}{2}.$$

Для количественной оценки введем относительную погрешность:

$$\delta(n) = \frac{|r(n)/2 - r_4(n)|}{r_4(n)},$$

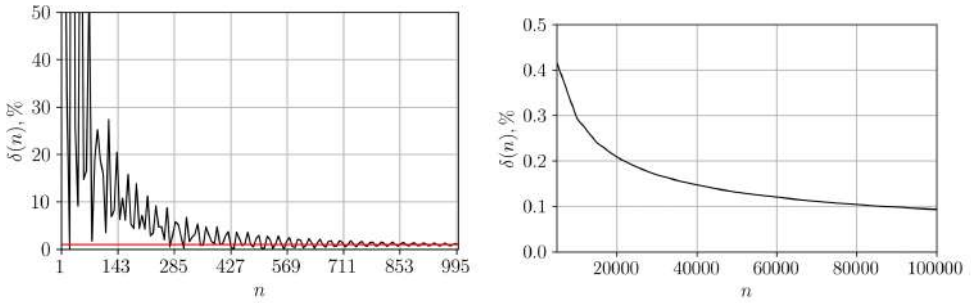
где $r_4(n)$ — точное число разбиений, вычисляемое по формуле:

$$r_4(n) = \sum_{\substack{k=1 \\ n \equiv k \pmod 4}}^{\lfloor \sqrt{n} \rfloor} \textsc{NrRestrictParts}\left(\frac{n-k^2}{2}, k\right).$$

Результаты расчетов сведены в табл. 3. При малых n относительная погрешность $\delta(n)$ испытывает заметные колебания, однако с ростом n они быстро сглаживаются. Начиная с $n = 1000$ относительная погрешность монотонно убывает и при $n \geq 90000$ становится менее 0.1%. На рис. 1 показана динамика $\delta(n)$ в двух масштабах. Красной горизонтальной линией отмечен уровень 1%. Таким образом, во всем исследованном диапазоне гипотеза Каргаполова подтверждается.

Таблица 3. Значения $r_4(n)$ и относительной погрешности $\delta(n)$ при различных n

n	$r_4(n)$	$\delta(n), \%$	n	$r_4(n)$	$\delta(n), \%$
100	1.01×10^{3}	31.349	10 000	5.65×10^{51}	0.294
200	1.72×10^{5}	7.327	20 000	3.97×10^{74}	0.208
300	6.52×10^{6}	6.530	30 000	1.48×10^{92}	0.169
400	1.73×10^{8}	0.939	40 000	1.00×10^{107}	0.147
500	3.04×10^{9}	0.088	50 000	1.20×10^{120}	0.131
600	4.01×10^{10}	2.123	60 000	8.06×10^{131}	0.120
700	4.46×10^{11}	1.533	70 000	6.14×10^{142}	0.111
800	4.24×10^{12}	0.627	80 000	8.31×10^{152}	0.104
900	3.50×10^{13}	0.772	90 000	2.74×10^{162}	0.098
1000	2.58×10^{14}	1.105	100 000	2.77×10^{171}	0.093



а) в диапазоне $n \in [1, 1000]$ б) в диапазоне $n \in [5000, 100000]$

Рис. 1. Относительная погрешность гипотезы

Заключение

В работе исследованы разбиения натурального числа n на ровно k попарно различных нечетных слагаемых. Установлено, что количество $r(n,k)$ таких разбиений выражается через классическую функцию разбиений $P((n-k^2)/2,k)$ с ограничением на части, не превосходящие k, что позволяет сводить задачи о различных нечетных разбиениях к хорошо изученному объекту. При $k = O(n^{1/3})$ доказано, что отношения $q(n,k)/r(n,k)$ и $p(n,k)/r(n,k)$ асимптотически стремятся к 2^{k-1} при $n \to \infty$.

На основе полученных результатов предложены два алгоритма динамического программирования для вычисления $r(n,k)$ с временной сложностью $O(n \cdot k)$. Первый алгоритм основан на рекуррентном соотношении различных нечетных разбиений. Второй алгоритм, использующий биекцию с функцией $P(n,k)$, обладает оптимизированной пространственной сложностью $O(n)$ против $O(n \cdot k)$ и на практике показывает существенно более высокую производительность. Проведенные вычислительные эксперименты подтвердили корректность алгоритмов и справедливость асимптотических формул.

Разработанные алгоритмы позволили провести проверку гипотезы Каргаполова о рангах группы центральных единиц целочисленного группового кольца знакопеременной группы A_n, связанных с разбиениями на различные нечетные части с дополнительными ограничениями. Вычисления, выполненные до $n = 100000$, подтвердили справедливость гипотезы на исследованном диапазоне, причем относительная погрешность $\delta(n)$ после начальных колебаний монотонно убывает и начиная с $n = 90000$ не превышает 0.1%

В качестве перспективного направления можно выделить исследование связи между разбиениями числа n на ровно k нечетных частей (с возможными повторениями) и функциями $r(n,k)$ и $P(n,k)$.

Литература

1. Sagan B.E. The Symmetric Group. Vol. 203. New York, NY: Springer New York, 2001. Graduate Texts in Mathematics. DOI: 10.1007/978-1-4757-6804-6.
2. Szalay M., Turán P. On some problems of the statistical theory of partitions with application to characters of the symmetric group. I // Acta Mathematica Academiae Scientiarum Hungaricae. 1977. Sept. Vol. 29, no. 3–4. P. 361–379. DOI: 10.1007/BF01895857.
3. Cossaboom C.H. Hook length biases for self-conjugate partitions and partitions with distinct odd parts // Journal of Number Theory. 2025. Vol. 277. P. 290–324. DOI: 10.1016/j.jnt.2025.02.002.
4. Amdeberhan T., Andrews G., Ono K., Singh A. Hook lengths in self-conjugate partitions // Proceedings of the American Mathematical Society, Series B. 2024. July. Vol. 11, no. 31. P. 345–357. DOI: 10.1090/bproc/220.
5. Dawsey M., Sharp B. Self-conjugate t-core partitions and applications // Australasian Journal of Combinatorics. 2022. Jan. Vol. 82. P. 212–227.
6. Santos J.P.O., Matte M.L. A New Approach to Integer Partitions // Bulletin of the Brazilian Mathematical Society, New Series. 2018. Dec. Vol. 49, no. 4. P. 811–847. DOI: 10.1007/s00574-018-0082-z.
7. Самойлов А.А. Распределение квадратов и проверка гипотез в нечетных разбиениях чисел // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2024. Т. 13, № 1. С. 22—37. DOI: 10.14529/cmse240102.

8. Каргаполов А.В. Центральные единицы целочисленных групповых колец знакопеременных групп: дис. канд. физ.-мат. наук. Южно-Уральский государственный университет, Россия, 2012. 87 с.

9. Szekeres G. Some asymptotic formulae in the theory of partitions (II) // The Quarterly Journal of Mathematics. 1953. Jan. Vol. 4, no. 1. P. 96–111. DOI: 10.1093/qmath/4.1.96.

10. Canfield E.R. From Recursions to Asymptotics: On Szekeres' Formula for the Number of Partitions // The Electronic Journal of Combinatorics. 1996. Nov. Vol. 4, no. 2. R6. DOI: 10.37236/1321.

11. Inc. O.F. Sequence A000700: Number of partitions of n into distinct odd parts. 2026. URL: https://oeis.org/A000700 (дата обращения: 23.03.2026).

12. Самойлов А.А. Исследование разбиений на различные нечетные части. URL: https://github.com/SashaSamoilov/NrDistinctOddParts (дата обращения: 21.04.2026).

13. Ferraz R.A. Simple components and central units in group algebras // Journal of Algebra. 2004. Vol. 279, no. 1. P. 191–203. DOI: 10.1016/j.jalgebra.2004.05.005.

Самойлов Александр Андреевич, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse260105

ON RELATIONS BETWEEN PARTITION CLASSES WITH A RESTRICTED NUMBER OF PARTS

© 2026 A.A. Samoilov

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: samoilovaa@susu.ru

Received: 27.01.2026

This paper investigates relations between classes of partitions of a positive integer n with restricted number of parts. The main focus is on partitions into exactly k distinct odd parts. To relate this class of partitions to those with restricted part size, we employ conjugation of Young diagrams. This allows one to construct an explicit bijection and prove that the number of such partitions $r(n,k)$ exactly equals $P((n-k^2)/2,k)$ for $n \equiv k \pmod 2$ and $n \geq k^2$, where the parts of the latter do not exceed k. Using this relation, we compare the asymptotic behavior of the partition functions $r(n,k)$, $p(n,k)$ — the number of partitions of n into exactly k parts, and $q(n,k)$ — the number of partitions of n into k distinct parts. For $k = O(n^{1/3})$, it is proved that the ratios $q(n,k)/r(n,k)$ and $p(n,k)/r(n,k)$ asymptotically tend to 2^{k-1} as $n \to \infty$. For the numerical verification of the obtained theorems, two algorithms based on dynamic programming are implemented. The first algorithm relies on the recurrence relation for counting partitions into distinct odd parts $r(n,k)$. The second one relies on the proved bijection and the recurrence formula for the number of partitions $P(m,k)$ whose parts do not exceed k. Computational experiments show that the algorithm based on the established bijection provides a significant performance gain as n increases. The developed algorithms are also applied to verify the Kargapolov conjecture concerning the ranks of the group of central units of the integer group ring of the alternating group A_n, which are related to partitions into distinct odd parts under additional constraints. Computations performed up to $n = 100000$ confirm the asymptotic behavior, with a relative error below 0.1%.

Keywords: integer partitions, partitions into distinct odd parts, combinatorial bijection, asymptotic analysis, dynamic programming.

FOR CITATION

Samoilov A.A. On Relations between Partition Classes with a Restricted Number of Parts. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2026. Vol. 15, no. 1. P. 96–109. (in Russian) DOI: 10.14529/cmse260105.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Sagan B.E. The Symmetric Group. Vol. 203. New York, NY: Springer New York, 2001. Graduate Texts in Mathematics. DOI: 10.1007/978-1-4757-6804-6.
2. Szalay M., Turán P. On some problems of the statistical theory of partitions with application to characters of the symmetric group. I. Acta Mathematica Academiae Scientiarum Hungaricae. 1977. Sept. Vol. 29, no. 3–4. P. 361–379. DOI: 10.1007/BF01895857.
3. Cossaboom C.H. Hook length biases for self-conjugate partitions and partitions with distinct odd parts. Journal of Number Theory. 2025. Vol. 277. P. 290–324. DOI: 10.1016/j.jnt.2025.02.002.
4. Amdeberhan T., Andrews G., Ono K., Singh A. Hook lengths in self-conjugate partitions. Proceedings of the American Mathematical Society, Series B. 2024. July. Vol. 11, no. 31. P. 345–357. DOI: 10.1090/bproc/220.
5. Dawsey M., Sharp B. Self-conjugate t-core partitions and applications. Australasian Journal of Combinatorics. 2022. Jan. Vol. 82. P. 212–227.
6. Santos J.P.O., Matte M.L. A New Approach to Integer Partitions. Bulletin of the Brazilian Mathematical Society, New Series. 2018. Dec. Vol. 49, no. 4. P. 811–847. DOI: 10.1007/s00574-018-0082-z.
7. Samoilov A.A. Distribution of Squares and Hypothesis Verification in Odd Integer Partitions. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2024. Vol. 13, no. 1. P. 22–37. (in Russian) DOI: 10.14529/cmse240102.
8. Kargapolov A.V. Central units of integral group rings of alternating groups: PhD thesis. South Ural State University, Russia, 2012. 87 p. (in Russian).
9. Szekeres G. Some asymptotic formulae in the theory of partitions (II). The Quarterly Journal of Mathematics. 1953. Jan. Vol. 4, no. 1. P. 96–111. DOI: 10.1093/qmath/4.1.96.
10. Canfield E.R. From Recursions to Asymptotics: On Szekeres' Formula for the Number of Partitions. The Electronic Journal of Combinatorics. 1996. Nov. Vol. 4, no. 2. R6. DOI: 10.37236/1321.
11. Inc. O.F. Sequence A000700: Number of partitions of n into distinct odd parts. 2026. URL: https://oeis.org/A000700 (accessed: 23.03.2026).
12. Samoilov A.A. Study of integer partitions into distinct odd parts. URL: https://github.com/SashaSamoilov/NrDistinctOddParts (accessed: 21.04.2026) (in Russian).
13. Ferraz R.A. Simple components and central units in group algebras. Journal of Algebra. 2004. Vol. 279, no. 1. P. 191–203. DOI: 10.1016/j.jalgebra.2004.05.005.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 1.2.3 – Теоретическая информатика, кибернетика (физико-математические науки), 2.3.5 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции: 454080, г. Челябинск, ул. С. Кривой, 79, Издательский центр ЮУрГУ, каб. 2.

Адрес издателя: 454080, г. Челябинск, проспект Ленина, 76.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии `https://vestnikvmi.susu.ru`. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.**
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:

   Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, НОЦ ИИКТ,
   зам. главного редактора Цымблеру М.Л.
3. Адрес электронной почты редакции: `vestnikvmi@susu.ru`.
4. **Плата с авторов за публикацию рукописей не взимается, гонорары авторам не выплачиваются.**

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 1.2.3 – Теоретическая информатика, кибернетика (физико-математические науки), 2.3.5 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции: 454080, г. Челябинск, ул. С. Кривой, 79, Издательский центр ЮУрГУ, каб. 2.

Адрес издателя: 454080, г. Челябинск, проспект Ленина, 76.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии `https://vestnikvmi.susu.ru`. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.**
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:

   Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, НОЦ ИИКТ,
   зам. главного редактора Цымблеру М.Л.
3. Адрес электронной почты редакции: `vestnikvmi@susu.ru`.
4. **Плата с авторов за публикацию рукописей не взимается, гонорары авторам не выплачиваются.**

ВЕСТНИК
ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
Серия
«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»
2026 Том 15, № 1

16+

Техн. редактор *А.В. Миних*

Издательский центр Южно-Уральского государственного университета

Подписано в печать 25.05.2026. Дата выхода в свет 11.06.2026. Формат 60×84 1/8. Печать цифровая.
Усл. печ. л. 13,02. Тираж 500 экз. Заказ 84/127. Цена свободная.

Отпечатано в типографии Издательского центра ЮУрГУ.
454080, г. Челябинск, проспект Ленина, 76.