

Сибирское отделение
Российской академии наук

ISSN 2073-0667

ПРОБЛЕМЫ ИНФОРМАТИКИ

ПРОБЛЕМЫ ИНФОРМАТИКИ № 4 2025



ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
ПАРАЛЛЕЛЬНОЕ СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ
И ВЫЧИСЛИТЕЛЬНЫЕ ТЕХНОЛОГИИ

4
—
2025

ПРОБЛЕМЫ ИНФОРМАТИКИ № 4 (69) 2025 г.

Журнал выходит ежеквартально, издается с 2008 г.

Учредитель журнала — Институт вычислительной математики и математической геофизики СО РАН.

Редакционный совет

Председатель — акад. НАН РК М. Н. Калимолдаев,
акад. РАН А. Л. Асеев, проф. В. А. Васенин, акад. РАН С. Н. Васильев,
проф. В. М. Вишневецкий, акад. РАН С. С. Гончаров, акад. РАН Н. А. Колчанов,
акад. РАН Н. А. Кузнецов, акад. РАН А. П. Кулешов, проф. РАН М. А. Марченко,
проф. А. Г. Марчук, А. Ю. Пальянов, проф. Б. Я. Рябко, акад. РАН И. А. Соколов,
проф. А. Н. Сотников, чл.-кор. РАН Ю. А. Флеров.

Редколлегия

Главный редактор — проф. В. Э. Малышкин,
Д. А. Афонников, Д. Ж. Ахмед-Заки, А. Г. Вострецов, Б. С. Гольдштейн, В. И. Гужов,
Ю. А. Загорюлько, В. А. Иванисенко, С. Д. Каракозов, В. Н. Касьянов, О. В. Кибис,
В. В. Корнеев, И. В. Котенко, И. М. Куликов, М. Г. Курносов, С. А. Лашин, Т. П. Любимова,
А. Н. Ляхов, Ю. Г. Матушкин, В. В. Окольников, Б. В. Поллер, М. П. Пономаренко,
А. С. Родионов (зам. гл. редактора), А. Н. Савостьянов, М. А. Сонькин, В. В. Шахов (зам. гл.
редактора), М. С. Хайретдинов, И. Г. Черных, Moonseong Kim (Korea), V. D. Nguyen (Vietnam),
Michele Pagano (Italy).

Редакция: отв. секретарь М. С. Делидович, системный администратор В. А. Перепелкин,
верстка Д. В. Лазуткин, логист Л. В. Трофимова.

Адрес редакции, издателя: 630090, г. Новосибирск, просп. Академика Лаврентьева, д. 6,
ИВМ и МГ СО РАН

тел. (383) 330-96-43; e-mail: problem-info@sscc.ru, <http://www.problem-info.sssc.ru>.

Журнал зарегистрирован в Федеральной службе по надзору в сфере массовых коммуникаций,
связи и охраны культурного наследия. Свидетельство ПИ № ФС77-32088 от 27 мая 2008 г.

Журнал распространяется по подписке. Оформление подписки на сайте «Объединенного
каталога „Пресса России“» https://www.pressa-rf.ru/cat/1/edition/y_e69980/, подписной
индекс 69980, и через интернет-магазин «Пресса по подписке»

https://www.akc.ru/itm/problemu_i-informatiki/. Цена свободная. Журнал
распространяется на территории России.

**Журнал включен в Перечень ведущих рецензируемых научных журналов,
рекомендованных для публикаций Высшей аттестационной комиссией. Входит в
«Белый список» Единого государственного перечня научных изданий (ЕГПНИ).**

Все права авторов сохранены. Использование материалов журнала возможно только
с разрешения редакции и авторов.

Отпечатано в типографии «АЛЕКСПРЕСС» ИП Малыгин Алексей Михайлович.

Адрес: 630090, Новосибирск, пр-т Академика Лаврентьева, 6/1, оф. 104, тел. +7 (383) 217-43-46.

Формат 60 × 84 1/8. Усл. печ. л. 10,7. Печать офсетная.

Тираж 50 экз. Заказ № 1026. Подписано в печать 15.12.2025 г. Выход в свет 25.12.2025 г.

© Институт вычислительной математики и математической геофизики СО РАН, 2025

JOURNAL “PROBLEMS OF INFORMATICS”. No. 4 (69) 2025

Publisher: Institute of Computational Mathematics and Mathematical Geophysics of Siberian Branch of Russian Academy of Sciences.

Editorial Council

Chairman Academician of the National Academy of Sciences of the Republic of Kazakhstan

M. N. Kalimoldayev

Full Member of the RAS A. L. Aseev, Professor V. A. Vasenin, Full Member of RAS C. N. Vassilyev,

Professor V. M. Vishnevsky, Full Member of RAS S. S. Goncharov, Full Member of RAS

N. A. Kolchanov, Full Member of RAS N. A. Kuznetsov, Full Member of RAS A. P. Kuleshov,

Professor of RAS M. A. Marchenko, Professor A. G. Marchuk, A. YU. Palyanov, Professor

B. Y. Ryabko, Full Member of RAS I. A. Sokolov, Professor A. N. Sotnikov, Corr. Member RAS

Y. A. Flerov.

Editorial board

The Editor-in-Chief Professor V. E. Malyshkin

Associate Editors-in-Chief: A. S. Rodionov, V. V. Shakhov

D. A. Afonnikov D. Zh. Akhmed-Zaki, A. G. Vostretsov, B. S. Goldstein, V. I. Guzhov,

Y. A. Zagorulko, V. A. Ivanisenko, S. D. Karakozov, V. N. Kasyanov, O. V. Kibis, V. V. Korneev,

I. V. Kotenko, I. M. Kulikov, M. G. Kurnosov, S. A. Lashin, T. P. Lyubimova, A. I. Lyakhov,

V. V. Okolnishnikov, B. V. Poller, M. P. Ponomarenko, A. N. Savostyanov, M. A. Sonkin,

M. S. Khairtdinov, I. G. Chernykh, Moonseong Kim (Korea), Van Duc Nguyen (Vietnam),

Michele Pagano (Italy).

Editorial staff: Managing Editor M. S. Delidovich, System Administrator V. A. Perepelkin,

Maker-up D. V. Lazutkin, Logistician L. V. Trofimova.

Address of the editorial office: 630090, pr. Lavrentieva, 6, Novosibirsk, Russia, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS.

Phone: +7 (383) 330-96-43; e-mail: problem-info@sscc.ru, <http://www.problem-info.sssc.ru>.

The journal has been registered in accordance with Legislation of the Russian Federation. Certificate of Mass Media Registration: ПИ № ФС77-32088, of 27 May, 2008, ISSN 2073-0667. The journal is distributed in Russia.

The journal “Problems of Informatics” is in the List of Peer-Reviewed Scientific Journals for publication of scientific results of Ph.D. and Dr. of Sci.

All rights reserved. The journal contents may only be used by the permission of editors and authors.

СОДЕРЖАНИЕ

Прикладные информационные технологии

<i>Бакулина М. П.</i> Эффективный алгоритм сжатия с помощью преобразования данных словарного типа	5
<i>Бредихин С. В., Щербакова Н. Г.</i> Выявление сообществ в мультиплексной сети авторов научного журнала	11
<i>Герб А. Р., Девятых Е. Е., Омарова Г. А.</i> Сравнение методов графовой редукции первого, второго и третьего поколения в моделях химической кинетики	25
<i>Монахов О. Г., Монахова Э. А.</i> Генерация многоуровневых регулярных сетей на основе операции композиции модифицированных хордальных графов с использованием больших языковых моделей	38
<i>Усова М. А., Лебедев И. Г., Штанюк А. А., Баркалов К. А.</i> Алгоритм глобальной оптимизации для настройки гиперпараметров методов машинного обучения	52

Параллельное системное программирование и вычислительные технологии

<i>Малышкин В. Э., Перепелкин В. А., Спирин В. А.</i> Обеспечение автоматического конструирования программ с использованием спецвычислителей на основе концепции активных знаний в системе LuNA	73
Алфавитный указатель авторов за 2025 год	89
Правила представления и подготовки рукописей для публикации в журнале «ПРОБЛЕМЫ ИНФОРМАТИКИ»	91

На сайте «Объединенного каталога "Пресса России"» www.pressa-rf.ru
можно оформить подписку на печатную версию журнала
«Проблемы информатики» по подписному индексу **69980**,
а также подписаться через интернет-магазин
«Пресса по подписке» www.akc.ru

CONTENTS

Applied information technologies

<i>Bakulina M. P.</i> An Efficient Compression Algorithm Using Dictionary-Type Data Transformation	5
<i>Bredikhin S. V., Scherbakova N. G.</i> Community Detection in the Multiplex Network of Scientific Journal Authors	11
<i>Gerb A. R., Deviatykh E. E., Omarova G. A.</i> Comparison of First, Second and Third Generation Graph Reduction Methods in Chemical Kinetics Models	25
<i>Monakhov O. G., Monakhova E. A.</i> Generation of Multi-Level Regular Networks Based on the Composition Operation of Modified Chordal Graphs Using Large Language Models	38
<i>Usova M. A., Lebedev I. G., Shtanyuk A. A., Barkalov K. A.</i> A Global Optimization Algorithm for Tuning Hyperparameters of Machine Learning Methods	52

Parallel system programming and computational technologies

<i>Malyshkin V. E., Perepelkin V. A., Spirin V. A.</i> Automatic Program Construction with Accelerators Usage Based on the Active Knowledge Concept in LuNA System	73
Authors by Alphabetical Order Published in 2025	89
Rules of Presentation and Preparation of Manuscripts Offered for Publication	91

AN EFFICIENT COMPRESSION ALGORITHM USING DICTIONARY-TYPE DATA TRANSFORMATION

M. P. Bakulina

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-5-10

EDN: KUQHBT

The problem of efficient lossless compression for dictionary-type data is considered. For such data, the coding algorithm is based on the use of a dictionary formed from the text received for compression. It is also known that data processing, such as BWT, can improve the text compression ratio. In this paper, an efficient dictionary-type data compression algorithm based on the modification of BWT is proposed. Experimental results are presented. The results confirm the increase in the data compression ratio by the proposed algorithm compared to the classic archiver bzip2.

Key words: dictionary, BWT-transformation, compression ratio, encoding time, archiver.

References

1. Burrows, M. A block sorting lossless data compression algorithm. M. Burrows, D. Wheeler, Technical Report 124, Digital Equipment Corporation, 1994. P. 18.
2. Ryabko B. Ya. Data compression by means of a “book stack” // Problems of Information Transmission. 1980, V. 16: (4). P. 265–269.
3. Hmelev D. V. Preobrazovanie Barrouza-Willera, massiv suffiksov i szhatie slovarei // Vse o szhatii dannyh, izobrazhenii i video. [Electron. Res.]: http://compression.ru/download/articles/bwt/khmelev_2003_bwt.pdf.
4. Kärkkäinen J., Sanders P. Simple linear work suffix array construction // In 30th International Colloquium on Automata, Languages and Programming, number 2719 in LNCS, 2003. P. 943–955.
5. Bell T. C., Cleary J. H., Witten I. H. Text compression. / T. C. Bell, J. H. Cleary, I. H. Witten, Prentice Hall. Englewood Cliffs, 1990. P. 318.

The research was carried out within the framework of a state assignment of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS (ICM&MG SB RAS) 0251-2022-0005.

ЭФФЕКТИВНЫЙ АЛГОРИТМ СЖАТИЯ С ПОМОЩЬЮ ПРЕОБРАЗОВАНИЯ ДАННЫХ СЛОВАРНОГО ТИПА

М. П. Бакулина

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.722

DOI: 10.24412/2073-0667-2025-4-5-10

EDN: KUQHBT

Рассматривается задача эффективного сжатия без потерь для данных словарного типа. Для таких данных алгоритм кодирования основан на использовании словаря, формируемого по тексту, поступающему для сжатия. Известно также, что предварительная обработка данных, например, BWT-преобразование, может улучшить коэффициент сжатия текста. В данной работе предлагается эффективный алгоритм сжатия данных словарного типа, основанный на модификации BWT-преобразования. Приведены экспериментальные результаты, подтверждающие увеличение степени сжатия данных предложенным алгоритмом по сравнению с классическим архиватором.

Ключевые слова: словарь, BWT-преобразование, степень сжатия, время кодирования, архиватор.

Введение. Сжатие данных является одной из важных задач теории информации. Это связано с тем, что задача компактного представления данных с сохранением возможности их однозначного восстановления (декодирования) возникает достаточно часто на практике. Так, предварительное сжатие позволяет существенно улучшить характеристики системы связи. Сжатие применяется при хранении информации, ее передаче со спутников и в других приложениях, причем актуальность проблемы сжатия возрастает в связи с увеличением объемов передаваемой и хранимой информации.

В настоящее время широко известны и находят практическое применение многие алгоритмы компактного представления данных, которые дают сжатие лучше, чем стандартные архиваторы. Один из таких подходов при создании алгоритмов сжатия основан на преобразовании Барроуза-Уилера или BWT-преобразовании [1]. BWT используется для предварительной обработки данных перед сжатием. Преобразование меняет порядок символов во входной строке таким образом, что повторяющиеся подстроки образуют на выходе идущие подряд последовательности одинаковых символов.

Так как в полученной после BWT-преобразования последовательности вероятности появления одинаковых символов выше, чем редких, то возникает задача эффективного кодирования длин серий. В известных алгоритмах сжатия на основе BWT для этого выполняется шаг по замене каждого символа расстоянием до его предыдущей встречи (например, в алгоритме move to front, MTF [2]), а далее к полученному набору чисел

Исследования выполнены в рамках государственного задания ИВМиМГ СО РАН 0251-2022-0005.

применяется метод энтропийного сжатия, например, кодирование Хаффмана или арифметическое кодирование. На практике алгоритм сжатия вида $BWT > MTF > \text{Хаффман}$ применен в архиваторе bzip2.

Кроме того, практика показывает, что большинство архиваторов, использующих BWT, достигают хорошего сжатия на текстовых данных, например, текстовых файлов на естественных языках. В [3] доказана обратимость BWT-преобразования и возможность восстановить массив суффиксов.

Кроме алгоритмов, использующих побуквенное кодирование, примером которого является код Хаффмана, известны также методы словарного сжатия данных [4]. В таких методах для имеющегося сообщения, например, текста на естественном языке, составляется словарь, представляющий собой список повторяющихся последовательностей символов (слов) с указанием частоты их появления. Тогда при кодировании сообщения каждому слову приписывается код, длина которого тем меньше, чем выше частота его встречаемости. Отметим, что архиватор bzip2 является одним из лучших архиваторов для данных словарного типа.

В данной работе предлагается эффективный алгоритм сжатия данных словарного типа, основанный на модификации BWT-преобразования. Приводятся экспериментальные результаты, подтверждающие эффективность предложенного метода.

1. Преобразование Барроуза-Уилера. Процедуру BWT-преобразования можно условно разделить на 4 этапа: 1) выделяется блок из входного потока исходных данных; 2) формируется матрица всех перестановок, полученных в результате циклического сдвига блока; 3) все перестановки сортируются в соответствии с лексикографическим порядком символов каждой перестановки; 4) на выход подаются последний столбец матрицы и номер строки, соответствующий первоначальному блоку.

Опишем теперь формальную процедуру BWT-преобразования.

Пусть текст T состоит из $N + 1$ букв, занумерованных с нуля: $T[0 \dots N]$. Буквы $T[i]$ принадлежат некоторому упорядоченному алфавиту A . Зададим на строках из букв алфавита A лексикографический порядок $\leq$. Обозначим через $S_k(T)$ циклический сдвиг текста T на k символов влево:

$$S_k(T)[j] = T[(j + k)(\text{mod } n + 1)].$$

Существует перестановка σ чисел $\{0, \dots, N\}$, которая удовлетворяет условию

$$S_{\sigma(i)}(T) \leq S_{\sigma(i+1)}(T), \quad i = 0, \dots, N - 1. \quad (1)$$

Перестановку σ называют суффиксным массивом. Она играет важную роль в индексации информации (например, с их помощью можно найти все повторяющиеся подстроки текста). Тогда преобразование Барроуза-Уилера текста T — это текст $B[0 \dots N] = BW(T)$, буквы которого заданы соотношением

$$B_i = (S_{\sigma(i)}T)[N] = (S_{\sigma(i)-1}T)[0] = T[\sigma(i) - 1 \pmod{N + 1}].$$

В [3] доказано, что для восстановления исходного текста T из преобразования B достаточно знать число I , задаваемое условием $\sigma(I) = 0$ или $S_{\sigma(I)}T = T$. Чтобы найти перестановку σ , можно воспользоваться любым из известных методов построения суффиксных массивов для текста T (см., например, [5]), а затем добавить суффикс, соответствующий символу $T(N)$.

2. Преобразование данных словарного типа. Рассмотрим теперь BWT-преобразование для данных словарного типа и на основе него опишем новое преобразование, которое будет основой нового алгоритма сжатия данных.

Пусть текст $T[0 \dots N]$ состоит из n частей, разделенных специальными символами (разделителями) $\Lambda_0 \dots \Lambda_{n-1}$, положение которых упорядочено, то есть

$$T = T_0 \Lambda_0 \dots T_{n-1} \Lambda_{n-1}.$$

Будем считать также, что разделители лексикографически строго предшествуют буквам алфавита A :

$$\Lambda_k < a \text{ для } \forall a \in A \setminus \{\Lambda_0, \dots, \Lambda_{n-1}\} \text{ и } \forall k = 0, \dots, n-1.$$

Тогда при применении BWT-преобразования все разделители будут стоять в первом столбце матрицы следующего вида:

$$\begin{array}{c} \Lambda_{\omega(0)} \dots B_0 \\ \vdots \\ \Lambda_{\omega(n-1)} \dots B_{n-1} \\ B_{\eta(n)} \dots B_n \\ B_{\eta(n+1)} \dots B_{n+1} \\ \vdots \\ B_{\eta(N)} \dots B_N, \end{array}$$

где ω — перестановка для разделителей, а η — перестановка для букв последнего столбца.

Рассмотрим теперь преобразование исходного текста B , то есть текст $\tilde{B}$, который получен из текста B с помощью замены всех разделителей Λ_k на один разделитель Λ , предшествующий всем символам алфавита A исходного текста T . Формально это преобразование запишется так:

$$\tilde{B}_i = \begin{cases} B_i, & \text{если } B_i \neq \Lambda_k, \quad k = 0, \dots, n-1 \\ \Lambda, & \text{если } B_i = \Lambda_k \text{ для некоторого } 0 \leq k \leq n-1. \end{cases}$$

В [3] показано, что если по $\tilde{B}$ с соответствующими перестановками вместо индекса i сформировать строки $\tilde{T}_j$, то наборы строк преобразованного текста $(\tilde{T}_j, j = 0, \dots, n-1)$ и строк исходного текста $(T_k, k = 0, \dots, n-1)$ совпадают с точностью до перестановки. Кроме того, по $\tilde{B}$ однозначно восстанавливается и исходный текст T .

3. Алгоритм сжатия данных В_NEW. На основе приведенного выше преобразования опишем теперь новый алгоритм сжатия данных словарного типа В_NEW.

Пусть мы имеем текст, представляющий собой n последовательностей символов $T_0, \dots, T_{n-1}$, разделенных специальными символами-разделителями:

$$T = T_0 \Lambda_0 \dots T_{n-1} \Lambda_{n-1},$$

при этом порядок разделителей соответствует лексикографическому порядку текста, то есть $\Lambda_i < \Lambda_j \Leftrightarrow T_i < T_j$.

Алгоритм В_NEW состоит из двух этапов. На первом этапе построим перестановку σ , удовлетворяющую условию (1). При этом для простоты порядок на суффиксах будем задавать так: $\Lambda_i < \Lambda_j \Leftrightarrow i < j$. Далее с помощью перестановки σ строим текст $\tilde{B}$ по правилу:

Таблица 1

Результаты сравнения степеней сжатия текстов с помощью B_NEW и bzip2

№	Название	V текста (байт)	k (бит/символ)	
			B_NEW	bzip2
1	article	67739	2,79	2,82
2	book	349270	3,06	3,11
3	journal	307930	2,71	2,74
4	document	962186	2,68	2,79
5	works	5410342	2,51	2,66

Таблица 2

Результаты сравнения времени кодирования с помощью B_NEW и bzip2

№	Название	V текста (байт)	t (с)	
			B_NEW	bzip2
1	article	67739	0,86	0,73
2	book	349270	5,43	5,01
3	journal	307930	5,12	4,78
4	document	962186	19,05	18,32
5	works	5410342	63,15	61,48

$$\tilde{B}_i = \begin{cases} S_{\sigma(i)-1}T[0], & \text{если } S_{\sigma(i)-1}T[0] \neq \Lambda_k, k = 0, \dots, n-1 \\ \Lambda, & \text{если } S_{\sigma(i)-1}T[0] = \Lambda_k \text{ для некоторого } 0 \leq k \leq n-1. \end{cases}$$

На втором этапе полученный после такого преобразования (назовем его B_new) текст $\tilde{B}$ подвергнем сжатию с помощью известного метода MTF (см. [2]), а к получившемуся набору чисел — алгоритм Хаффмана, то есть осуществим преобразование B_new > MTF > Хаффман. Отметим, что эта цепочка преобразований аналогична преобразованию в архиваторе bzip2. Отличие заключается в том, что вместо BWT-преобразования к данным словарного типа применяется новое преобразование B_new, позволяющее с помощью группировки данных повысить эффективность сжатия.

4. Экспериментальные результаты и их сравнение. Для подтверждения эффективности предложенного алгоритма сжатия B_NEW было проведено сравнение результатов его работы с результатами работы архиватора bzip2. В качестве данных словарного типа были взяты следующие данные: article — статья в журнале; book — монография; journal — математический журнал; document — документация; works — собрание сочинений А. С. Пушкина.

Сравнение проводилось по двум характеристикам — степени сжатия (использовалась наиболее употребляемая единица измерения бит/символ) и времени кодирования и декодирования. В Таблице 1 приведены результаты степеней сжатия выбранных текстов алгоритмом B_NEW и архиватором bzip2.

В Таблице 2 приведены результаты времени кодирования и декодирования этих же текстов. В качестве результата принималось усредненное общее время работы программы по 10 запускам (использовался таймер в 1000 Гц).

Из таблиц 1 и 2 видно, что предложенный алгоритм B_NEW показывает лучшую степень сжатия по сравнению с алгоритмом bzip2 при сравнительно небольшом увеличении

времени кодирования и декодирования. Заметим также, что наиболее существенное увеличение сжатия наблюдается для текстов с большим объемом памяти, например, `document` и `works`. Эти тексты обладают большим объемом словаря, и поэтому предложенное в п. 3 преобразование данных словарного типа должно давать для этих текстов наиболее эффективное сжатие, что и подтверждается практическими результатами.

Заключение. Проведенное экспериментальное сравнение предложенного метода кодирования, основанного на модификации BWT-преобразования, со стандартным архиватором `bzip2` подтверждает эффективность предложенного алгоритма `B_NEW`. Данный алгоритм может быть использован для сжатия различных данных словарного типа, например, текстов на естественных языках.

Список литературы

1. Burrows M. A block sorting lossless data compression algorithm // M. Burrows, D. Wheeler. Technical Report 124, Digital Equipment Corporation, 1994. 18 pages.
2. Рябко Б. Я. Data compression by means of a «book stack» // Problems of Information Transmission. 1980, V. 16: (4). P. 265–269.
3. Хмелев Д. В. Преобразование Барроуза-Уилера, массив суффиксов и сжатие словарей // Все о сжатии данных, изображений и видео: сайт. [Эл. Рес.]: http://compression.ru/download/articles/bwt/khmelev_2003_bwt.pdf.
4. Bell T. C., Cleary J. H., Witten I. H. Text compression // T. C. Bell, J. H. Cleary, I. H. Witten, Prentice Hall. Englewood Cliffs, 1990. С. 318.
5. Kärkkäinen J., Sanders P. Simple linear work suffix array construction // In 30th International Colloquium on Automata, Languages and Programming, number 2719 in LNCS. 2003. С. 943–955.



Бакулина Марина Павловна — канд. физ.-мат. наук, научный сотрудник Института вычислительной математики и математической геофизики (ИВМиМГ СО РАН). Научные интересы: теория кодиро-

вания, сжатие данных, моделирование информационных систем, e-mail: marina@rav.sssc.ru.

Bakulina Marina Pavlovna is a Researcher of Institute of Computational Mathematics and Mathematical Geophysics (ICMMG SB RAS). PhD in Physics and Mathematics. Research interests: coding theory, data compression, information systems modeling.

Дата поступления — 17.08.2025

COMMUNITY DETECTION IN THE MULTIPLEX NETWORK OF SCIENTIFIC JOURNAL AUTHORS

S. V. Bredikhin, N. G. Scherbakova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-11-24

EDN: TWNDJO

The article presents a model of a multiplex network reflecting a real scheme of collaboration of authors of a scientific journal. The initial data are extracted from the XML archive of the journal articles. The model is presented as a two-layer graph, the vertices of which correspond to the authors of the articles, and the edges — to binary relations of co-authorship and citation. The purpose of the work is to identify non-intersecting communities of authors of the network and is achieved in two stages. At the first phase, the network is reduced to the form of an undirected graph G_f using “flattening algorithm”, this allows to apply the known algorithms of community detection intended for single-layer networks. So, at the second phase, two traditional clustering algorithms, Walktrap and Infomap, based on the “random walk” method are applied to the flattened network. The result of the algorithm is a set of identifiers of nodes included in the community, by which the original multilayer network structure can be restored. The comparison of the algorithms’ results is made using the *rand*, *adjusted_rand* and *nmi* indices. For G_f , all indices demonstrate a high level of similarity in the algorithms’ results. The parameter ρ_c , which characterizes the degree of overlapping of layers at the community level, serves as a characteristic of multilayering. The study of this parameter showed that most communities have a zero value, i.e. the communities consist of vertices of one of the layers. It should be noted that these are the actors of the co-authorship layer, while the citation layer contains 37 % of single nodes. The results of the analysis are presented in the form of tables.

Key words: multiplex network, scientific community, co-authorship, citation, modularity, bibliometrics.

References

1. Barabási A-L., Pósfai M. Network Science. Cambridge Univ. Press. 456 p. ISBN 1107076269.
2. Radicchi F., Castellano C., Cecconi F., Loreto V., Parisi D. Defining and identifying communities in networks // PNAS. 2004. V. 101. P. 2658–2663. DOI: 10.1073/pnas.0400054101.
3. Fortunato S. Community detection in graphs // Phys. Rep. 2010. V. 486, iss. 3–5. P. 75–174. DOI: 10.1016/j.physrep.2009.11.002.
4. Distel’ R. Teoriya grafov. Novosibirsk: Izd-vo In-ta matematiki, 2002. 336 s. ISBN 5-86134-101-X.
5. Peel L., Larremore D. B., Clauset A. The ground truth about metadata and community detection in networks // Sci. Adv. 2017. V. 3, iss. 5. e1602548. DOI: 10.1126/scadv.1602548.
6. Newman M. E. J. Modularity and community structure in networks // Proc. Natl. Acad. Sci. USA. 2006. V. 103. P. 8577–8582. DOI: 10.1037/pnas.0601602103.

This work was carried out under state contract with ICMG SB RAS (FWNM-2025-0005).

7. Newman M. E. J., Girvan M. Finding and evaluating community structure in networks // *Phys. Rev. E*. 2004. V. 69. 026113. DOI: 10.1103/PhysRevE.69. 026113.
8. Magnani M., Hanteer O., Interdonato R., Rossi L., Tagarelli A. Community detection in multiplex networks // *arXiv*: 0911.1824. DOI: 10.48550/arXiv.1910.07646.
9. Interdonato R., Tagarelli A., Ienco D., Sallaberry A., Poncelet P. Node-centric community detection in multilayer networks with layer-coverage diversification bias // *Proc. of the 8th Conf. on Complex Networks*. 2017. P. 57–66. Springer Intern. Publ., 2017. DOI: 10.48550/arXiv.1704.03441.
10. Jeub L. G. S., Mahoney M. W., Mucha P. J., Porter M. A. A local perspective on community structure in multilayer networks // *Network Sci.* 2017. V. 5, iss.2. P. 144–163. DOI: 10.48550/arXiv.1510.05185.
11. Kim J., Lee J-G. Community detection in multi-layer graphs: A survey // *ACM SIGMOD Record*. 2015. V. 44, iss. 3. P. 37–48. DOI: 10.1145/2854006.2854012.
12. Huang X., Chen D., Ren T., Wang D. A survey of community detection methods in multilayer networks // *Data Mining and Knowledge Discovery*. 2021. V. 35. P. 1–45. DOI: 10.1007/s10618-020-00716-6.
13. Mucha P. J., Richardson T., Macon K., Porter M. A., Onnela J. P. Community structure in time-dependent, multiscale, and multiplex networks // *Science*. 2010. V. 328, iss. 5980. P. 876–878. DOI: 10.1126/science.1184819.
14. De Domenico M., Lancichinetti A., Arenas A., Rosvall M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems // *Phys. Review*. 2015. X 5, 011027. DOI: 10.1103/PhysRevX.5. 011027.
15. Afsarmanesh N., Magnani M. Finding overlapping communities in multiplex networks // *Proc. of the 2018 Intern. conf. on Social Informatics*, 2018. DOI: 10.48550/arXiv.1602.03746.
16. Bianconi G. *Multilayer networks. Structure and functions*. Oxford. 2018. Online ISBN: 9780191815676.
17. Lancichinetti A., Fortunato S. Consensus clustering in complex networks // *Sci. Rep.* 2012. V. 2. Art. num. 336. DOI: 10.1038/srep00336.
18. Mondragon R. J., Iacovacci J., Bianconi G. Multilink communities of multiplex networks // *arXiv*:1706.09011. DOI: 10.48550/arXiv.1706.09011.
19. De Domenico M., Solé-Ribalta A., Cozzo E., Kivelä M., Moreno Y., Porter M. A., Gómez S., Arenas A. Mathematical formulation of multilayer networks // *Phys. Rev.* 2013. X 3. 041022. DOI:10.1103/PhysRevX.3.041022.
20. Bredihin S. V., Shcherbakova N. G. Vzveshennaya mul'tipleksnaya set' avtorov nauchnogo zhurnala // *Probl. inform.* 2025. N 1. S. 45–59. DOI: 10.24412/2073-0667-2025-1-45-59.
21. Bredihin S. V., Shcherbakova N. G. Strukturnye svojstva mul'tipleksnoj seti avtorov nauchnogo zhurnala // *Probl. inform.* 2025. N 2. S. 8–18. DOI: 10.24412/2073-0667-2025-2-5-18.
22. Boccaletti S., Bianconi G., Criado R., del Genio C. I., Gómez-Gardeñes J., Romance M., Sendiña-Nadal I., Wang Z., Zanin M. The structure and dynamics of multilayer networks // *Phys. Rep.* 2014. V. 544, iss. 1. P. 1–122. DOI: 10.1016/j.physrep.2014.07.001.
23. Wagner S., Wagner D. Comparing clusterings — An overview. 2007. DOI: 10.5445/IR/1000011477. <https://publikationen.bibliothek.kit.edu/1000011477>.
24. Collins L. M., Dent C. W. Omega: A general formulation of the Rand index of cluster recovery suitable for non-disjoint solutions // *Multivariate Behav. Res.* 1988. V. 23, iss. 2. P. 231–242. DOI: 10.1207/s15327906mbr2302_6.
25. Murray G., Carenini G., Ng R. Using the omega index for evaluating abstractive community detection // *Proc. of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, Montréal (Canada), 2012. Assoc. for Comput. Linguistics. P. 10–18.
26. Hanteer O., Rossi L. The meaning of dissimilar: An evaluation of various similarity quantification approaches used to evaluate community detection solutions // *Proc. of the IEEE/ACM*

Intern. conf. on Advances in Social Networks Analysis and Mining, Vancouver (Canada), 2019. P. 513–518. DOI: 10.1145/3341161.3342941.

27. Berlingerio M., Coscia M., Giannotti F. Finding and characterizing communities in multidimensional networks // Intern. conf. on Advances in Social Networks Analysis and Mining (ASONAM). P. 490–494. IEEE Computer Society Washington, DC, USA, 2011. DOI: 10.1107/ASONAM.2011.104.

28. Kim J., Lee J.-G., Lim S. Differential flattening: A novel framework for community detection in multi-layer graphs // ACM Trans. on Intell. Syst. and Technol. (TIST). 2016. V. 8, iss. 2. P. 27:1–27:23. DOI: 10.1145/2898362.

29. De Domenico M., Nicosia V., Arenas A., Latora V. Structural reducibility of multilayer networks // Nature Commun. 2015. V. 6. 6864. DOI: 10.1038/ncomms7864.

30. Bianconi G. Statistical mechanics of multiplex networks: entropy and overlap // Phys. Rev. E. 2013. V. 87, iss. 6. 062806. DOI: 10.1103/PhysRevE.87.062806.

31. Pons P., Latapy M. Computing communities in large networks using random walks. 2006. arXiv: physics/0512106. DOI: 10.48550/arXiv.physics/0512106.

32. Rosvall M., Axelsson D., Bergstrom C. T. Map equation. // Eur. Phys. J. 2009. V. 178. P. 13–23. DOI: 10.1140/epjst/e2010-01179-1.

ВЫЯВЛЕНИЕ СООБЩЕСТВ В МУЛЬТИПЛЕКСНОЙ СЕТИ АВТОРОВ НАУЧНОГО ЖУРНАЛА

С. В. Бредихин, Н. Г. Щербакова

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.177

DOI: 10.24412/2073-0667-2025-4-11-24

EDN: TWNDJO

Представлена модель мультиплексной сети, отражающая реальную схему сотрудничества авторов научного журнала. Исходные данные извлечены из XML-архива статей журнала. Модель выполнена в виде двухслойного графа, вершины которого соответствуют авторам статей, а ребра — бинарным отношениям соавторства и цитирования. Цель работы состоит в выявлении непересекающихся сообществ авторов сети и достигается в два этапа. На первом этапе сеть приводится к виду неориентированного графа, на втором к построенному графу применяются два традиционных алгоритма кластеризации, основанные на методе случайного блуждания. Выполнен вычислительный эксперимент.

Ключевые слова: мультиплексная сеть, научное сообщество, соавторство, цитирование, модульность, библиометрия.

Введение. Выявление сетевых сообществ — важный шаг современного сетевого анализа, результаты которого позволяют строить обоснованные предположения о топологии изучаемого сетевого объекта и измерять его параметры. Термин *сетевое сообщество* (*кластер*) не имеет общепринятого определения и интуитивно определяется как группа узлов (*акторов*), которые более плотно связаны друг с другом, чем с другими узлами в сети. Определения сообществ опираются на следующие гипотезы А.-Л. Барабаши [1]. Фундаментальная гипотеза: «*Структура сообщества однозначно закодирована в схеме соединения ее узлов*». Гипотеза связности: «*Сообщество соответствует связному подграфу*». Гипотеза плотности: «*Сообщество является локально плотным подграфом*». В этом контексте полный подграф (клика) является сообществом.

Другой способ определения опирается на разделение степеней каждого узла, принадлежащего сообществу, на внутреннюю степень, означающую число ребер с узлами того же сообщества, и внешнюю, означающую число ребер с узлами других сообществ. При этом сообщество называют *сильным*, если каждый узел имеет большее число ребер внутри сообщества, нежели вне его. Сообщество называют *слабым*, если его общая внутренняя степень превышает его общую внешнюю степень [2]. Таким образом, каждое сильное сообщество является слабым, обратное неверно.

Задача выявления сообществ весьма трудоемка и на сегодняшний день не имеет универсального решения. В обзоре [3] представлены определения основных элементов проблемы

Исследования выполнены в рамках государственного задания ИВМиМГ СО РАН (FWNM-2025-0005).

и ряд разработанных методов их решения. Один из традиционных методов выявления сообществ — разбиение сети на заранее определенное число групп приблизительно одинакового размера, так что число ребер между группами минимально. В случае, когда число групп равно двум, процедура кластеризации называется *разрезом*. Основополагающая теорема Менгера (доказана в 1927 г.) о минимальном разрезе, характеризующая связность неориентированного либо ориентированного графа, утверждает, что минимальное число ребер, удаление которых разъединяет узлы i и j , равно максимальному числу попарно непересекающихся путей из i в j [4].

Учитывая актуальность проблемы, крайне важно построить эффективные процедуры и выбрать соответствующие алгоритмы для идентификации структуры сетевых сообществ. В большинстве случаев сообщества определяются алгоритмически, т. е. они являются конечным продуктом алгоритма, без формального определения [3]. Производительность алгоритмов обнаружения сообществ зачастую оценивают по их способности находить так называемые истинные сообщества. Это широко применяется в синтетических сетях, тогда как в реальных сетях заранее известных сообществ нет. На практике принято обрабатывать дополнительную информацию об узлах сети — *метаданные*. Этот прием следует использовать осторожно, поскольку не существует принятого определения того, каким должен быть правильный результат [5]. Популярной функцией определения качества разбиения на сообщества является *модульность* [6]. В ее основе лежит гипотеза о том, что случайный граф не имеет кластерной структуры. Существование кластеров выявляется путем сравнения фактической плотности ребер в кластере и плотности, которой можно было бы ожидать, если бы вершины графа были связаны независимо от структуры сообществ. Модульность также может выступать в качестве оптимизирующей функции алгоритма кластеризации [7].

Мультиплексная сеть — это модель, используемая для представления реальных объектов с несколькими типами отношений между акторами. Каждый тип отношений представлен отдельным слоем, моделируемым графом. Традиционные методы обнаружения сообществ для обычных графов недостаточны для сложной мультиплексной модели по следующим причинам [8]. Во-первых, без возможности анализа подмножеств слоев некоторые сообщества могут оказаться скрытыми из-за рассмотрения слоев, не имеющих к ним отношения. Во-вторых, алгоритмы, не представляющие различные слои явно, не могут различать различные типы мультиплексных сообществ, например присутствующие только в одном слое или состоящие из определенных комбинаций слоев. В-третьих, без концепции слоя невозможно включить один и тот же узел в разные сообщества в зависимости от слоя, на котором узел активен. Для устранения вышеуказанных ограничений был разработан ряд алгоритмов обнаружения сообществ для мультиплексных сетей, основанных на различных определениях сообщества и различных вычислительных методах. В мультиплексных сетях, где узлы связаны несколькими типами бинарных отношений, возникли новые проблемы в обнаружении сообществ. Современный сетевой анализ появился как перспективный подход к исследованию подобных сетей.

Результатом алгоритмов выявления сообществ в мультиплексных сетях является множество кластеров, каждый из которых содержит непустое подмножество множества узлов. При этом кластеры могут охватывать несколько слоев. Для выявления многослойной природы сообществ были разработаны различные методы. В работе [8] рассматриваются три уровня классификации методов. Верхний уровень различает *глобальные* (выявление всех сообществ в исследуемой сети) или *локальные* (одно сообщество вокруг одного или

нескольких выбранных узлов) методы (см. [9, 10]). Второй уровень связан со *способом*, которым алгоритмы обрабатывают наличие нескольких слоев: *сглаживание* (*flattening*) — сведение к одному слою; *послойная кластеризация* — обработка каждого слоя независимо для последующего объединения результатов обработки; *многослойная кластеризация* — рассмотрение всех слоев одновременно (см. обзоры [8, 11, 12]). Последний уровень классификации группирует алгоритмы, основанные на конкретных методах, таких как оптимизация целевой функции [13], случайное блуждание [14] или выявление плотных подграфов [15]. Классификация конкретных методов многослойной кластеризации приведена в работе [16], при этом рассматриваются пять основных классов: максимизация многослойной модульности [13]; консенсусная кластеризация [17]; кластеризация на основе свойств многослойных случайных блужданий [14]; кластеризация на основе мультилинков [18]; методы, основанные на тензорном разложении и соответствующих вычислениях [19].

В данной статье представлена двухслойная мультиплексная сеть $\mathcal{M}^{cc}$, моделирующая реальную схему сотрудничества авторов научного журнала. Произведена кластеризация узлов сети с использованием метода сглаживания. Апробированы два алгоритма выявления сообществ, основанные на методе случайного блуждания. Выполнена оценка качества разбиения. Данная работа является продолжением работ [20, 21].

1. Мультиплексная сеть. Многослойная сеть [22] — это пара $\mathcal{M} = (G, C)$, в которой $G = \{G_\alpha, \alpha \in \{1, \dots, M\}\}$ — семейство ориентированных (либо неориентированных), взвешенных (либо невзвешенных) графов $G_\alpha = (V_\alpha, E_\alpha)$, называемых слоями, а $C = \{E_{\alpha\beta} \subseteq V_\alpha \times V_\beta; \alpha, \beta \in \{1, \dots, M\}, \alpha \neq \beta\}$ — множество взаимосвязей между узлами различных слоев G_α и G_β , $\alpha \neq \beta$. Элементы E_α называются *внутрислойными* ребрами, а элементы $E_{\alpha\beta}$ — *межслойными*.

Мультиплексная сеть $\mathcal{M}$ является особым видом многослойной сети, в которой $V_1 = V_2 = \dots = V_M = V$ и межслойные ребра разрешены только между репликами одних и тех же физических узлов, т. е. $\forall \alpha, \beta \in \{1, \dots, M\}, \alpha \neq \beta$ ($E_{\alpha\beta} = \{(v, v), v \in V\}$).

Проекцией мультиплексной сети называют граф $proj(\mathcal{M}) = (V_M, E_M)$, элементы матрицы смежности которого определяются следующим образом:

$$a_{ij} = \begin{cases} 1, & \text{если } a_{ij}^\alpha = 1, 0 \leq \alpha \leq M \\ 0 & \text{в противном случае.} \end{cases} \quad (1)$$

В статье многослойная сеть определяется кортежем (A, L, V, E) , где A — множество акторов; $L = \{l_1, l_2, \dots, l_m\}$ — множество слоев; (V, E) — граф на $V \subseteq A \times L$. Это определение не требует, чтобы все акторы присутствовали во всех слоях. В мультиплексной сети множество E ограничено внутриуровневыми ребрами, т. е. ребро $((v_1, l_1), (v_2, l_2)), l_1, l_2 \in L$ существует только если $l_1 = l_2$, а множество ребер $E = \{E_1, E_2, \dots, E_m\}$.

2. Сетевые сообщества. С точки зрения покрытия различают *полную* (*total*) кластеризацию (каждый узел принадлежит по крайней мере одному сообществу) и *частичную* (*partial*) кластеризацию (не все узлы распределены по сообществам). Кластеризацией с *перекрыванием узлов* (*node-overlapping*) называется кластеризация, в которой существует по крайней мере один узел, принадлежащий более чем одному кластеру, в противном случае это кластеризация *без перекрывания узлов* (*node-disjoined*). Аналогично, если есть по крайней мере один актор, принадлежащий более чем одному кластеру, кластеризация называется кластеризацией с *перекрыванием акторов* (*actor-overlapping*), в противном случае кластеризацией *без перекрывания акторов* (*actor-disjoined*). Заметим, что кластеризация с перекрыванием узлов является также кластеризацией с перекрыванием акторов, однако кла-

стеризация с перекрытием акторов не обязательно является кластеризацией с перекрытием узлов. Наконец, мультиплексное сообщество называется *полуколонной* (*semi-pillar*) на слоях $L' \subseteq L$, если для каждого актора $a \in A$ в сообществе все узлы $\{(a, l) \in V; l \in L'\}$ входят в сообщество. Если $L' = L$, то это *колонна*. Два сообщества-колонны либо не пересекаются, либо пересекаются как по акторам, так и по узлам.

2.1. *Качество кластеризации.* Проверка качества кластеризации — нетривиальная задача даже для однослойных сетей, более того, не существует единого общепринятого определения того, каким должен быть правильный результат. Так, в работе [5] показано, что метаданные (например, аффилиация авторов) — это не то же самое, что «абсолютная истина», и что обращение с ними как с таковыми вызывает серьезные теоретические и практические проблемы. Обнаруженные сообщества и метаданные фиксируют различные аспекты структуры сети. В работе [8] подчеркнуто, что в случае мультиплексной сети может оказаться, что не все слои вносят положительный вклад в определение структуры сообществ. А тот факт, что большинство методов обнаружения сообществ всегда дают выход независимо от того, какие слои включены во входную мультиплексную сеть, делает включение большего количества входных слоев потенциально проблематичным. Слои, помимо того что определяются внутренней топологией, также определяются внутренней логикой, которая может быть согласованной или не согласованной с логикой других слоев.

2.2. *Сравнение результатов кластеризации.* Для измерения парного сходства между результатами глобальной кластеризации могут использоваться традиционные критерии оценки разбиений, такие как индексы *Rand*, *Adjusted_Rand* или *NMI* [23]. Однако эти критерии применимы только в случае неперекрывающихся сообществ. Индекс *Omega* [24] является мерой, которую можно применять для сравнения результатов разбиения в ситуациях, когда оба результата разбиения содержат перекрывающиеся сообщества, один содержит или ни один не содержит [25, 26].

2.3. *Метод сглаживания (flattening)* [27, 28]. В общем случае метод регламентирует процесс формального представления взаимодействия акторов в обозримом формате, например табличном. В нашем случае применяется с целью преобразования многослойной сети в однослойную сеть (возможно, взвешенную) путем игнорирования информации, характерной для индивидуальных слоев, т. е. не принимая во внимание информацию о типе ребер. Его применение позволяет использовать традиционные алгоритмы обнаружения сообществ.

Метод *полного сглаживания* позволяет конструировать невзвешенные графы, где два узла смежны, если их соответствующие акторы смежны в любом из входных слоев, т. е. для мультиплекса это построение проекции сети (1). Преимущество этого метода заключается в том, что существует большое число хорошо апробированных алгоритмов кластеризации для невзвешенных графов, а при наличии взвешенных ребер зачастую подразумевается дополнительный уровень сложности. В свою очередь, *взвешенное сглаживание* отражает важные структурные свойства исходной мультиплексной сети в форме весов, назначенных ребрам. Например, один из способов назначения веса, в котором параметр вес означает число слоев, в которых присутствует данное ребро: $w(u, v) = |\{i, (u, v) \in E_i\}|$. В работах [27, 28] приведены также другие способы определения весов. Альтернативный подход к полному сглаживанию, состоящий из компромисса между оригинальной многослойной сетью и сетью, полученной полным сглаживанием, приведен в работе [29].

Методы сглаживания приводят к образованию сообществ-колонн, так как акторы из разных слоев представлены одним узлом в сглаженном графе (далее — G_f). В результа-

те работы алгоритма могут возникнуть сообщества, которые не являются сообществом в каком-либо отдельном слое. Сложность алгоритма в данном случае соответствует сложности шага обнаружения сообществ. Более высокое сходство слоев, например с точки зрения присутствия ребер между одноименными акторами, может повлечь меньшее время выполнения однослойного алгоритма.

2.4. Оценка результатов кластеризации. В однослойной сети плотность сообщества определяется соотношением между числом существующих ребер среди узлов сообщества и числом всех возможных ребер. В многослойной сети плотность может обеспечиваться одним слоем, а может — несколькими. В работе [27] определен параметр ρ_c , характеризующий сообщества многослойной сети. Пусть s — сообщество, сформированное алгоритмом поиска сообществ в многослойной сети. Обозначим M — множество слоев; P — множество пар (u, v) , связанных по крайней мере в одном слое сети; P^2 — множество пар, связанных по крайней мере в двух слоях; P_c — подмножество пар из P , входящих в сообщество s . Рассмотрим подмножество пар $P_c^2 \subseteq P^2$, состоящее из пар множества P^2 , входящих в s . Обозначим ребро, принадлежащее слою $m \in M$, триадой (u, v, m) ; E — множество всех ребер исходной сети.

Определяется мера (2), названная *избыточностью*, отражающая ситуацию, при которой множество узлов, составляющих сообщество в одном слое, составляет сообщество также и в других слоях. Эта мера может рассматриваться как индикатор избыточности ребер: чем больше слоев соединяют каждую пару узлов в сообществе, тем выше будет избыточность. Теперь можно определить *нормализованную* избыточность, подсчитав число пар, имеющих избыточные связи, нормализуя по теоретическому максимуму:

$$\rho_c = \sum_{(u, v) \in P_c^2} \frac{|m : \exists (u, v, m) \in E|}{|M| \cdot |P_c|}. \quad (2)$$

Параметр ρ_c принимает значения в диапазоне $[0, 1]$. Поскольку для сообществ, состоящих из одного элемента, $|\rho_c| = 0$, такие сообщества не рассматриваются. По существу, параметр ρ_c подобен мере перекрытия слоев на уровне сообществ [30].

3. Вычислительный эксперимент.

3.1. Данные. Взвешенная двухслойная мультиплексная сеть авторства M^{cc} построена на основе данных, извлеченных из записей электронного XML-архива научного журнала «Сахарный диабет» (ISSN 2072-0378), находящегося в свободном доступе. Рассматривается множество статей P , опубликованных в журнале, $|P| = 991$, и множество авторов A , $|A| = 1694$, участвовавших в их создании. В свою очередь, P представлено объединением множеств P^m — статей, созданных в соавторстве, и множества P^s — индивидуальных статей (в нашем случае $|P^m| = 843$, $|P^s| = 148$). Аналогично, множество авторов A представлено объединением множеств $A = A^m \cup A^s$, где A^m — подмножество авторов, имеющих совместные публикации, A^s — имеющих исключительно индивидуальные публикации (в нашем случае $|A^m| = 1676$, $|A^s| = 18$). Для вычислений использовался пакет *igraph* на основе *R*.

В качестве узлов сети выступают авторы статей. Первый слой (l_1) — граф соавторства $G_1 = (V, E_1)$, $|V| = |A|$, узлы соответствуют авторам; ребро $(u, v, 1)$ между двумя узлами устанавливается в случае, если соответствующие авторы имеют совместные публикации; вес ребра равен числу публикаций. Второй слой (l_2) — ориентированный граф цитирования $G_2 = (V, E_2)$; ребро $(u, v, 2)$ между цитирующим и цитируемым устанавливается,

Таблица 1

Параметры слоев сети $\mathcal{M}^{cc}$ и графа G_f

Граф	n	m	nc	mcs	$dens$	cc	apl	dia
G_1	1676	16418	96	1112	0,005848324	0,3627196	3,781167	11
G_2	1075	21171	763	305	0,018337014	0,2198572	2,815648	7
G_f	1682	24134	49	1484	0,017071126	0,2437155	2,607897	7

если в списке литературы статьи, в которой участвует цитирующий автор, присутствует ссылка на статью из журнала, в которой участвует цитируемый автор. Детали построения приведены в работе [20]. В качестве сглаженной сети рассматривается невзвешенный граф (сглаживания) $G_f = (V, E)$, в котором два узла смежны, если их соответствующие акторы смежны хотя бы в одном слое, ориентация второго слоя не учитывается.

Параметры слоев и графа G_f приведены в табл. 1. Обозначения: n — число узлов; m — число ребер; nc — число компонент; mcs — размер максимальной компоненты; $dens$ — реберная плотность; cc — коэффициент кластеризации (глобальный — соотношение числа треугольников к числу связных троек); apl — средняя длина пути; dia — диаметр. Одиночные узлы не включены в состав слоев сети.

3.2. *Алгоритмы.* Рассмотрим два алгоритма, использующие методы, основанные на случайных блужданиях. Они предполагают, что актор, случайным образом следующий по ребрам в сети, будет иметь тенденцию попадать в ловушку внутри сообществ из-за более высокой плотности ребер между узлами внутри одного сообщества, реже перемещаясь из одного сообщества в другое. Результатом работы алгоритмов является множество непересекающихся сообществ, при этом каждый актор попадает в какое-либо сообщество.

3.2.1. Алгоритм *Walktrap*, представленный в работе [31], выявляет плотные подграфы исходного графа на основе случайного блуждания. Агломеративный алгоритм объединяет узлы в кластеры на основе определенной в работе меры сходства (расстояния) между узлами. На начальном этапе каждый узел является кластером. Далее, по критерию расстояния, на каждом этапе два соседних кластера объединяются в новый кластер, образуя новое разбиение узлов. Пересчитываются расстояния. Ключевой момент алгоритма — способ выбора сообществ для слияния и тот факт, что расстояния можно эффективно обновлять. Сложность алгоритма оценивается $O(mn^2)$, где n и m — число узлов и ребер соответственно.

Выбор результирующего разбиения производится на основе модульности. Модульность Q — это функция качества, которая сравнивает число ребер между узлами сообществ с числом, которое можно было бы ожидать при группировании случайным образом. То есть для идентификации сообществ в сети требуется найти разделение, которое максимизирует модульность. Если вероятность того, что два узла связаны ребром в графе, определяется равенством $p_{ij} = \frac{k_i \cdot k_j}{\langle k \rangle \cdot N}$, а c_i — это сообщество, к которому принадлежит узел i , то параметр модульность Q определяется равенством

$$Q = \frac{1}{\langle k \rangle \cdot N} \sum_{ij} (a_{ij} - p_{ij}) \cdot \delta(c_i, c_j), \quad (3)$$

где δ — дельта Кронекера; a_{ij} — элемент матрицы смежности графа; k_i — степень узла i ; $\langle k \rangle$ — среднее значение степени узлов [6].

Таблица 2

Статистика структуры сообществ

Граф \ Параметры	K_1	K_2	K_3	K_4	Q
Алгоритм <i>Walktrap</i>					
G_f	230	552	4	46	0,2873692
G_1	259	305	3	15	0,5499898
G_2	189	433	2	143	0,1195941
Алгоритм <i>Infomap</i>					
G_f	146	458	8	0	0,3215479
G_1	211	123	3	0	0,5809985
G_2	16	969	0	0	0,0218763

Таблица 3

Индексы сравнения результатов разбиений графов с использованием алгоритмов *Walktrap* и *Infomap*

Граф \ Индекс	G_f	G_1	G_2
<i>rand</i>	0,9475448	0,9686040	0,3534814
<i>adjustedrand</i>	0,7048701	0,3995592	0,0837723
<i>nmi</i>	0,8706313	0,8995698	0,2737676

3.2.2. Алгоритм *Infomap* [32] идентифицирует составляющие сеть модули путем эффективного описания процесса распространения информации по сети. Основан на идее случайного блуждания по сети с целью минимизировать длину пути обхода узлов, что позволяет выявлять сообщества на основе сетевых потоков информации. Выделяется группа узлов, в которой обмен информацией осуществляется достаточно быстро, узлы этой группы объединяются, далее она выступает как единый модуль. Связи между модулями, построенными подобным образом, отражают сетевые информационные потоки. Для описания этого процесса применяется эффективный код Хаффмана, позволяющий генерировать уникальные имена узлов. Этот прием позволяет построить карту сети, в результате случайного блуждания (путешествия) по которой можно выделить модули, в которых «путешественник» провел большую часть времени. Вычислительная сложность алгоритма $O(n \log n)$.

3.3. *Результаты разбиения.* Для сравнения приводятся результаты разбиения на сообщества для графа G_f , а также для слоев, представленных графами G_1 и G_2 . Результаты работы алгоритмов *Walktrap* и *Infomap* представлены в табл. 2.

В заголовках табл. 2 используются следующие сокращения: K_1 — число кластеров; K_2 — размер максимального кластера; K_3 — число кластеров с числом узлов меньше максимального и больше 30; K_4 — число одиночных кластеров.

При разбиении на сообщества взвешенного ориентированного графа G_2 с помощью алгоритма *Infomap* приблизительно 90 % узлов попало в максимальное сообщество. Из 458 узлов, входящих в максимальное сообщество при разбиении графа сглаживания G_f ,

Таблица 4
Значения параметра ρ_c

<i>Infomap</i>			<i>Walktrap</i>		
$c(\cdot)$	n	ρ_c	$c(\cdot)$	n	ρ_c
106	2	1,000	26	2	1,000
55	10	0,756	63	2	1,000
63	10	0,600	14	10	0,756
11	7	0,500	42	7	0,583
72	9	0,438	20	8	0,529
22	26	0,244	61	5	0,400
6	10	0,211	58	15	0,365
2	69	0,202	19	7	0,357
73	11	0,185	12	17	0,239
29	12	0,179	35	9	0,222
38	33	0,166	7	62	0,221
35	34	0,163	6	69	0,202
25	26	0,160	9	32	0,167
1	458	0,152	18	12	0,158
5	31	0,113	1	552	0,147
10	51	0,087	5	74	0,083
59	31	0,078	53	13	0,014
32	10	0,077			
9	31	0,068			

114 узлов попало в максимальное сообщество при кластеризации G_1 (из 123), и 420 входят в максимальное при кластеризации G_2 (из 969).

Сравнение результатов разбиений с помощью алгоритмов *Walktrap* и *Infomap* представлено в табл. 3. Наиболее стабильные результаты у графа сглаживания G_f .

3.4. *Характеристика сообществ на основе параметра ρ_c* . В табл. 4 приведены ненулевые значения параметра ρ_c , характеризующего сообщества, выявленные с помощью алгоритмов *Infomap* и *Walktrap*. Здесь $c(\cdot)$ — номер сообщества, n — число узлов в сообществе. Для алгоритма *Infomap* ненулевое значение ρ_c имеют приблизительно 13 % кластеров, а для алгоритма *Walktrap* — 7,4 % кластеров. При этом в обоих случаях нулевое значение имеют в основном кластеры небольшого размера. Следует также заметить, что большое число нулевых значений объясняется тем, что в слое цитирования 37 % акторов представлены одиночными вершинами.

Отметим, что кластеры наибольшего размера (458 и 552), полученные с использованием алгоритмов *Infomap* и *Walktrap*, пересекаются по 436 узлам.

Заключение. Рассмотрен один из методов выявления непересекающихся сообществ многослойной сети в применении к двухслойной мультиплексной сети авторов научного журнала. Он осуществляет построение отображения многослойной сети на однослойную с применением метода сглаживания. Это позволило использовать известные алгоритмы выявления сообществ, предназначенные для однослойных сетей. Результатом работы алгоритма является множество идентификаторов узлов, входящих в сообщество, по которым можно восстановить исходную многослойную сетевую структуру. В качестве характеристики многослойности выступает параметр ρ_c , характеризующий степень перекрытия сло-

ев на уровне сообществ. Рассмотрены два алгоритма выявления сообществ, основанные на технике свободного блуждания — *Walktrap* и *Infomap*. Сравнение результатов работы алгоритмов проведено с использованием индексов *rand*, *adjusted_rand* и *nmi*. Для G_f все индексы демонстрируют высокий уровень сходства результатов работы алгоритмов. Использование параметра ρ_c показало, что большинство сообществ имеют нулевое значение параметра, т. е. сообщества состоят из вершин одного из слоев. В дальнейшем предполагается апробировать методы послойной и многослойной кластеризации к анализируемой сети.

Авторы благодарят А. В. Феофанова за техническую поддержку на этапе выполнения вычислительного эксперимента.

Список литературы

1. Barabási A-L., Pósfai M. Network Science. Cambridge Univ. Press. 456 p. ISBN 1107076269.
2. Radicchi F., Castellano C., Cecconi F., Loreto V., Parisi D. Defining and identifying communities in networks // PNAS. 2004. V. 101. P. 2658–2663. DOI: 10.1073/pnas.0400054101.
3. Fortunato S. Community detection in graphs // Phys. Rep. 2010. V. 486, iss. 3–5. P. 75–174. DOI: 10.1016/j.physrep.2009.11.002.
4. Дистель Р. Теория графов. Новосибирск: Изд-во Ин-та математики, 2002. 336 с. ISBN 5-86134-101-X.
5. Peel L., Larremore D. B., Clauset A. The ground truth about metadata and community detection in networks // Sci. Adv. 2017. V. 3, iss. 5. e1602548. DOI: 10.1126/scadv.1602548.
6. Newman M. E. J. Modularity and community structure in networks // Proc. Natl. Acad. Sci. USA. 2006. V. 103. P. 8577–8582. DOI: 10.1037/pnas.0601602103.
7. Newman M. E. J., Girvan M. Finding and evaluating community structure in networks // Phys. Rev. E. 2004. V. 69. 026113. DOI: 10.1103/PhysRevE.69. 026113.
8. Magnani M., Hanteer O., Interdonato R., Rossi L., Tagarelli A. Community Detection in Multiplex Networks // arXiv: 0911.1824. DOI: 10.48550/arxiv: 0911.1824.
9. Interdonato R., Tagarelli A., Ienco D., Sallaberry A., Poncelet P. Node-centric community detection in multilayer networks with layer-coverage diversification bias // Proc. of the 8th Conf. on Complex Networks. 2017. P. 57–66. Springer Intern. Publ., 2017. DOI: 10.48550/arXiv.1704.03441.
10. Jeub L. G. S., Mahoney M. W., Mucha P. J., Porter M. A. A local perspective on community structure in multilayer networks // Network Sci. 2017. V. 5, iss. 2. P. 144–163. DOI: 10.48550/arXiv.1510.05185.
11. Kim J., Lee J-G. Community detection in multi-layer graphs: A survey // ACM SIGMOD Record. 2015. V. 44, iss. 3. P. 37–48. DOI: 10.1145/2854006.2854012.
12. Huang X., Chen D., Ren T., Wang D. A survey of community detection methods in multilayer networks // Data Mining and Knowledge Discovery. 2021. V. 35. P. 1–45. DOI: 10.1007/s10618-020-00716-6.
13. Mucha P. J., Richardson T., Macon K., Porter M. A., Onnela J. P. Community structure in time-dependent, multiscale, and multiplex networks // Science. 2010. V. 328, iss. 5980. P. 876–878. DOI: 10.1126/science.1184819.
14. De Domenico M., Lancichinetti A., Arenas A., Rosvall M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems // Phys. Review. 2015. X 5, 011027. DOI: 10.1103/PhysRevX.5. 011027.

15. Afsarmanesh N., Magnani M. Finding overlapping communities in multiplex networks // Proc. of the 2018 Intern. conf. on Social Informatics, 2018. DOI: 10.48550/arXiv.1602.03746.
16. Bianconi G. Multilayer networks. Structure and functions. Oxford. 2018. Online ISBN: 9780191815676.
17. Lancichinetti A., Fortunato S. Consensus clustering in complex networks // Sci. Rep. 2012. V. 2. Art. num. 336. DOI: 10.1038/srep00336.
18. Mondragon R. J., Iacovacci J., Bianconi G. Multilink communities of multiplex networks // arXiv:1706.09011. DOI: 10.48550/arXiv.1706.09011.
19. De Domenico M., Solé-Ribalta A., Cozzo E., Kivelä M., Moreno Y., Porter M. A., Gómez S., Arenas A. Mathematical formulation of multilayer networks // Phys. Rev. 2013. X 3. 041022. DOI:10.1103/PhysRevX.3.041022.
20. Бредихин С. В., Щербакова Н. Г. Взвешенная мультиплексная сеть авторов научного журнала // Пробл. информ. 2025. № 1. С. 45–59. DOI: 10.24412/2073-0667-2025-1-45-59.
21. Бредихин С. В., Щербакова Н. Г. Структурные свойства мультиплексной сети авторов научного журнала // Пробл. информ. 2025. № 2. С. 8–18. DOI: 10.24412/2073-0667-2025-2-5-18.
22. Boccaletti S., Bianconi G., Criado R., del Genio C. I., Gómez-Gardeñes J., Romance M., Sendiña-Nadal I., Wang Z., Zanin M. The structure and dynamics of multilayer networks // Phys. Rep. 2014. V. 544, iss. 1. P. 1–122. DOI: 10.1016/j.physrep.2014.07.001.
23. Wagner S., Wagner D. Comparing clusterings — An overview. 2007. DOI: 10.5445/IR/1000011477. <https://publikationen.bibliothek.kit.edu/1000011477>.
24. Collins L. M., Dent C. W. Omega: A general formulation of the Rand index of cluster recovery suitable for non-disjoint solutions // Multivariate Behav. Res. 1988. V. 23, iss. 2. P. 231–242. DOI: 10.1207/s15327906mbr2302_6.
25. Murray G., Carenini G., Ng R. Using the omega index for evaluating abstractive community detection // Proc. of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization, Montréal (Canada), 2012. Assoc. for Comput. Linguistics. P. 10–18.
26. Hanteer O., Rossi L. The meaning of dissimilar: An evaluation of various similarity quantification approaches used to evaluate community detection solutions // Proc. of the IEEE/ACM Intern. conf. on Advances in Social Networks Analysis and Mining, Vancouver (Canada), 2019. P. 513–518. DOI: 10.1145/3341161.3342941.
27. Berlingerio M., Coscia M., Giannotti F. Finding and characterizing communities in multidimensional networks // Intern. conf. on Advances in Social Networks Analysis and Mining (ASONAM). P. 490–494. IEEE Computer Society Washington, DC, USA, 2011. DOI: 10.1107/ASONAM.2011.104.
28. Kim J., Lee J.-G., Lim S. Differential flattening: A novel framework for community detection in multi-layer graphs // ACM Trans. on Intell. Syst. and Technol. (TIST). 2016. V. 8, iss. 2. P. 27:1–27:23. DOI: 10.1145/2898362.
29. De Domenico M., Nicosia V., Arenas A., Latora V. Structural reducibility of multilayer networks // Nature Commun. 2015. V. 6. 6864. DOI: 10.1038/ncomms7864.
30. Bianconi G. Statistical mechanics of multiplex networks: entropy and overlap // Phys. Rev. E. 2013. V. 87, iss. 6. 062806. DOI: 10.1103/PhysRevE.87.062806.
31. Pons P., Latapy M. Computing communities in large networks using random walks. 2006. arXiv: physics/0512106. DOI: 10.48550/arXiv.physics/0512106.

32. Rosvall M., Axelsson D., Bergstrom C. T. Map equation. // Eur. Phys. J. 2009. V. 178. P. 13–23. DOI: 10.1140/epjst/e2010-01179-1.



Бредихин Сергей Всеволодович — канд. техн. наук, ведущий научный сотрудник Ин-та вычислительной математики и математической геофизики СО РАН; e-mail:

bredikhin@sscc.ru.

Сергей Бредихин окончил механико-математический факультет Новосибирского государственного университета в 1968 г. С 1968 г. — сотрудник Института автоматизации и электрометрии СО РАН. Кандидат технических наук с 1983 г. С 1988 г. — заведующий Лабораторией прикладных систем Института вычислительной математики и математической геофизики СО РАН. Лауреат государственной премии по науке и технике 2012 г. Автор и соавтор многих научных статей, докладов и трех монографий.

Sergey Bredikhin graduated from Novosibirsk State University in 1968 (faculty of Mechanics and Mathematics). In 1968 he became an employee of Institute of Automation and Electrometry SB RAS. In 1983 he received PhD degree in Engineering Science. Since 1988 he is the head of Applied Systems laboratory of Institute of Computational Mathematics and Mathematical Geophysics SB RAS. He was the technical manager of “Akademgorodok Internet Project”. He is the state prize winner in science and engineering (2012).



Щербакова Наталия Григорьевна — ст. науч. сотр. Ин-та вычислительной математики и математической геофизики СО РАН; e-mail: scherbakova@sscc.ru.

Наталия Щербакова окончила Новосибирский государственный университет по специальности «Математическая лингвистика» в 1967 г. С 1967 г. работала в Институте математики СО РАН, затем в Институте автоматизации и электрометрии СО РАН в области создания программного обеспечения систем передачи данных. С 2000 г. — сотрудник Института вычислительной математики и математической геофизики СО РАН, где с 2002 г. занимает должность старшего научного сотрудника. Автор и соавтор более 60 работ и двух монографий.

Natalia Shcherbakova graduated from Novosibirsk State University in 1967 (mathematical linguistics). Since 1967 she worked at Institute of Mathematics SB RAS, then at Institute of Automation and Electrometry SB RAS in the field of software design for data transmission systems. In 2000 — the employee of Institute of Computational Mathematics and Mathematical Geophysics SB RAS, since 2002 works as senior researcher. She is the co-author of two monographs.

Дата поступления — 12.08.2025



COMPARISON OF FIRST, SECOND AND THIRD GENERATION GRAPH REDUCTION METHODS IN CHEMICAL KINETICS MODELS

A. R. Gerb, E. E. Deviatykh*, G. A. Omarova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

*Novosibirsk State University,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-25-37

EDN: VBFMRT

A detailed description of the mechanisms of chemical reactions of hydrocarbon oxidation includes many different pathways and sets of elementary reactions, which makes it difficult to use large-size models to calculate complex combustion phenomena. To address this problem, methods of reduction of chemical kinetic mechanisms are applied. In this paper, a comparative analysis of the performance of graph reduction methods of different generations is carried out. The initial kinetic mechanism is represented as an oriented graph, the nodes of which correspond to chemical substances, the arcs reflect the dependencies between substances. The advantage of these methods lies in their lower computational costs and their ability to form rather compact reduced models.

Key words: graph, reduction, chemical kinetics model, DRG, DRGEP, PFA, GPS.

References

1. Turanyi V. Reduction of large reaction mechanisms // *New J. Chemistry*. 1990. V. 14. N 11. P. 795–803.
2. Tomlin A. S., Pilling M. J., Turanyi T., Merkin J. H., Brindley J. Mechanism reduction for the oscillatory oxidation of hydrogen: sensitivity and quasi-steady-state analyses // *Combust. and Flame*. 1992. V. 91. N 2. P. 107–130.
3. Massias A., et al. An algorithm for the construction of global reduced mechanisms with CSP data // *Combust. and Flame*. 1999. V. 117. N 4. P. 685–708.
4. Lu T., Ju Y., Law P. K. Complex CSP for chemistry reduction and analysis // *Combust. and Flame*. 2001. V. 126. N 1–2. P. 1445–1455.
5. Peters N. Flame calculations with reduced mechanisms — an outline / *Reduced kinetic mechanisms for applications in combustion systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. P. 3–14.
6. Rabitz H., Kramer M., Dacol D. Sensitivity analysis in chemical kinetics // *Annual Rev. of Phys. Chem.* 1983. V. 34, N 1. P. 419–461.
7. Niemeyer K. E., Sung C.-J., Raju M. P. Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis // *Combust. and Flame*. 2010. V. 157, N 9. P. 1760–1770.

This study was conducted within the scientific program of the National Center for Physics and Mathematics, section N 2 “Mathematical modeling on Zetta-scale and Exa-scale Supercomputers. Stage 2023–2025”.

8. Mauersberger G. ISSA (iterative screening and structure analysis) a new reduction method and its application to the tropospheric cloud chemical mechanism RACM/CAPRAM2.4 // *Atmosph. Environ.* 2005. V. 39, iss. 2324. P. 4341–4350.
9. Zeuch T., Moreac G., Ahmed S., Mauss F. A comprehensive skeletal mechanism for the oxidation of n-heptane generated by chemistry-guided reduction // *Combust. and Flame.* 2008. V. 155. P. 651–674.
10. Pepiot-Desjardins P., Pitsch H. An automatic chemical lumping method for the reduction of large chemical kinetic mechanisms // *Combust. Theory and Modell.* 2008. V. 12, iss. 6. P. 1089–1108.
11. Lu T., Law C. K. A directed relation graph method for mechanism reduction // *Proc. of the Combust. Institute.* 2005. V. 30, iss. 1. P. 1333–1341.
12. Sun W., Chen Z., Gou X., Ju Y. A path ux analysis method for the reduction of detailed chemical kinetic mechanisms // *Combustion and Flame.* 2010. V. 157, N 7. P. 1298–1307.
13. Pepiot-Desjardins P., Pitsch H. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms // *Combust. and Flame.* 2008. V. 154, N 12. P. 6781.
14. Gao X., Yang S., Sun W. A global pathway selection algorithm for the reduction of detailed chemical kinetic mechanisms // *Combust. and Flame.* 2016. V. 167. P. 238–247. DOI: 10.1016/j.combustame.2016.02.007.
15. Dijkstra E. W. A note on two problems in connection with graphs // *Numer. Math.* 1959. V. 1. P. 269–271. <https://doi.org/10.1007/BF01386390>.
16. Yen J. Y. Finding the k shortest loopless paths in a network // *Manag. Sci.* 1971. V. 17. P. 712–716.
17. Gerb A. R., Deviatykh E. E., Omarova G. A. Metody grafovoi redukcii v modeliakh himicheskoi kinetiki // *Probl. inform.* 2024. N 3 (64).
18. Goodwin D. G., Moffat H. K., Speth R. L. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. [Electron. Res.]: <http://www.cantera.org>.
19. [Electron. Res.]: https://eigen.tuxfamily.org/index.php?title=Main_Page.
20. [Electron. Res.]: https://www.boost.org/doc/libs/1_83_0/doc/html/program_options.html.
21. [Electron. Res.]: <https://github.com/jbeder/yaml-cpp>.
22. The CRECK Modeling Group. Detailed kinetic mechanisms. [Electron. Res.]: <http://creckmodeling.chem.polimi.it/menu-kinetics/menu-kinetics-detailed-mechanisms/>.

СРАВНЕНИЕ МЕТОДОВ ГРАФОВОЙ РЕДУКЦИИ ПЕРВОГО, ВТОРОГО И ТРЕТЬЕГО ПОКОЛЕНИЯ В МОДЕЛЯХ ХИМИЧЕСКОЙ КИНЕТИКИ

А. Р. Герб, Е. Е. Девятых*, Г. А. Омарова

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

*Новосибирский Государственный Университет,
630090, Новосибирск, Россия

УДК 519.17+51-7

DOI: 10.24412/2073-0667-2025-4-25-37

EDN: VBFMRT

Исследованы и реализованы методы графовой редукции. Проведен сравнительный анализ работы рассматриваемых методов на пяти моделях с различными порогами ошибки. Исходный кинетический механизм представлен в виде ориентированного графа, узлы которого соответствуют химическим веществам, дуги отражают зависимости между веществами. Преимущество методов заключается в меньших вычислительных затратах и способности формировать достаточно компактные редуцированные модели.

Ключевые слова: граф, редукция, модель химкинетики, DRG, DRGEP, PFA, GPS.

Введение. Детальное описание механизмов протекания химических реакций окисления углеводородов включает множество всевозможных путей и наборов скоростей элементарных реакций для промежуточных продуктов реакций. Основными параметрами этих механизмов являются такие параметры как давление, температура, отношение эквивалентности и время пребывания. Модели больших размеров трудно использовать для расчета даже умеренно сложных явлений горения, таких как нестационарное, двумерное и трехмерное ламинарное пламя. Кроме того, их использование при моделировании явлений горения часто приводит к существенным различиям во временных масштабах системных переменных из-за сильно различающихся реакционных способностей химических соединений.

Результатирующая жесткость системы может значительно ограничить применение многих быстрых и простых численных методов, таким образом усложнив решение задачи. Моделирование таких систем осуществляется путем численного интегрирования указанных дифференциальных уравнений. Решение систем дифференциальных уравнений, описывающих динамику подобных механизмов, требует значительных вычислительных ресурсов и времени. Таким образом, существует необходимость разработки систематических методологий, которые могут уменьшить детализированные механизмы и их жесткость. Для

Исследование выполнено в рамках научной программы Национального центра физики и математики, направление № 2 «Математическое моделирование на супер-ЭВМ экса- и зеттафлопсной производительности. Этап 2023–2025».

решения этой проблемы применяются методы редукции химических кинетических механизмов, сокращающие количество веществ и реакций в модели. Правильно разработанные механизмы редукции позволяют сократить вычислительные и временные затраты, сохранив достаточную точность.

Редукция механизма может быть проведена на двух уровнях детализации. Первый уровень — скелетная редукция, которая устраняет неважные реакции и вещества либо на основе комплексного рассмотрения, либо для конкретного применения. Разработано несколько методов, каждый обладает как достоинствами, так и ограничениями. Например, два метода идентификации избыточных частиц в больших механизмах были разработаны с помощью анализа скорости реакции и матрицы Якоби соответственно [1, 2]. Метод анализа скорости реакции предполагает, что вещество является избыточным, если устранение всех потребляющих его реакций не вызывает существенной ошибки для остальных веществ. Этот метод прост в использовании, однако его реализация занимает много времени из-за проверки каждого исключенного вещества. Применение матрицы Якоби может идентифицировать вещества, относящиеся к важным. Однако для этого требуется итерационная процедура [2] с произвольным выбором пороговых значений [1]. Метод детальной редукции позволяет систематически выявлять второстепенные реакции, сравнивая скорость конкретной реакции со скоростью предварительно выбранной контролирующей реакции. Однако идентификация контролирующей реакции непростая, особенно для крупных механизмов, из-за отсутствия универсально строгого определения контролирующего поведения и возможной смены контролирующих процессов в различных условиях. Более того, медленная реакция необязательно является неважной.

Второй уровень — это редукция скелета CSP (computational singular perturbation), для ее выполнения вводится вычислительное сингулярное возмущение, которое может быть использовано для выявления и исключения только элементарных реакций, не важных ни для одного вещества. Для оценивания используется индекс важности [3, 4]. Редукция веществ с помощью CSP реализуется только при дальнейшем допущении состояний реакций и веществ, составляющих второй уровень редукции. Второй уровень редукции включает применение допущений о частичном равновесии и квазистационарном состоянии (quasi-steady-state) [5] к скелетному механизму с использованием методов анализа скорости реакции с критериями малых мольных долей, нормированных чистых дебитов, а также анализ чувствительности. Эти методы в основном требуют полных знаний, зависящих от механизма, и могут быть неэффективными для удаления мод короткого масштаба времени. Метод CSP полностью учитывает зависимость матрицы Якоби от времени и может точно идентифицировать быстрые моды, хотя процедура уточнения зависящего от времени якобиана требует большого времени вычислений. Следовательно, для практических целей при создании статических приведенных механизмов CSP часто используется в сочетании с предположением о постоянном якобиане.

Среди методов сокращения имеются и другие методы, такие как анализ чувствительности [6, 7], структурный анализ (ISSA) [8] и метод химического восстановления (CGR) [9]. Данные методы позволяют найти и удалить несущественные вещества или реакции путем оценки их влияния на ключевые характеристики модели. Однако данные методы являются вычислительно затратными, так как требуют многократного вычисления ошибок, поэтому зачастую их применяют для моделей, уже сокращенных с помощью других методов. Метод группировки химических веществ, обладающих схожими свойствами [10], менее эффективен для моделей, которым присуща химическая жесткость, обусловленная большой

разницей в скоростях реакций. В работе уделяется особое внимание методам редукции с использованием теории графов. Исходный кинетический механизм представляется в виде ориентированного графа, узлы которого соответствуют химическим веществам, дуги отражают зависимости между веществами (через совместное участие в реакциях). Например, DRG [11] и PFA [12] используют алгоритмы обхода графа для вычисления важных соединений, тогда как DRGEP [13] и GPS [14] для анализа связей применяют алгоритмы поиска кратчайших путей, такие как алгоритм Дейкстры [15] и алгоритм Йена поиска k кратчайших путей [16]. Преимущество таких методов заключается в меньших вычислительных затратах и их способности формировать достаточно компактные редуцированные модели, которые делают их эффективными для предварительного упрощения сложных химических кинетических механизмов.

Таким образом, разработка и реализация графовых методов редукции имеет важное значение для химической кинетики горения. Эти методы позволяют создавать компактные и точные модели, что особенно актуально при работе с большими механизмами, использующими сложные расчеты. Обзор программных продуктов графовой редукции химических кинетических моделей сделан в работе [17]. Учитывая ограничения существующих решений, целью данной работы является разработка программы на C++ для реализации основных методов графовой редукции химических кинетических моделей, включая DRG, DRGEP, PFA и GPS, с использованием библиотеки Cantera [18].

1. Алгоритм глобального выбора путей GPS. Методы DRG, DRGEP и PFA, применяемые для сокращения кинетических механизмов, подробно описаны в работе [17]. Алгоритм глобального выбора путей GPS является методом третьего поколения.

Несмотря на эффективность методов DRG, DRGEP, PFA, все они в той или иной мере фокусируются на сохранении веществ, важных для отдельных целевых компонентов или локальных показателей первого-второго поколения (скоростей реакций, потоков). Алгоритм глобального выбора путей был предложен Гао, Яном и Суном в 2016 г. [14] для более систематического выявления ключевых глобальных реакционных путей во всем механизме. Идея GPS заключается в анализе превращений химических элементов (атомов) через цепочки реакций, что позволяет выявлять важнейшие пути протекания вещества от исходных реагентов до конечных продуктов.

Алгоритм GPS можно разделить на четыре основных шага.

1) Построение графов потоков элементов.

Для каждого химического элемента (C, H, O, N и т. д.), присутствующего в системе, строится ориентированный граф, отражающий перенос атомов элемента между веществами в реакциях. Узлами графа являются химические соединения, содержащие данный элемент. Если элемент (например, углерод) переходит из соединения А в соединение В (А прореагировал и дал продукт В, сохранив атом углерода) с использованием одной или нескольких реакций, то проводится дуга от А к В. Вес этой дуги пропорционален потоку атомов данного элемента через соответствующие реакции и вычисляется следующим образом [14]:

$$r_{AB}^e = \sum_{i=0}^{N_r} r_{AB}^{e,i},$$

$$r_{AB}^{e,i} = \max(0, C_{AB}^{e,i} w_i),$$

$$C_{AB}^{e,i} = \frac{u_{A,i}^- n_e(A)}{\sum_{X \in \text{Reag}(i)} u_{X,i}^- n_e(X)} u_{B,i}^+ n_e(B),$$

где $r_{AB}^{e,i}$ — скорость перетекания элемента e от А к В в реакции i ; w_i — скорость i -й реакции (положительна, если скорость прямой реакции больше, чем скорость обратной, и наоборот); $C_{AB}^{e,i}$ — количество атомов элемента e , перемещаемых реакцией i из вещества А в вещество В; $u_{A,i}^-$ — стехиометрический коэффициент вещества А как реагента в реакции i ; $u_{B,i}^+$ — стехиометрический коэффициент вещества В как продукта реакции i ; $n_e(A)$ — число атомов элемента e в веществе А; $\text{Reag}(i)$ — множество реагентов реакции i .

Таким образом, дуги графа характеризуются числом молей элемента, переходящих от одного соединения к другому в единицу времени.

2) Поиск веществ-концентратов.

В каждом графе потока элемента ведется поиск соединений, через которые протекают наибольшие потоки элемента. Вещество А является концентратом, если для него выполняется условие:

$$\frac{\max \left(\sum_{A \neq B} r_{AB}^e, \sum_{A \neq B} r_{BA}^e \right)}{\max_M \left(\max \left(\sum_{M \neq B} r_{MB}^e, \sum_{M \neq B} r_{BM}^e \right) \right)} = \alpha_A^e > \alpha,$$

где α_A^e — относительный поток элемента e , протекающий через данное вещество А; α — заранее заданный порог.

Таким образом, отбираются соединения, у которых относительный входной или выходной поток превосходит порог α . Если вещество А является концентратом, то можно сделать вывод о том, что значительная часть атомов элемента исходной смеси, например в процессе горения, преобразуется в А.

3) Поиск глобальных путей по графам элементов.

Глобальным путем в заданном графе потока элемента называется путь от вещества в исходной смеси (например, CH_4) до конечного продукта (CO_2), проходящий через вещество-концентрат. Глобальный путь позволяет идентифицировать последовательные цепочки превращения соединений от начальных реагентов до конечных продуктов всего реакционного процесса. Учитывая основные глобальные пути, через которые проходят наибольшие потоки элемента, можно идентифицировать максимально важные цепочки превращения, которые должны быть сохранены в скелетном механизме.

Для каждого вещества-концентрата существует множество глобальных путей, поэтому для идентификации основных глобальных путей используется алгоритм нахождения k кратчайших независимых путей в ориентированном взвешенном графе. Определяется граф с обратно пропорциональными весами. Глобальный путь вычисляется как комбинация двух путей: путь от вещества из исходной смеси до вещества-концентрата и путь от вещества-концентрата до одного из продуктов. На выходе получаем множество глобальных путей мощностью k^2 . Все такие пути для каждого вещества-концентрата и каждой пары заданных соединений в каждом графе потока элемента сохраняются и образуют множество веществ, входящих в итоговый скелетный механизм.

4) Сохранение второстепенных соединений, важных для связывания соединений в глобальных путях.

Глобальный путь представляет собой комбинацию этапов преобразования, каждый из них описывается группой реакций, которые могут включать второстепенные соединения, изначально не входящие в множество сохраняемых веществ. Некоторые реакции из такой группы могут иметь большее значение в преобразовании, а некоторые — меньшее. Поэтому для сохранения целостности этапов преобразования важного глобального пути необходимо также сохранить второстепенные соединения, участвующие в основных реакциях этапов преобразования пути. Определение потоков, вносящих наибольший вклад в этап преобразования вещества А в В, происходит следующим образом: берутся S реакций, в которых элемент e перетекает быстрее всего от А к В, и все потоки, у которых эти реакции составляют долю потока меньше порога β , отбрасываются.

Метод GPS ориентирован на сохранение целостности ключевых глобальных путей превращения, особенно путей протекания каждого химического элемента от начала до конца химического процесса. В отличие от DRG, DRGEP, PFA, которые могут фокусироваться на одном целевом веществе (например, топливе) и рискуют упустить какой-либо элемент, GPS поэлементно гарантирует, что, например, весь углеродный поток из топлива до CO_2 в скелетном механизме сохранен через непрерывную цепочку реакций. Такой подход минимизирует вероятность ситуаций, когда соединение присутствует, но дальше по цепочке элемент никуда не перетекает или, наоборот, появляется из ниоткуда. В работе [14] показано, что GPS позволяет получать весьма компактные механизмы при хорошем сохранении точности по широкому кругу условий, а также служит инструментом для анализа кинетики: выявленные пути несут ценную информацию о протекании реакции.

К недостаткам алгоритма GPS можно отнести большие вычислительные затраты. Сложность построения матрицы смежности для графа аналогична сложности метода DRG, однако задача поиска путей в графе элементов является нетривиальной, и в первоначальной реализации GPS применяется алгоритм Йена для поиска k кратчайших путей [16], сложность которого равна $O(kn(m + n \log n))$.

2. Постановка задачи. Для заданной химической кинетической модели реализовать графовую редукцию на основе методов DRG, DRGEP, PFA и GPS.

Входные данные: 1) YAML-файл, описывающий детальную химическую кинетическую модель горения, в формате, поддерживаемом библиотекой Cantera. 2) Наборы начальных условий для моделирования. 3) Параметры редукции.

Файл, описывающий химическую кинетическую модель, содержит следующую информацию: 1) Описание одной фазы, т.е. физической среды, в которой происходят химические и термодинамические процессы. Это может быть среда идеального газа, идеального раствора или поверхности катализатора как идеальной поверхности, контактирующей с другой фазой. В задаче дана только одна фаза — фаза идеального газа, она включает информацию о химических элементах, веществах и реакциях, задействованных в ней. 2) Информация о химических элементах фазы, представленная списком названий химических элементов, составляющих соединения. 3) Состав веществ, присутствующих в фазе, представленный списком объектов, описывающих химические соединения. Объект включает: псевдоним (химическая формула), элементный состав, термодинамические свойства вещества (энтропия, теплоемкость) и транспортную информацию (коэффициенты диффузии, параметры вязкости). 4) Набор происходящих в фазе реакций, также представленный списком объектов. Объект описывает: уравнение реакции, ее тип (например, элементарная реакция, реакция трех тел) и некоторые важные параметры, специфичные для каждого типа реакции.

Информация о начальных условиях моделирования: 1) Тип симуляции: симуляция идеального газа с постоянным объемом или давлением. 2) Начальное состояние: химический состав, давление и температура.

Информация о параметрах редуцирования: 1) Максимальная ошибка в процентах. 2) Метод редукции. 3) Информация, специфичная для каждого метода. 3.1) Для методов DRG, DRGEP и PFA указываются целевые вещества. 3.2) Для GPS указываются начальные реагенты, конечные продукты, количество кратчайших путей, используемых для поиска глобальных, параметры α и β .

Ошибка редуцированной модели рассчитывается как процент отклонения времени с начала симуляции до воспламенения между полным механизмом и редуцированным.

Редукция проводится в автоматическом режиме с заданным порогом и выбранным методом.

Выходные данные включают YAML файл с редуцированной моделью.

3. Численные результаты. Программа разработана на языке программирования C++ на базе linux (Ubuntu версии 24.04).

Библиотеки, используемые программой: 1) Cantera [18] версии 3.1.0. Используется для проведения численного моделирования, чтения/записи и анализа YAML файлов химических кинетических механизмов. 2) Eigen [19] версии 3.4.0. Используется для реализации матриц смежности в форматах CSR и CSC. 3) Boost с модулем Program_options [20] версии 1.83.0. Используется для чтения и анализа аргументов командной строки. 4) yaml-cpp [21] версии 0.8.0. Используется для чтения и анализа файла постановки задачи редуцирования.

Проведено сравнительное исследование разработанных методов DRG, DRGEP, PFA и GPS. Исследование проводилось на моделях:

CRECK_2003_TOT_HT_LT_ME — 582 вещества и 21174 реакции;

CRECK_2003_C1_C3_HT — 114 веществ и 1999 реакций;

CRECK_2003_TOT_HT — 368 веществ и 14462 реакции;

CRECK_2003_TPRF_HT_ALC_NOX — 299 веществ и 8028 реакций;

GRI-Mech Version 3.0 — 53 веществ и 325 реакции.

Первые 4 модели взяты с официального сайта CRECK Modeling Group [22], GRI-Mech поставляется вместе с библиотекой Cantera. Использован один набор начальных условий для численного моделирования, для всех запусков программы использовались одни и те же настройки, отличающиеся только заранее заданной ошибкой и методом.

Настройки включают: целевые вещества: O_2 , C_2H_6 , C_3H_8 ; вещества, не подлежащие удалению: N_2 ; условия для моделирования процесса воспламенения (давление 1 атм.; температура 1200 K; молярный состав: 1 моль C_3H_8 , 6 молей O_2 , 22.56 моли N_2).

В табл. 1–5 представлены результаты запусков программы. ME (Max error) — максимальная ошибка; SA (Species after) — количество веществ уменьшенной модели; RA (Reactions after) — количество реакций уменьшенной модели; E (Error) — ошибка для уменьшенной модели; T (Time) — время работы программы.

3.1. Анализ результатов. Наиболее эффективными с точки зрения размера итоговой модели оказались два метода: DRGEP и GPS. Результат связан с особенностями алгоритма. После анализа путей от целевых веществ и поиска максимального пути, удаляются вещества, значение «длины пути» для которых не превышает заданного порогового значения. Таким образом, указанные методы напрямую удаляют вещества, в то время как другие методы тем или иным образом делят граф на классы связности и оставляют те, в которых есть целевые вещества. Можно отметить, что метод третьего поколения работает

Таблица 1

Результаты модели CRECK_2003_TOT_HT_LT_ME

ME	DRG				DRGEP				PFA				GPS			
	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time
0.10	263	8766	0.04	111.95	118	2439	0.06	96.70	281	9083	0.10	163.59	129	2942	0.08	131.54
0.20	263	8766	0.04	112.15	105	2194	0.16	98.83	281	9083	0.10	162.17	110	2511	0.11	136.47
0.30	214	5636	0.25	117.22	96	1918	0.26	101.67	281	9083	0.10	159.52	99	2149	0.29	138.12
0.40	214	5636	0.25	113.43	96	1918	0.26	100.41	240	6699	0.35	181.83	99	2149	0.29	137.37
0.50	214	5636	0.25	112.85	95	1854	0.47	98.07	237	6602	0.50	186.48	99	2149	0.29	138.32
0.60	214	5636	0.25	118.13	95	1854	0.47	103.57	209	5704	0.57	101.05	99	2149	0.29	134.07
0.70	214	5636	0.25	116.33	95	1854	0.47	101.18	81	1276	0.65	101.84	59	956	0.17	127.13
0.80	214	5636	0.25	114.66	89	1726	0.74	101.68	81	1276	0.65	102.93	59	956	0.17	126.08
0.90	214	5636	0.25	120.80	89	1726	0.74	104.51	72	1140	0.61	104.15	59	956	0.17	123.50
1.00	214	5636	0.25	113.86	87	1713	0.97	106.46	72	1140	0.61	104.13	59	956	0.17	126.54
5.00	72	1228	0.03	119.22	56	803	2.20	92.61	63	1037	4.71	105.29	54	838	3.39	124.80

Таблица 2

Результаты модели CRECK_2003_C1_C3_HT

ME	DRG				DRGEP				PFA				GPS			
	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time
0.10	95	1757	0.05	15.41	92	1572	0.02	9.07	95	1753	0.08	7.78	97	1812	0.01	20.45
0.20	95	1757	0.05	14.41	49	674	0.14	4.32	94	1750	0.18	11.07	95	1805	0.10	9.43
0.30	87	1650	0.22	7.12	49	674	0.14	3.80	94	1750	0.18	9.28	95	1805	0.10	12.38
0.40	87	1650	0.22	6.75	49	674	0.14	3.76	90	1666	0.36	11.97	95	1805	0.10	9.52
0.50	87	1650	0.22	7.08	49	674	0.14	3.87	90	1666	0.36	14.25	86	1572	0.47	10.41
0.60	87	1650	0.22	6.90	49	674	0.14	3.76	85	1628	0.59	12.84	86	1572	0.47	12.57
0.70	87	1650	0.22	7.15	49	674	0.14	3.78	82	1593	0.66	6.40	77	1362	0.52	12.49
0.80	87	1650	0.22	9.99	49	674	0.14	3.83	82	1593	0.66	9.46	69	1228	0.74	18.88
0.90	87	1650	0.22	7.17	49	674	0.14	6.75	82	1593	0.66	7.13	69	1228	0.74	16.50
1.00	87	1650	0.22	6.71	49	674	0.14	3.92	64	1074	0.91	6.75	69	1228	0.74	17.44
5.00	45	771	1.33	4.00	49	674	0.14	3.79	51	847	4.79	7.89	45	678	2.43	5.35

чуть лучше, это видно по результатам 1-й и 4-й модели, где-то хуже модель 2, результаты 3-й модели почти такие же. Полученная ошибка тоже меньше, однако время работы хуже, чем у методов первого и второго поколений. Это связано с алгоритмом поиска k путей для проверки важности сохранения веществ.

Из приведенных таблиц видно, что ошибка редуцированной модели не всегда возрастает с уменьшением количества оставленных веществ, что указывает на сложную взаимосвязь между числом включенных веществ и точностью воспроизведения исходного механизма. Так как задача сводится к нахождению минимального набора веществ, при котором ошибка моделирования не превышает заранее заданного допустимого уровня, ключевым параметром этой процедуры становится порог отсечения дуг в графе. Однако подбор оптимального значения этого порога представляет собой отдельную сложную задачу, которая требует дополнительных исследований и выходит за рамки рассматриваемой в данной работе тематики.

Таблица 3

Результаты модели CRECK_2003_TOT_HT

ME	DRG				DRGEP				PFA				GPS			
	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time
0.10	233	7649	0.01	101.04	117	2437	0.08	44.53	247	7901	0.03	73.72	150	3678	0.09	94.59
0.20	218	6934	0.12	50.96	105	2194	0.18	46.42	247	7901	0.03	71.47	126	2933	0.12	67.13
0.30	218	6934	0.12	50.01	96	1918	0.28	51.83	233	7368	0.26	86.16	97	2145	0.12	71.85
0.40	178	4540	0.31	53.27	96	1918	0.28	46.96	233	7368	0.26	81.66	97	2145	0.12	73.92
0.50	178	4540	0.31	49.30	95	1854	0.50	51.41	181	4826	0.48	106.31	97	2145	0.12	71.25
0.60	178	4540	0.31	52.67	95	1854	0.50	47.89	75	1194	0.59	45.03	97	2145	0.12	71.22
0.70	178	4540	0.31	50.23	95	1854	0.50	54.92	75	1194	0.59	47.16	97	2145	0.12	71.92
0.80	178	4540	0.31	52.33	89	1726	0.76	51.02	75	1194	0.59	44.60	59	956	0.15	60.43
0.90	178	4540	0.31	50.17	89	1726	0.76	53.57	65	1047	0.57	47.75	59	956	0.15	57.94
1.00	178	4540	0.31	52.91	87	1713	1.00	54.71	65	1047	0.57	45.70	59	956	0.15	60.37
5.00	68	1214	0.09	50.76	56	803	2.18	40.44	63	1037	4.73	45.17	54	838	3.41	61.60

Таблица 4

Результаты модели CRECK_2003_TPRF_HT_ALC_NOX

ME	DRG				DRGEP				PFA				GPS			
	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time
0.10	219	5455	0.05	66.11	117	2437	0.08	28.57	222	5401	0.03	45.30	149	3638	0.09	73.37
0.20	212	5303	0.12	36.12	105	2194	0.18	30.57	222	5401	0.03	44.02	80	1464	0.13	36.50
0.30	212	5303	0.12	33.14	97	1921	0.30	30.75	213	5170	0.26	54.81	80	1464	0.13	37.20
0.40	171	4064	0.32	34.26	97	1921	0.30	31.99	213	5170	0.26	54.44	80	1464	0.13	56.32
0.50	171	4064	0.32	34.22	95	1854	0.50	32.71	192	4203	0.47	72.11	59	956	0.14	45.60
0.60	171	4064	0.32	34.68	95	1854	0.50	32.36	101	1482	0.58	28.64	59	956	0.14	42.73
0.70	171	4064	0.32	34.60	95	1854	0.50	32.70	101	1482	0.58	26.44	59	956	0.14	45.25
0.80	171	4064	0.32	32.56	89	1726	0.77	33.04	101	1482	0.58	29.27	59	956	0.14	43.93
0.90	171	4064	0.32	34.29	89	1726	0.77	33.83	66	1053	0.57	30.58	59	956	0.14	46.59
1.00	171	4064	0.32	34.17	87	1713	1.00	36.14	66	1053	0.57	30.85	59	956	0.14	47.13
5.00	69	1220	0.09	35.52	56	803	2.17	19.74	64	1043	4.74	30.92	54	838	3.41	44.55
10.00	48	799	5.86	20.12	56	803	2.17	21.59	54	866	9.24	32.15	52	803	8.38	41.95

Алгоритм поиска и итерации методом второго поколения PFA аналогичен применяемому в методе DRG, различие заключается только в способе расчета коэффициентов прямого взаимодействия. В отличие от DRG, алгоритм PFA учитывает пути второго поколения между соединениями при расчете их коэффициентов прямого взаимодействия. Значения рассчитываются с использованием потоков производства и потребления для каждого соединения. Результаты по времени у PFA выше, так как увеличиваются затраты на вычисление коэффициентов взаимодействия первого и второго поколений. Полученные модели меньше для некоторых моделей, чем у DRG, но больше DRGEP и GPS.

Модель GRI-Mech, модель небольшая и была проредуцирована всеми методами. По результатам редукции видны слабые и сильные места методов. Например, метод GPS при максимальной ошибке от 0.1 до 0.3 дает наименьшую модель, а от 0.4 лучший результат

Таблица 5

Результаты модели gri30

ME	DRG				DRGEP				PFA				GPS			
	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time	SA	RA	Err	Time
0.10	48	302	0.00	1.37	34	205	0.00	0.93	45	275	0.00	3.69	30	179	0.04	1.17
0.20	48	302	0.00	1.38	34	205	0.00	0.92	45	275	0.00	1.56	30	179	0.04	1.26
0.30	39	229	0.29	1.91	34	205	0.00	0.99	36	205	0.29	1.19	30	179	0.04	1.34
0.40	39	229	0.29	4.26	28	155	0.11	1.29	36	205	0.29	1.25	30	179	0.04	1.29
0.50	39	229	0.29	2.07	26	128	0.39	1.53	36	205	0.29	1.11	30	179	0.04	1.17
0.60	39	229	0.29	2.03	26	128	0.39	1.83	36	205	0.29	1.16	30	179	0.04	1.29
0.70	39	229	0.29	1.94	26	128	0.39	1.52	36	205	0.29	1.03	28	160	0.58	1.51
0.80	39	229	0.29	1.89	26	128	0.39	1.56	36	205	0.29	1.01	28	160	0.58	1.64
0.90	25	115	0.81	0.85	26	128	0.39	1.48	36	205	0.29	1.03	28	160	0.58	1.51
1.00	25	115	0.81	0.88	26	128	0.39	1.48	36	205	0.29	1.07	28	160	0.58	1.49
5.00	25	115	0.81	0.91	26	128	0.39	1.44	36	205	0.29	1.05	28	160	0.58	1.68
10.00	25	115	0.81	0.89	24	109	5.34	0.76	36	205	0.29	1.05	23	108	6.79	4.31

обеспечивает DRGEP. Если нужно выбрать метод, для которого критерием будет выходная ошибка, то это PFA.

Заключение. В данной работе для реализации методов DRG, DRGEP и PFA использовались основные идеи, описанные в работе [17]. В ходе работы авторы постарались устранить недостатки программного комплекса ruMARS.

1) Формирование графа для последующей редукции в формате CSR и CSS. Данные форматы хранения матрицы отлично подходят для хранения разреженных матриц, оптимизируя потребление памяти.

2) Реализация алгоритмов редукций C/C++ отражается на скорости работы алгоритмов.

3) Поддержка YAML формата хранения моделей химической кинетики Cantera.

4) Оптимизированы графовые алгоритмы — поиск в глубину и алгоритм Дейкстры, — используемые в методах DRG, PFA и DRGEP.

Разработан и реализован оптимизированный алгоритм поиска k кратчайших путей для реализации метода глобальных путей GPS.

Проведено тестирование программы, получены численные результаты для 5 моделей с 11 порогами максимальной ошибки.

Разработана программа для автоматизированной редукции с заданной точностью химических кинетических механизмов, принимающая YAML файлы механизмов в формате, совместимом с библиотекой Cantera, и обладающая широким спектром функций и возможностей.

Список литературы

1. Turanyi V. Reduction of large reaction mechanisms // New J. Chemistry. 1990. V. 14. № 11. P. 795–803.
2. Tomlin A. S., Pilling M. J., Turanyi T., Merkin J. H., Brindley J. Mechanism reduction for the oscillatory oxidation of hydrogen: sensitivity and quasi-steady-state analyses // Combust. and Flame. 1992. V. 91. № 2. P. 107–130.

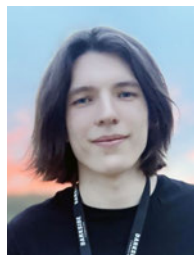
3. Massias A., et al. An algorithm for the construction of global reduced mechanisms with CSP data // *Combust. and Flame*. 1999. V. 117. № 4. P. 685–708.
4. Lu T., Ju Y., Law P. K. Complex CSP for chemistry reduction and analysis // *Combust. and Flame*. 2001. V. 126. № 1–2. P. 1445–1455.
5. Peters N. Flame calculations with reduced mechanisms — an outline / *Reduced kinetic mechanisms for applications in combustion systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. P. 3–14.
6. Rabitz H., Kramer M., Dacol D. Sensitivity analysis in chemical kinetics // *Annual Rev. of Phys. Chem.* 1983. V. 34, № 1. P. 419–461.
7. Niemeyer K. E., Sung C.-J., Raju M. P. Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis // *Combust. and Flame*. 2010. V. 157, № 9. P. 1760–1770.
8. Mauersberger G. ISSA (iterative screening and structure analysis) a new reduction method and its application to the tropospheric cloud chemical mechanism RACM/CAPRAM2.4 // *Atmosph. Environ.* 2005. V. 39, iss. 2324. P. 4341–4350.
9. Zeuch T., Moreac G., Ahmed S., Mauss F. A comprehensive skeletal mechanism for the oxidation of n-heptane generated by chemistry-guided reduction // *Combust. and Flame*. 2008. V. 155. P. 651–674.
10. Pepiot-Desjardins P., Pitsch H. An automatic chemical lumping method for the reduction of large chemical kinetic mechanisms // *Combust. Theory and Modell.* 2008. V. 12, iss. 6. P. 1089–1108.
11. Lu T., Law C. K. A directed relation graph method for mechanism reduction // *Proc. of the Combust. Institute*. 2005. V. 30, iss. 1. P. 1333–1341.
12. Sun W., Chen Z., Gou X., Ju Y. A path ux analysis method for the reduction of detailed chemical kinetic mechanisms // *Combustion and Flame*. 2010. V. 157, № 7. P. 1298–1307.
13. Pepiot-Desjardins P., Pitsch H. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms // *Combust. and Flame*. 2008. V. 154, № 12. P. 6781.
14. Gao X., Yang S., Sun W. A global pathway selection algorithm for the reduction of detailed chemical kinetic mechanisms // *Combust. and Flame*. 2016. V. 167. P. 238–247. DOI: 10.1016/j.combustame.2016.02.007.
15. Dijkstra E. W. A note on two problems in connection with graphs // *Numer. Math.* 1959. V. 1. P. 269–271. <https://doi.org/10.1007/BF01386390>.
16. Yen J. Y. Finding the k shortest loopless paths in a network // *Manag. Sci.* 1971. V. 17. P. 712–716.
17. Герб А. Р., Девярых Е. Е., Омарова Г. А. Методы графовой редукции в моделях химической кинетики // *Пробл. информ.* 2024. № 3 (64).
18. Goodwin D. G., Moffat H. K., Speth R. L. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. [Electron. Res.]: <http://www.cantera.org>.
19. [Electron. Res.]: https://eigen.tuxfamily.org/index.php?title=Main_Page.
20. [Electron. Res.]: https://www.boost.org/doc/libs/1_83_0/doc/html/program_options.html.
21. [Electron. Res.]: <https://github.com/jbeder/yaml-cpp>.
22. The CRECK Modeling Group. Detailed kinetic mechanisms. [Electron. Res.]: <http://creckmodeling.chem.polimi.it/menu-kinetics/menu-kinetics-detailed-mechanisms/>.



Герб Артем Родионович — аспирант Института вычислительной математики и математической геофизики. В 2023 году окончил механико-математический факультет НГУ, степень магистра. E-mail: gerb-artem@mail.ru.

Gerb Artem is a postgraduate student of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. Graduated from Novosibirsk State University's Math Faculty in 2023.

Девятых Егор Евгеньевич — студент 1 курса магистратуры механико-математического факультета Новосибирского государственного университета. E-mail: klasifate@gmail.com.



Devyatykh Egor is a 1st year master's student of the Faculty of Mechanics and Mathematics of Novosibirsk State University.

Омарова Гульзира Алимовна — канд. физ.-мат. наук, науч. сотр. Института вычислительной математики и математической геофизики СО РАН; e-mail: gulzira@rav.sccc.ru.



Omarova Gulzira is PhD in Physics and Mathematics, Research Fellow at the Institute of Computational Mathematics and Mathematical Geophysics SB RAS.

Дата поступления — 01.10.2025

GENERATION OF MULTI-LEVEL REGULAR NETWORKS BASED ON THE COMPOSITION OPERATION OF MODIFIED CHORDAL GRAPHS USING LARGE LANGUAGE MODELS

O. G. Monakhov, E. A. Monakhova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-38-51

EDN: VRVDCR

The paper is devoted to the development and study of a new class of topologies for communication networks used in multiprocessor systems and networks-on-chip. The primary goal of this paper is to solve the problem of network topology optimization. Key network characteristics are its diameter (the maximum shortest distance between any two nodes) and the average distance between nodes: the smaller these parameters, the lower the data transmission latency and the higher the overall system performance. The goal of this paper is to propose a new network construction model that achieves better performance (smaller average distance) compared to existing optimal circulant networks with the same hardware costs (i. e., the same number of nodes and links).

The paper proposes a new method for constructing multi-level networks and introduces a new operation, multi-level composition, which allows the use of a wide range of regular graphs, previously proposed as computing system structures, as layer elements, combining them optimally. In this paper, the operation of multi-level composition is applied to a class of chordal rings and modified chordal graphs. The essence of the method is as follows: 1. Level creation: The network is constructed from several (m) levels. Each level is a regular graph (in this paper, a chordal graph g_1). In the first step, m disconnected copies of this graph are created. 2. Level connection: All nodes from all levels are connected to each other via a second, “global” graph (G_2), which is also chordal. The result is a new, more complex regular network G , which is the sum of the graphs G_1 (intra-level connections) and G_2 (inter-level connections).

To find the best configuration of such a network (i. e., to find the generators for graphs g_1 and G_2), the authors use the Simulated Annealing algorithm. This algorithm allows for efficient finding of parameters that minimize the average distance in the resulting network. The algorithm for synthesizing optimal multi-level networks was developed using large language models and implemented in sequential and parallel versions on the Kumpeng cluster. The authors conducted numerical simulations and compared their multi-layer networks with the best known circulant networks (C). Optimal (suboptimal) multi-layer regular networks of degrees 4 and 6 were obtained. The influence of multi-layer network parameters on the average distance including the number of levels was studied. The paper presents the key experimental results. (1) Improvement: in some configurations, for degree 4, the improvement reached more than 3 times compared to optimal circulant networks. (2) Scalability: the advantage of the new approach becomes more pronounced with an increasing number of nodes in the network. This makes it promising for building large computing systems and networks. (3) Use of AI in development: an interesting feature of the work is that the authors used large language models (LLM), such as

Supported by the state assignment of ICMMG SB RAS N FWNM-2025-0005.

Gemini-2.5Pro, to assist in writing and parallelizing C code for running simulations. According to the authors' estimates, this reduced development time by 20–30 %.

The article demonstrates the effectiveness of the new approach and convincingly demonstrates that the proposed multi-level composition operation is a new tool for constructing high-performance network topologies. The practical significance of the article lies in the fact that the resulting networks can be used to design real multiprocessor systems and networks-on-chip, providing lower latency and improved performance. The proposed method is general and can be applied in the future to other classes of graphs, not just chordal ones, opening up new avenues for research. The paper utilizes modern artificial intelligence tools and demonstrates the successful integration of modern AI assistants into scientific research and the development of complex software. Future work is planned to explore the use of other classes of graphs to obtain multi-level regular networks, as well as to theoretically study their characteristics and the possibility of constructing efficient routing algorithms in such networks.

Key words: chordal network, average distance, parametric description, circulant network, optimal graph, large language model.

References

1. Multilevel Network Analysis for the Social Sciences: Theory, Methods and Applications. Lazega, E., Snijders, T. (eds.). // Cham, Heidelberg: Springer. 2016.
2. Kivela M., Arenas A., Barthélemy M., Gleeson J. P., Moreno Y., Porter M. Multilayer Networks // Journal of Complex Networks. 2014. N 2 (3).
3. Kalney A. M. Modeli mnogourovnevnyh setej (kratkij obzor) // Problemy informatiki. 2021. N 3. P. 5–20.
4. Kalney A. M., Rodionov A. S. Analiz nadezhnosti mnogourovnevnyh setej s nenadezhnymi vershinami // Problemy informatiki. 2020. N 2. P. 5–15.
5. Hwang F. K. A survey on multi-loop networks // Theoret. Computer Science. 2003. V. 299.
6. Reyes, M. A., Dalfo, C., Fiol, M. A. Structural and Spectral Properties of Chordal Ring, Multi-Ring, and Mixed Graphs // Symmetry, 2024. 16. 1135.
7. Gutierrez J., Riaz T., Pedersen J., Labeaga S., Madsen O. Degree 3 networks topological routing // Image Processing and Communication. 2009. N 14.
8. Ledzinski, D., Smigiel, S., Zabłudowski, L. Analyzing methods of network topologies based on chordal rings // Turkish Journal of Electrical Engineering and Computer Sciences. 2018. V. 26: N 3, Article 25.
9. Monakhova, E. A., Monakhov, O. G. Metod avtomaticheskogo poiska semejstv optimal'nyh hordal'nyh kol'cevnyh setej // Diskretnyj analiz i issledovanie operacij. 2024. N 1. P. 85–108.
10. Monakhova, E. A., Monakhov, O. G., Otkrytie analiticheskikh zavisimostej parametrov optimal'nyh hordal'nyh setej na osnove analiza dannyh // Problemy informatiki. 2023. N 4.
11. Ahmad M., Zahid Z., Zavaid M., and Bonyah E. Studies of Chordal Ring Networks via Double Metric Dimensions // Math. Problems in Engineering. 2022. (ArticleID 8303242).
12. Arden B. W. and Lee H. Analysis of Chordal Ring Network // IEEE Trans. on Computers. 1981. N C-30.
13. Morillo P., Comellas F., Fiol M. A. The optimization of Chordal Ring Networks // Communication Technology, Eds. Q. Yasheng and W. Xiuying. World Scientific, 1987. P. 295–299.

14. Huang, X., Ramos, A. F., Deng, Y. Optimal circulant graphs as low-latency network topologies // J. of Supercomputing. 2022. N 78. P. 13491–13510.
15. Deng Y., Guo M., Ramos A. F., Huang X., Xu Z., Liu W. Optimal low-latency network topologies for cluster performance enhancement // J. Supercomput, 2020. N 76 (12). P. 9558–9584.
16. Monakhova E. A Survey on Undirected Circulant Graphs // Discrete Mathematics, Algorithms and Applications. 2012. N 4 (1). 1250002.
17. Monakhov O., Monakhova E. A Class of Parametric Regular Networks for Multicomputer Architectures // Computacion y Sistemas. 2000. N 4. P. 85–93.
18. Karpenko A. P. Sovremennye algoritmy poiskovoj optimizacii. Algoritmy, vdohnovlennye prirodoj // Moskva: MGTU im. N. E. Baumana, 2017.

ГЕНЕРАЦИЯ МНОГОУРОВНЕВЫХ РЕГУЛЯРНЫХ СЕТЕЙ НА ОСНОВЕ ОПЕРАЦИИ КОМПОЗИЦИИ МОДИФИЦИРОВАННЫХ ХОРДАЛЬНЫХ ГРАФОВ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

О. Г. Монахов, Э. А. Монахова

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.8 + 519.7

DOI: 10.24412/2073-0667-2025-4-38-51

EDN: VRVDCR

Предложена новая модель топологий сетей связи для многопроцессорных систем и сетей на кристалле — класс многоуровневых регулярных параметрически задаваемых сетей (графов). В качестве элементов уровней можно использовать известные регулярные графы, предложенные ранее в качестве структур вычислительных систем, объединяя их оптимальным образом. В данной работе в качестве элементов построения при генерации многоуровневых сетей рассмотрены хордальные графы, которые объединялись с помощью предложенной операции многоуровневой композиции. При синтезе многоуровневых сетей применен алгоритм моделирования отжига для определения оптимальных параметров генерируемой топологии, минимизирующих среднее расстояние сети при заданном числе узлов, числе уровней и степени узлов. Алгоритм синтеза оптимальных сетей разработан с помощью больших языковых моделей и реализован в последовательной и параллельной версиях на кластере Kunpeng 920. Построенные многоуровневые сети имеют лучшие структурные характеристики, чем циркулянтные сети при одинаковых затратах оборудования (количестве узлов и линий связи).

Ключевые слова: хордальная сеть, среднее расстояние, параметрическое описание, циркулянтная сеть, оптимальный граф, большая языковая модель.

Введение. Теория многоуровневых сетей является важным инструментом описания и анализа сложных систем в социальных, биологических, физических, информационных и инженерных науках [1–4]. В данной работе многоуровневые оптимальные сети предлагаются в качестве топологий универсальных коммуникационных сетей для многопроцессорных систем и сетей на кристалле для широкого класса прикладных задач. В качестве базовой структуры построения многоуровневых регулярных сетей исследуется класс хордальных сетей степени три, рассмотренных в работах [5–11] и других.

Хордальные кольцевые графы (chordal ring graph) введены в [12] как модели сетей связи компьютерных систем. *Хордальный кольцевой* граф $H_N(1, -1, c)$ или $H(N; \pm 1, c)$ имеет N вершин, помеченных $0, 1, \dots, N-1$, где N — четное число, и ребра двух видов — $(i, i \pm 1 \bmod N)$, $i = 0, 1, \dots, N-1$, и $(i, i + c \bmod N)$ для нечетных i и c , $1 < c \leq N/2$.

Работа выполнена при финансовой поддержке бюджетным проектом ИВМиМГ СО РАН (код проекта FWNM-2025-0005).

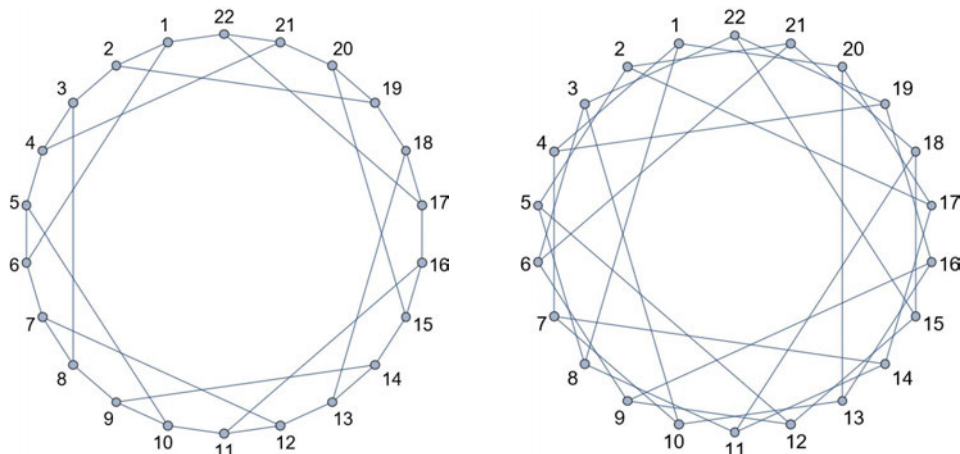


Рис. 1. Хордальный кольцевой граф $H(22; \pm 1, 5)$ (слева),
модифицированный хордальный граф $H(22; \pm 3, 7)$ (справа)

Число c — длина хорды (образующая) графа, N — порядок графа, степень вершин равна трем. На рис. 1, слева, изображен граф $H(22; \pm 1, 5)$. В [13] введено понятие обобщенных хордальных графов (generalized chordal ring graph), а именно, графов степени три вида $H(N; a, b, c)$, где порядок N — четное число, образующие a, b, c — нечетные числа $\leq N/2$.

При построении многоуровневых хордальных графов в данной работе в качестве базовых элементов построения используются как хордальные кольцевые графы $H(N; \pm 1, c)$, так и модифицированные хордальные графы степени три вида $H(N; \pm a, c)$, где N — четное число, образующая c — нечетное число $\leq N/2$, а a — любое число $\leq N/2$. Пример модифицированного хордального графа $H(22; \pm 3, 7)$ с образующими, равными 3 и 7, изображен на рис. 1, справа. В статье использована нумерация вершин графов, начиная с 1.

Топологическими характеристиками графов, изучаемыми при проектировании топологий многопроцессорных систем, являются диаметр D (длина максимального кратчайшего пути на множестве всевозможных пар вершин графа) и среднее расстояние D_{av} — математическое ожидание расстояния в графе при равновероятном выборе пар вершин. Как показывают исследования, начиная с классических работ Клейнрока и ряда последующих работ [14], [15], именно среднее расстояние сети оказывает сильное влияние на производительность кластера при решении параллельных задач. При построении многоуровневых хордальных сетей назовем *оптимальным* регулярный граф, имеющий минимально возможное среднее расстояние для заданных N и степени.

Цель данной работы — (1) применить новую операцию многоуровневой композиции для построения многоуровневых сетей на основе хордальных графов, (2) получить оптимальные многоуровневые хордальные сети для заданных числа вершин N и степени v с минимальным средним расстоянием, меньше среднего расстояния оптимальных циркулянтных сетей тех же размеров. Циркулянтные сети [5, 16] — известный класс регулярных параметрически задаваемых графов, используемых при построении и анализе топологий компьютерных сетей и вычислительных систем, сетей на кристалле, в теории кодирования и других приложениях. По определению, циркулянтная сеть $C(N; S)$ порядка N и размерности k , имеет N вершин $0, 1, \dots, N-1$ и длины хорд (образующие) $S = \{\pm s_1, \pm s_2, \dots, \pm s_k\}$, где $1 \leq s_1 < \dots < s_k < N/2$. Будем сравнивать многоуровневые хордальные графы по

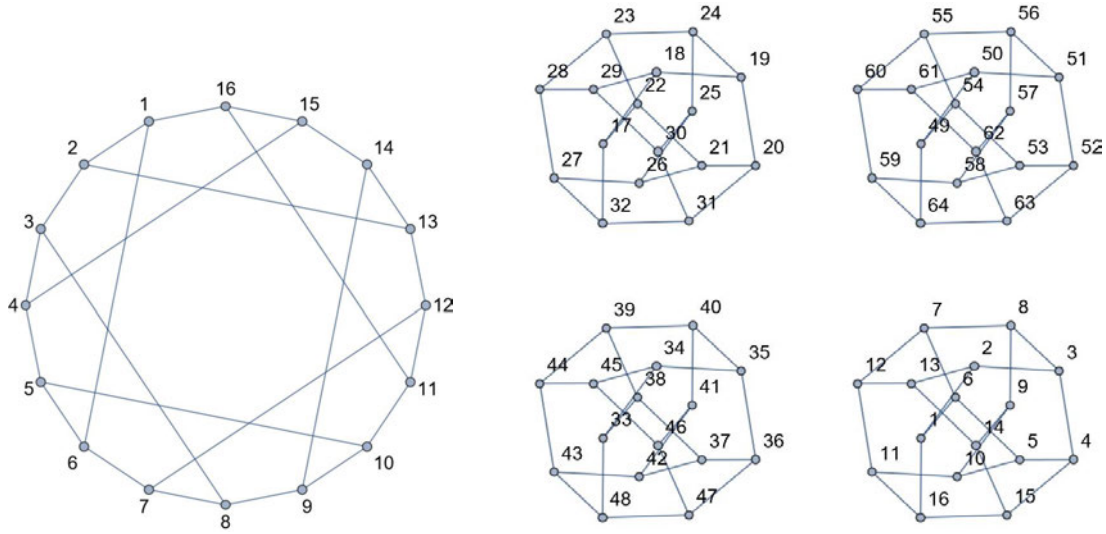


Рис. 2. Граф g_1 степени 3 с числом вершин 16 — хордальный граф $H(16; \pm 1, 5)$ (слева). Граф G_1 степени 3 с числом вершин 64, несвязное объединение 4-х копий графов g_1 (справа)

диаметру и среднему расстоянию с наиболее изученными двумерными и трехмерными циркулянтными графами степеней $v = 4$ и $v = 6$ соответственно, полученными в [14, 16].

Статья организована следующим образом. В секции 1 даны определения операции многоуровневой композиции и многоуровневой хордальной сети. В секции 2 кратко описан алгоритм моделирования отжига, применяемый для оптимизации многоуровневых хордальных сетей по среднему расстоянию. В секции 3 представлены экспериментальные результаты построения многоуровневых хордальных сетей. Использование больших языковых моделей (LLM) для построения многоуровневых хордальных сетей и распараллеливания программы оптимизации представлено в секции 4.

1. Построение многоуровневых хордальных сетей. Рассмотрим общий процесс построения многоуровневых регулярных сетей с помощью операции многоуровневой композиции. Пусть система связи состоит из m уровней, сеть связи на каждом уровне представлена одинаковыми регулярными графами g_1 степени v_1 и числом вершин n_1 . Операция многоуровневой композиции состоит из двух шагов: (1) на первом шаге используется несвязное объединение графов всех уровней, в результате получается несвязный регулярный граф G_1 степени v_1 с числом вершин $N = m * n_1$, состоящий из m копий графов g_1 ; (2) на втором шаге берется сумма двух графов — графа G_1 степени v_1 с числом вершин N , полученного на первом шаге, и графа G_2 степени v_2 с числом вершин N , соединяющего все уровни. При этой операции каждое мультиребро заменяется одним ребром. В результате получается регулярный граф G с числом вершин N и степенью $v \leq v_1 + v_2$.

Рассмотрим пример многоуровневой композиции с четырьмя уровнями $m = 4$. На рис. 2 слева показан граф g_1 с числом вершин 16 и степенью 3 — хордальный граф $H(16; \pm 1, 5)$, представляющий сеть связи на одном уровне. На рис. 2 справа показан граф G_1 степени 3 с числом вершин 64, представляющий сети связи всех 4-х уровней как несвязное объединение 4-х копий графов g_1 . На рис. 3, слева, показан граф G_2 степени 3 с числом вершин 64 — хордальный граф $H(64; \pm 1, 15)$, служащий для соединения всех уровней. Полученный многоуровневый граф G степени 4, с числом вершин 64, числом уровней 4, показан на рис. 3, справа, как сумма двух графов — графа G_1 и графа G_2 .

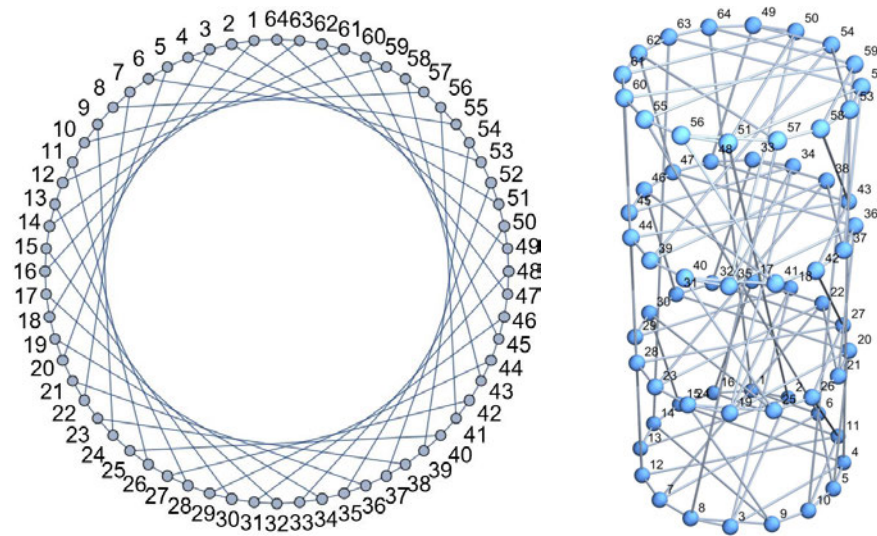


Рис. 3. Граф G_2 степени 3 с числом вершин 64 — хордальный граф $H(64; \pm 1, 15)$ (слева).

Многоуровневый граф G (справа) степени 4, с числом вершин 64, с числом уровней 4, как сумма двух графов — графа G_1 и графа G_2

Операцию многоуровневой композиции можно применять к различным классам графов: циркулянтным графам и их модификациям, обобщенным и модифицированным хордальным графам [6, 8], параметрически задаваемым сетям общего вида [17] и др. В данной работе эта операция применяется к хордальным графам (к хордальным кольцевым графам g_1 и модифицированным хордальным графам G_2). Будет показано, что построение многоуровневых регулярных сетей на основе хордальных графов позволяет сократить среднее расстояние и диаметр сети по сравнению с циркулянтами при одинаковых затратах оборудования (количества узлов и связей), что, соответственно, позволяет уменьшить задержки в сети и увеличить ее надежность.

Перечислим возможные модификации операции многоуровневой композиции в зависимости от количества вершин на уровнях и значений, их образующих.

При первом варианте многоуровневая сеть имеет одинаковое число вершин n_1 на всех уровнях, одинаковую степень и одинаковое множество S_1 образующих на всех уровнях и следующий вид параметрического описания: $MH(N, m, n_1, v, \{S_2\}, \{S_1\})$, где N — число ее вершин, m — число уровней, n_1 — число вершин на уровне, v — степень вершин сети, S_2 — множество образующих хордального графа G_2 , S_1 — множество образующих хордального графа g_1 . Например, граф G на рис. 3, справа, имеет следующее параметрическое описание: $MH(64, 4, 16, 4, \{\pm 1, 15\}, \{\pm 1, 5\})$. Набор введенных параметров позволяет компактно описывать все множество связей как между уровнями сети, так и внутри каждого уровня.

Во втором варианте в многоуровневой сети на каждом уровне могут быть хордальные графы с одинаковым числом вершин и степенью, но с разными образующими. В этом случае вместо множества образующих S_1 используется последовательность множеств образующих — свое множество $\{S_{1i}\}$ для каждого уровня $1 \leq i \leq m$. Третья модификация операции многоуровневой композиции состоит в использовании на каждом уровне хордальных графов с разным числом вершин и одинаковой степенью. В этом случае вместо одного числа вершин n_1 на каждом уровне используется различное число вершин $\{n_{1i}\}$ для каждого уровня $1 \leq i \leq m$. Такие структуры полезны, если многоуровневый граф

имеет простое число вершин. В настоящей работе используется только первая модификация, более предпочтительная в реализации в силу ее однородности и симметрии.

2. Оптимизация многоуровневых хордальных сетей. Для оптимизации (минимизации среднего расстояния) многоуровневых хордальных сетей использовался алгоритм моделирования отжига (SA) [18]. Алгоритм SA относится к классу пороговых стохастических алгоритмов безусловной оптимизации. Он допускает, в отличие от других подобных алгоритмов, шаги, приводящие к увеличению значений фитнес-функции $F(X)$ (среднего расстояния сети X). Рассмотрим суть этого итерационного алгоритма. На каждой итерации алгоритма в окрестности $d(X)$ текущего приближения к решению X выбираем случайное решение X' . Если разность $F(X') - F(X) < \varepsilon$, то в качестве нового текущего приближения к решению принимаем X' . В противном случае в окрестности $d(X)$ выбираем новое решение. Здесь ε — заданный положительный порог, величина которого по тому или иному закону убывает с ростом числа итераций t , так что имеет место предельное соотношение $\lim_{t \rightarrow \infty} \varepsilon(t) = 0$. Алгоритм в процессе поиска допускает ухудшение значений фитнес-функции до заданного порога ε , и этот порог в процессе итераций последовательно снижается до нуля. В алгоритме SA величина ε представляет собой случайную величину с математическим ожиданием, равным $\bar{\varepsilon}$, которому придается смысл «температуры» отжигаемого металла. Таким образом, в алгоритме SA переход от решения X к решению X' допускается с вероятностью:

$$p(X \rightarrow X') = \begin{cases} 1, & \text{если } F(X') \leq F(X), \\ \exp(-(F(X') - F(X))/\bar{\varepsilon}), & \text{если } F(X') > F(X). \end{cases}$$

Последняя формула означает, что если переход от точки X к точке X' приводит к уменьшению значения фитнес-функции $F(X)$, то этот переход осуществляется безусловно. В противном случае переход выполняется с вероятностью, которая убывает с ростом разности $F(X') - F(X)$ и уменьшением «температуры» $\bar{\varepsilon}$.

При инициализации процесса поиска многоуровневого графа G генерируются два случайных хордальных графа, вершины каждого из которых соединены случайным набором образующих, отдельно для графа g_1 с числом вершин n_1 и степенью 3 и графа G_2 с числом вершин N и степенью 3. После этого с помощью операции многоуровневой композиции формируется полный многоуровневый хордальный граф. При этом происходит преобразование мультиребер между двумя вершинами в одно ребро, проверяется его связность и однородность заданной степени. Если какое-либо из этих условий не выполняется, то генерация графа повторяется. Далее на каждом шаге алгоритма SA происходит случайная мутация текущего графа с целью поиска графа с лучшим средним расстоянием. Мутация состоит в случайном равновероятном изменении одной из образующих или графа g_1 , или G_2 , который выбирается с равной вероятностью. После мутации полный многоуровневый граф изменяется, проверяется на связность, однородность заданной степени, отсутствие мультиребер, и вычисляется значение его фитнес-функции (среднего расстояния). Далее, в соответствии с алгоритмом SA происходит вычисление вероятности его принятия в качестве нового текущего приближения к решению и переход на новую итерацию. После заданного числа итераций программа завершается.

В данной работе поиск оптимальной (субоптимальной) многоуровневой хордальной сети осуществлялся алгоритмом SA при заданном числе вершин и степени путем описанных выше случайных мутаций образующих исходных хордальных графов. При поиске перебирались все возможные значения числа уровней сети (все делители N).

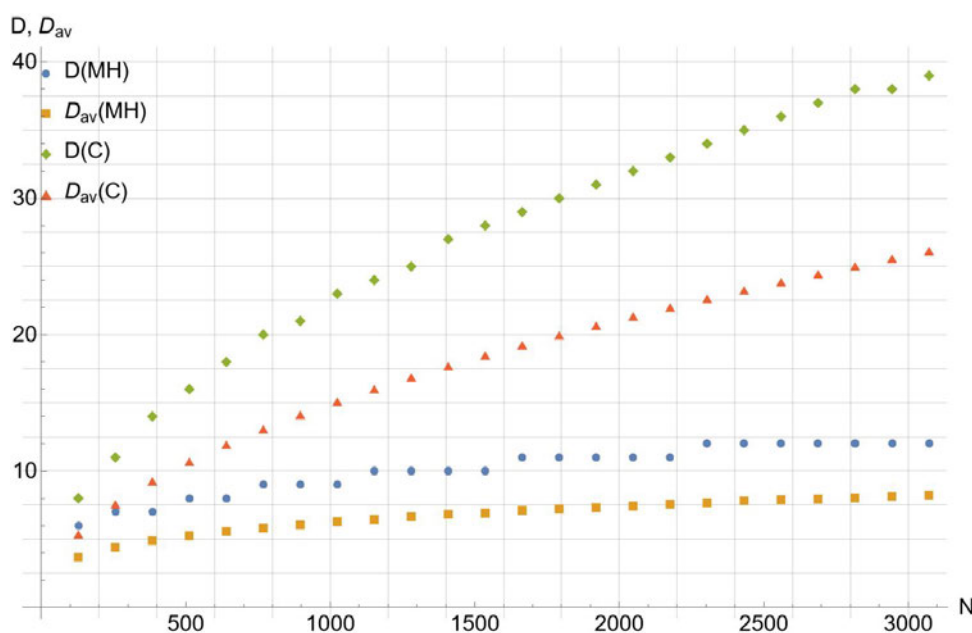


Рис. 4. Зависимость среднего расстояния и диаметра от числа вершин для оптимальных циркулянтов из [16] и найденных оптимальных MH

3. Экспериментальные результаты построения многоуровневых хордальных сетей. Проведем сравнение характеристик полученных многоуровневых хордальных сетей с известными оптимальными циркулянтами [14, 16] по трем следующим направлениям: (1) сравнение изменения среднего расстояния многоуровневых сетей при увеличении числа узлов для степеней четыре и шесть; (2) зависимости среднего расстояния от числа уровней в сети; (3) сравнение среднего расстояния многоуровневых сетей для числа узлов от 2^5 до $3 \cdot 2^{10}$ и степеней 4 и 6 с оптимальными циркулянтами, полученными в [14, 16].

На рис. 4 показаны полученные значения среднего расстояния для оптимальных циркулянтов (C) из [16] и найденных оптимальных многоуровневых хордальных сетей (MH) при степени $v = 4$ и с числом вершин N , меняющимся от 128 до 3072 с шагом, равным 128. На рис. 5 представлено отношение среднего расстояния для оптимальных циркулянтов к среднему расстоянию оптимальных MH сетей при тех же показателях N и v . Приведенные графики показывают, что многоуровневые хордальные сети имеют меньшее значение среднего расстояния в 1.47–3.17 раза, чем оптимальные циркулянты при тех же затратах оборудования.

Например, многоуровневый хордальный граф $MH(3072, 3, 1024, 4, \{\pm 2373, 2390\}, \{\pm 1, 368\})$ имеет $D = 12$, $D_{av} = 8.23681$, а оптимальный циркулянт $C(3072; \{39, 40\})$ имеет $D = 39$, $D_{av} = 26.13$. Экспериментальные исследования показали рост преимущества многоуровневых хордальных графов при росте числа вершин графа для заданной степени.

На рис. 6 представлена зависимость среднего расстояния от числа уровней сети для найденных оптимальных многоуровневых хордальных сетей MH с числом вершин $N = 1024$ и степенью четыре. Эксперименты показали, что для каждого числа вершин существует свое число уровней, при котором достигается минимальное значение среднего расстояния. Выбор оптимальных значений среднего расстояния многоуровневой сети при

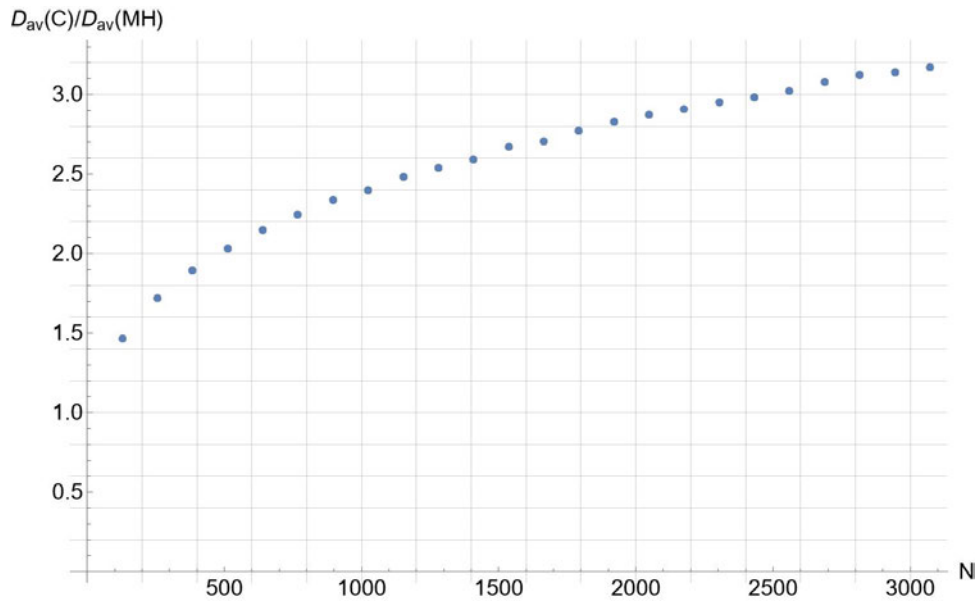


Рис. 5. Отношение среднего расстояния оптимальных циркулянтов из [16] к среднему расстоянию оптимальных MH

заданных N и v осуществлялся путем полного перебора всех значений числа уровней сети (всех делителей N) и выборе наилучшего возможного.

В табл. 1 приведены параметры описаний и значения D , D_{av} для некоторых найденных с помощью алгоритма моделирования отжига (суб) оптимальных многоуровневых хордальных сетей степеней 4 и 6 и оптимальных циркулянтов (обозначение N^*) из [14, 16].

Приведенные результаты из табл. 1 показывают, что многоуровневые хордальные сети степени 6 имеют меньшее среднее расстояние до 32 % по сравнению с оптимальными циркулянтами при одинаковых порядках графов. Указанное преимущество многоуровневых хордальных сетей над оптимальными циркулянтами увеличивается и далее при увеличении числа вершин.

4. Использование больших языковых моделей при построении многоуровневых сетей и распараллеливании программы оптимизации. При построении многоуровневых хордальных сетей для разработки программ на языке C использовались большие языковые модели (LLM) в качестве ИИ ассистента. С помощью LLM Qwen3-235B получены программы определения диаметра и среднего расстояния для заданных неориентированных графов, а также программа реализации алгоритма моделирования отжига, которая использовалась для оптимизации среднего расстояния параметрически описываемых графов. При обращении к LLM использовались простые запросы (промты), например:

Write a C program to compute the average distance of a given undirected graph represented by a list of edges или

Provide a C program for optimization of the average distance by simulated annealing algorithm of a given graph represented by a list of generators.

Совместно с запросами подавались фрагменты программ с описанием структур данных используемых графов. Сгенерированные на C программы принимались разработчиком для дальнейшего использования после многократного их тестирования и проверки на

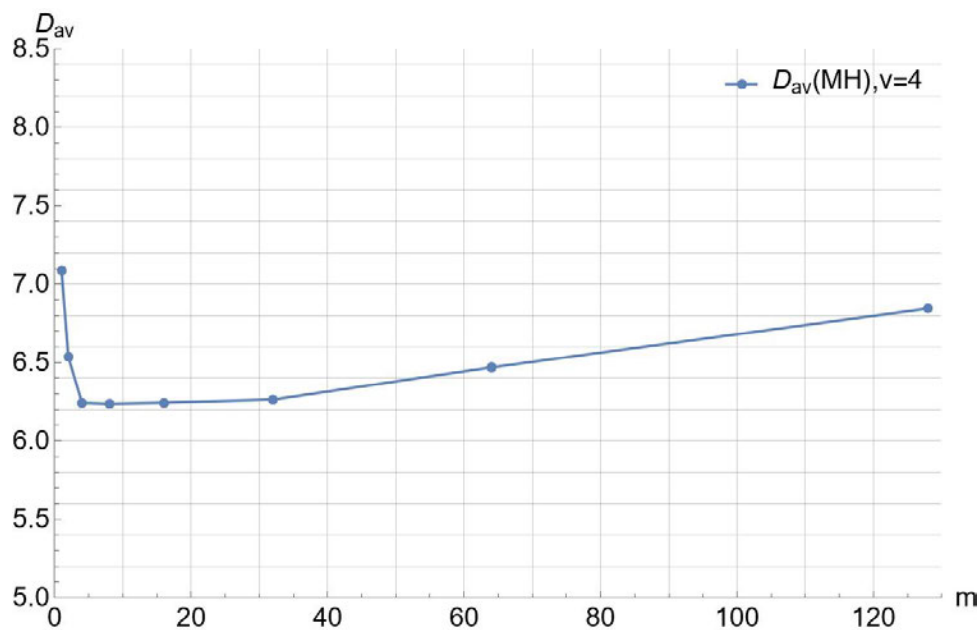


Рис. 6. Зависимость среднего расстояния от числа уровней для найденных оптимальных MH при числе вершин $N = 1024$ и степени $v = 4$

множестве графов с различными параметрами. Если возникали ошибки во время трансляции или исполнения, они сообщались LLM, которая генерировала новую версию программы, обычно этот процесс повторялся не более 2 раз.

С помощью LLM Gemini-2.5Pro произведено автоматическое распараллеливание последовательной версии программы генерации оптимальных многоуровневых хордальных сетей и получена многопоточная версия программы. При обращении к LLM использован следующий запрос с указанием названия нужной параллельной библиотеки, простой схемы распараллеливания и файла последовательной программы на C:

Give a parallel C program by parallelizing a given sequential C program to minimize the average distance of a graph G using the simulated annealing algorithm. To parallelize the given C program, use the OpenMP library with a given number of independent threads that interact only at the end of the optimization to select the threads with the minimum average distance of G among all threads, and derive this solution.

LLM сгенерировала многопоточную версию программы без ошибок трансляции и исполнения. Тестирование полученной программы при изменении числа потоков от 1 до 64 на процессоре Kynpeng 920 показало линейное ускорение и близкую к 1 эффективность, что объясняется минимальными обходами между потоками.

Отметим, что, хотя использование больших языковых моделей требует дополнительного времени на проверку сгенерированных алгоритмов и программ и их тщательного тестирования, применение LLM позволяет сократить общее время разработки программ, по нашим оценкам, на 20–30 %. С учетом ускоренного развития LLM и постоянного улучшения качества сгенерированного кода, использование ИИ помощников при разработке и распараллеливании программ достаточно успешно и перспективно.

Заключение. Предложена новая модель топологий сетей связи на основе обобщения хордальных сетей. Для конструирования многоуровневых сетей введена новая операция

Таблица 1

Оптимальные многоуровневые хордальные сети (N)
и оптимальные циркулянты (N^*) из [14] и [16]

N	m	n_1	v	D_{av}	D	$\{S_2\}, \{S_1\}$
32	2	16	4	2.51613	4	$\{\pm 1, 15\}, \{\pm 1, 5\}$
32*			4	2.71	4	$\{1, 7\}$
64	4	16	4	3.2381	5	$\{\pm 1, 15\}, \{\pm 1, 5\}$
64*			4	3.78	6	$\{1, 14\}$
128	4	32	4	3.64862	6	$\{\pm 21, 110\}, \{\pm 1, 10\}$
128*			4	5.35	8	$\{8, 9\}$
256	8	32	4	4.38946	7	$\{\pm 169, 218\}, \{\pm 1, 14\}$
256*			4	7.55	11	$\{11, 12\}$
512	2	256	4	5.25511	8	$\{\pm 94, 305\}, \{\pm 1, 178\}$
512*			4	10.68	16	$\{16, 17\}$
1024	8	128	4	6.23722	9	$\{\pm 40, 871\}, \{\pm 1, 46\}$
1024*			4	15.08	23	$\{23, 24\}$
2048	4	512	4	7.43028	11	$\{\pm 683, 1228\}, \{\pm 1, 356\}$
2048*			4	21.34	32	$\{32, 33\}$
3072	3	1024	4	8.23681	12	$\{\pm 2373, 2390\}, \{\pm 1, 368\}$
3072*			4	26.13	39	$\{39, 40\}$
32	2	16	6	2.01613	3	$\{\pm 4, 13\}, \{\pm 1, 7\}$
32*			6	2.03226	3	$\{1, 4, 10\}$
64	4	16	6	2.46627	4	$\{\pm 26, 21\}, \{\pm 1, 7\}$
64*			6	2.60317	4	$\{1, 4, 25\}$
128	8	16	6	2.97047	4	$\{\pm 22, 57\}, \{\pm 1, 5\}$
128*			6	3.33858	5	$\{1, 8, 54\}$
256	4	64	6	3.5201	5	$\{\pm 22, 91\}, \{\pm 1, 29\}$
256*			6	4.25098	6	$\{1, 47, 122\}$
512	4	128	6	4.17129	6	$\{\pm 210, 57\}, \{\pm 1, 49\}$
512*			6	5.39922	8	$\{1, 112, 139\}$
1024	2	512	6	4.87704	7	$\{\pm 277, 71\}, \{\pm 1, 229\}$
1024*			6	6.84066	10	$\{1, 24, 457\}$
2048	4	512	6	5.76962	9	$\{\pm 157, 141\}, \{\pm 1, 225\}$
2048*			6	8.673180	13	$\{901, 427, 397\}$
3072	4	768	6	6.75043	12	$\{\pm 71, 146\}, \{\pm 1, 195\}$
3072*			6	9.936828	15	$\{37, 113, 383\}$

многоуровневой композиции, которая позволяет использовать в качестве элементов уровней большой спектр регулярных графов, предложенных ранее в качестве структур вычислительных систем, объединяя их оптимальным образом. Операция многоуровневой композиции применена к классу модифицированных хордальных графов. Для минимизации среднего расстояния сетей нового класса топологий использован алгоритм моделирования отжига. Получены оптимальные (субоптимальные) многоуровневые регулярные сети степеней 4 и 6. Проведены исследования влияния параметров многоуровневой сети на величину среднего расстояния. Алгоритм синтеза оптимальных многоуровневых сетей разработан с помощью больших языковых моделей и реализован в последователь-

ной и параллельной версиях на кластере Kunpeng. Введенные многоуровневые сети на основе модифицированных хордальных графов позволили получить лучшие структурные характеристики, чем у циркулянтных сетей при тех же затратах оборудования. Для будущей работы планируется исследовать применение других классов графов для получения многоуровневых регулярных сетей, а также теоретически исследовать их характеристики и возможность построения эффективных алгоритмов маршрутизации в таких сетях.

Список литературы

1. Multilevel Network Analysis for the Social Sciences: Theory, Methods and Applications. Lazega, E., Snijders, T. (eds.). // Cham, Heidelberg: Springer. 2016.
2. Kivela M., Arenas A., Barthélemy M., Gleeson J. P., Moreno Y., Porter M. Multilayer Networks // Journal of Complex Networks. 2014. N 2 (3).
3. Кальней А. М. Модели многоуровневых сетей (краткий обзор) // Проблемы информатики. 2021. № 3. С. 5–20.
4. Кальней А. М., Родионов А. С. Анализ надежности многоуровневых сетей с ненадежными вершинами // Проблемы информатики. 2020. № 2. С. 5–15.
5. Hwang F. K. A survey on multi-loop networks // Theoret. Computer Science. 2003. V. 299.
6. Reyes, M. A., Dalfo, C., Fiol, M. A. Structural and Spectral Properties of Chordal Ring, Multi-Ring, and Mixed Graphs // Symmetry, 2024. 16. 1135.
7. Gutierrez J., Riaz T., Pedersen J., Labeaga S., Madsen O. Degree 3 networks topological routing // Image Processing and Communication. 2009. N 14.
8. Ledzinski, D., Smigiel, S., Zabłudowski, L. Analyzing methods of network topologies based on chordal rings // Turkish Journal of Electrical Engineering and Computer Sciences. 2018. V. 26: N 3, Article 25.
9. Монахова, Э. А., Монахов, О. Г. Метод автоматического поиска семейств оптимальных хордальных кольцевых сетей // Дискретный анализ и исследование операций. 2024. Т. 31. № 1. С. 85–108.
10. Монахова, Э. А., Монахов, О. Г. Открытие аналитических зависимостей параметров оптимальных хордальных сетей на основе анализа данных // Проблемы информатики. 2023. № 4. С. 37–48.
11. Ahmad M., Zahid Z., Zavaid M., and Bonyah E. Studies of Chordal Ring Networks via Double Metric Dimensions // Math. Problems in Engineering. 2022. (ArticleID 8303242).
12. Arden B. W. and Lee H. Analysis of Chordal Ring Network // IEEE Trans. on Computers. 1981. N C-30.
13. Morillo P., Comellas F., Fiol M. A. The optimization of Chordal Ring Networks // Communication Technology, Eds. Q. Yasheng and W. Xiuying. World Scientific, 1987. P. 295–299.
14. Huang, X., Ramos, A. F., Deng, Y. Optimal circulant graphs as low-latency network topologies // J. of Supercomputing. 2022. N 78. P. 13491–13510.
15. Deng Y., Guo M., Ramos A. F., Huang X., Xu Z., Liu W. Optimal low-latency network topologies for cluster performance enhancement // J. Supercomput, 2020. N 76 (12). P. 9558–9584.
16. Monakhova E. A Survey on Undirected Circulant Graphs // Discrete Mathematics, Algorithms and Applications. 2012. N 4 (1). 1250002.
17. Monakhov O., Monakhova E. A Class of Parametric Regular Networks for Multicomputer Architectures // Computacion y Sistemas. 2000. N 4. P. 85–93.
18. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой // Москва: МГТУ им. Н. Э. Баумана, 2017. 448 с.



Эмилия Анатольевна Монахова защитила кандидатскую диссертацию в Ленинградском электротехническом институте. В настоящее время является ведущим на-

учным сотрудником лаборатории системного моделирования и оптимизации Института вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия, e-mail: emilia@rav.sccc.ru. Она является автором двух книг и более 100 статей. Ее исследовательские интересы включают параллельную компьютерную архитектуру, сетевые топологии, алгоритмы маршрутизации, эволюционные вычисления.

Emilia A. Monakhova received the Ph.D. degree in computer science from Leningrad Electrotechnical Institute, Leningrad, USSR. She is currently a Leading Scientific Researcher in the System Modeling and Optimization Laboratory at the Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk, Russia, e-mail: emilia@rav.sccc.ru. She is the author of two books and more than 100 articles. Her research interests include parallel computer architecture, network topologies, routing algorithms, evolutionary computation.

Олег Геннадьевич Монахов получил степень кандидата технических наук в Таганрогском государственном радиотехническом уни-

верситете. С 2000 по 2002 год он был приглашенным профессором в Университете Айзу, Айзу-Вакамацу, Япония. В настоящее время является ведущим научным сотрудником лаборатории системного моделирования и оптимизации Института вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия, e-mail: monahov@rav.sccc.ru. Он является автором двух книг и более 120 статей. Его исследовательские интересы включают параллельные компьютерные архитектуры, сетевые топологии, алгоритмы маршрутизации, эволюционные вычисления, генетическое программирование, метаэвристические алгоритмы.



Oleg G. Monakhov received the Ph.D. degree in computer science from Taganrog State Radio Technical University, Taganrog, USSR. From 2000 to 2002, he was a Visiting Professor in the University of Aizu, Aizu-Wakamatsu, Japan. He is currently a Leading Scientific Researcher in the System Modeling and Optimization Laboratory at the Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk, Russia, e-mail: monakhov@rav.sccc.ru. He is the author of two books and 120 articles. His research interests include parallel computer architecture, network topologies, routing algorithms, evolutionary computation, genetic programming, metaheuristic algorithms.

Дата поступления — 06.11.2025

A GLOBAL OPTIMIZATION ALGORITHM FOR TUNING HYPERPARAMETERS OF MACHINE LEARNING METHODS

M. A. Usova, I. G. Lebedev, A. A. Shtanyuk, K. A. Barkalov

Lobachevsky State University,
603022, Nizhny Novgorod, Russia

DOI: 10.24412/2073-0667-2025-4-52-72

EDN: XGFNEQ

The paper discusses the problem of finding the best combination of hyperparameters of machine learning and artificial intelligence methods. In many cases, the efficiency (in a given metric) of the resulting solution for different values of hyperparameters can be quite different. In such problems, a significant issue is the potential for incorrect operation of the investigated artificial intelligence and machine learning methods within certain (a priori unknown) subregions of the hyperparameters search domain. Furthermore, the computational complexity of tuning makes manual or exhaustive search inappropriate. These characteristics have required the development of various intelligent automatic hyperparameter optimization methods. From a mathematical point of view, such a task can be represented as the problem of finding a global minimum of a function, given in the form of a “black box” and computable only in some part of the search domain. In this case, each computation of the objective function value at some point of the feasible domain may require significant computing resources. The objective function is assumed to satisfy the Lipschitz condition. The existence of subdomains where the objective function is undefined can be interpreted as the existence of some hidden, a priori unknown constraints of the problem. The authors propose an approach to solving this type of problem, which is an extension of the information-statistical global search algorithm (GSA) and takes into account the presence of undefined values of the objective function at some points. The algorithm partitions the search space with trial points and evaluates the characteristics of subregions based on the objective function values computed at their boundaries. If the function value at a point is unknown, the algorithm employs an estimate for this value, considering the size of the subregion under investigation. To minimize the number of redundant trials in subdomains where the function is not defined, the method parameter α was used that regulates the number of trial points in regions of non-computability. The solution of multidimensional problems implemented through reducing them to one-dimensional optimization problems using space-filling curves (Peano curves). The article provides a detailed description and a flowchart of the operation of the proposed search algorithm. The implementation of the global search algorithm for the case of a not everywhere computable objective function (GSA-N) was based on the iOpt open source framework of intelligent optimization methods. To carry out the experiments, a generator of test problems with hidden constraints GKLS-HC was developed. It is based on the GKLS generator, which allows generating multi-extremal functions with specified properties (number of minima, their regions of attraction, etc.). In the GKLS-HC generator, these functions were spoiled by areas of non-computability in the form of ellipsoids

The work was carried out with the support of the Ministry of Science and Higher Education of the Russian Federation (state assignment N FSWR-2023-0034) and Scientific and Educational Mathematical Center “Mathematics of Future Technologies”.

(the coordinates of centers and radii were generated randomly). The experimental results presented in the paper, obtained on a series of GKLS-HC test problems, demonstrate the reliability of global search and the efficiency of the developed algorithm. By adjusting the parameter α , it was possible to achieve the same performance of GSA-N operation as the basic GSA. The paper also considered the behavior of global optimization algorithms of the `scipy.optimize` library when solving problems with non-computable domains. In this study, the differential evolution and brute force methods showed the worst results, failing to solve this type of problem at all. The DIRECT and SHGO methods, although they solved the problems, were less effective than the developed GSA-N. Experiments were also conducted with hyperparameter tuning problems where undefined values of the quality metric arise. In these problems, certain hyperparameter combinations caused the method to return infinite values for the objective function. The LinearSVC classification algorithm was successfully tuned. GSA-N effectively solved the problem, identifying a better hyperparameter combination compared to standard scikit-learn algorithms. A time series prediction method from the FEDOT framework was also configured. During this experiment, GSA-N was compared to tuning algorithms from the popular Optuna framework. GSA-N achieved comparable performance in terms of the obtained the target metric value, significantly surpassing Optuna in solution time.

Key words: machine learning, hyperparameter tuning, global optimization, black-box functions, partially defined functions.

References

1. Zhou J. , Qiu Y., Zhu S., Armaghani D. J. , Li C., Nguyen H. , Yagiz S. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate // Eng. Appl. Artif. Intell. 2021. V. 97. P. 104015.
2. Yang W., Xia K., Fan S., Wang L., Li T., Zhang J., Feng Y. A Multi-Strategy Whale Optimization Algorithm and Its Application // Eng. Appl. Artif. Intell. 2022. V. 108. P. 104558.
3. Frazier P.I. A Tutorial on Bayesian Optimization // arXiv. 2018.
4. Archetti F., Candelieri A. Bayesian Optimization and Data Science. Cham: Springer Briefs in Optimization, 2019.
5. Jones D., Martins J. The direct algorithm: 25 years later // J. Glob. Optim. 2021. V. 79, N 3. P. 521–566.
6. Paulavicius R. and Zilinskas J. Simplicial Global Optimization. New York: Springer, 2014.
7. Paulavicius R., Sergeyev Y.D., Kvasov D.E., Zilinskas J. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization // Expert Syst. Appl. 2020. V. 144. P. 113052.
8. Sergeev Ya.D., Kvasov D.E. *Diagonalnye metody globalnoj optimizacii*. M.: Fizmatlit, 2008.
9. Liberti L., Kucherenko S. Comparison of deterministic and stochastic approaches to global optimization // Int. Trans. Oper. Res. 2005. V. 12. P. 263–285.
10. Sergeyev Y.D., Kvasov D.E., Mukhametzhonov M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // Sci. Rep. 2018. V. 8, N 1. P. 435.
11. Stripinis L., Paulavicius R. A new DIRECT-GLh algorithm for global optimization with hidden constraints // Optim. Lett. 2021. V. 15, N 6. P. 1865–1884.
12. Audet C., Batailly A., Kojtych S. Escaping unknown discontinuous regions in blackbox optimization // SIAM J. Optim. 2022. V. 32, N 3. P. 1843–1870.
13. Candelieri A. Sequential model based optimization of partially defined functions under unknown constraints // J. Glob. Optim. 2019. V. 79, N 2. P. 281–303.
14. Barkalov K. A., Strongin R. G. Metod globalnoj optimizacii s adaptivnym poryadkom proverki ogranichenij // Zhurn. vy'chisl. matem. i matem. fiz. 2002. T. 42, N 9. S. 1338–1350.

15. Strongin R. G., Barkalov K. A., Bevzuk S. A. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints // *Soft Comput.* 2020. V. 24, N 16. P. 11853–11865.
16. Sergeyev Y. D., Strongin R. G., Lera D. *Introduction to Global Optimization Exploiting Space-Filling Curves*. New York: Springer Briefs in Optimization, 2013.
17. Strongin R. G., Sergeyev Y. D. *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Dordrecht: Kluwer Academic Publishers, 2000.
18. Usova M. A., Barkalov K. A. An Algorithm for Finding the Global Extremum of a Partially Defined Function // *Communications in Computer and Information Science*. 2024. V. 1914. P. 147–161.
19. Barkalov K. A., et al. On solving the problem of finding kinetic parameters of catalytic isomerization of the pentane-hexane fraction using a parallel global search algorithm // *Mathematics*. 2022. V. 10, N 19. P. 3665.
20. Gubaydullin I. M., Enikeeva L. V., Barkalov K. A., Lebedev I. G., Silenko D. G. Kinetic modeling of isobutane alkylation with mixed C4 olefins and sulfuric acid as a catalyst using the asynchronous global optimization algorithm // *Commun. Comput. Inf. Sci.* 2022. V. 1618. P. 293–306.
21. Barkalov K. A., Lebedev I. G., Gergel V. P. Parallel Global Search Algorithm with Local Tuning for Solving Mixed-Integer Global Optimization Problems // *Lobachevskii Journal of Mathematics*. V. 7. N 42. 2021. P. 1492–1503.
22. Sysoev A. V., Kozinov E. A., Barkalov K. A., Lebedev I. G., Karchkov D. A., Rodionov D. M. Frejmvork metodov intelektualnoj evristicheskoj optimizacii iOpt // *V kn.: Superkompyuternye dni v Rossii: Trudy mezhdunarodnoj konferencii*. 2023. S. 179–185.
23. Ishodnyj kod frejmvorka iOpt. [Electron. Res.]: <https://github.com/aimclub/iOpt> (data obrasheniya: 26.01.2025).
24. Dokumentaciya iOpt. [Electron. Res.]: <https://iopt.readthedocs.io/ru/latest/> (data obrasheniya: 26.01.2025).
25. Gaviano M., Kvasov D. E., Lera D., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // *ACM Trans. Math. Softw.* 2003. V. 29, N 4. P. 469–480.
26. Storn R., Price K., *Differential Evolution — a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces* // *Journal of Global Optimization*. 1997. V. 11. P. 341–359.
27. Xiang Y., Gubian S., Suomela B., Hoeng J. Generalized Simulated Annealing for Efficient Global Optimization: the GenSA Package for R // *The R Journal*. 2013. V. 5, N 1.
28. Gablonsky J., Kelley C. A Locally-Biased form of the DIRECT Algorithm // *Journal of Global Optimization*. 2001. V. 21. P. 27–37.
29. Wales D. J., Doye J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms // *Journal of Physical Chemistry A*. 1997. V. 101. P. 5111.
30. Endres S. C., Sandrock C., Focke W. W. A simplicial homology algorithm for lipschitz optimisation // *Journal of Global Optimization*. 2018.
31. Filippou K., Aifantis G., Papakostas G. A., Tsekouras G. E. Structure learning and hyperparameter optimization using an automated machine learning (AutoML) pipeline // *Information*. 2023. V. 14, N 4. P. 232.
32. Automated modeling and machine learning framework FEDOT. [Electron. Res.]: <https://github.com/aimclub/FEDOT> (data obrasheniya: 25.07.2025).
33. Xu N. Time Series Analysis on Monthly Beer Production in Australia // *Highlights in Science, Engineering and Technology*. 2024. V. 94. P. 392–401.
34. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-Generation Hyperparameter Optimization Framework // *In Proceedings: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019. P. 2623–2631.

АЛГОРИТМ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ДЛЯ НАСТРОЙКИ ГИПЕРПАРАМЕТРОВ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

М. А. Усова, И. Г. Лебедев, А. А. Штанюк, К. А. Баркалов

ННГУ им. Н. И. Лобачевского,
603022, Нижний Новгород, Россия

УДК 519.853.4

DOI: 10.24412/2073-0667-2025-4-52-72

EDN: XGFNEQ

В статье рассматриваются задачи поиска наилучшего сочетания гиперпараметров методов машинного обучения и искусственного интеллекта. В таких задачах актуальной является проблема некорректной работы исследуемых методов ИИ и МО в некоторых (заранее неизвестных) подобластях области изменения гиперпараметров. С математической точки зрения такая задача может быть представлена как задача поиска глобального минимума функции, заданной в виде «черного ящика» и не всюду определенной в области поиска. Существование подобластей, где целевая функция является неопределенной, можно интерпретировать как наличие некоторых скрытых, заранее неизвестных ограничений. Предложен подход к решению такого рода задач, который является расширением информационно-статистического алгоритма глобального поиска и учитывает наличие неопределенных значений целевой функции в некоторых точках. В рамках предложенного алгоритма проводится разбиение области поиска точками испытаний и оцениваются характеристики подобластей на основе значений целевой функции, вычисленных на их границах. В случае отсутствия информации о значениях функции в алгоритме используются оценка, учитывающая размер исследуемой подобласти. Для сокращения количества испытаний в подобластях, в которых функция не определена, введен специальный параметр метода, позволяющий регулировать число точек испытаний в области невычислимости. Изложено подробное описание и приведена схема работы модифицированного алгоритма глобального поиска. Продемонстрированы результаты его сравнения с другими известными алгоритмами глобальной оптимизации, полученные при проведении численных экспериментов как с тестовыми функциями, так и с модельными задачами настройки гиперпараметров, в которых возникают неопределенные значения оптимизируемой метрики качества.

Ключевые слова: машинное обучение, настройка гиперпараметров, глобальная оптимизация, функции вида «черный ящик», частично определенные функции.

Введение. В настоящее время методы искусственного интеллекта (ИИ) и машинного обучения (МО) применяются для решения широкого круга задач. К их числу можно отнести ставшие уже классическими задачи распознавания образов, классификации, регрессии. Однако в большинстве методов ИИ и МО имеются гиперпараметры, от выбора конкретных значений которых может зависеть качество (в некоторой метрике) полученного решения. Тот факт, что разница в качестве получаемого решения при разных значениях

Работа выполнена при поддержке Министерства науки и высшего образования РФ (проект № FSWR-2023-0034) и научно-образовательного математического центра «Математика технологий будущего».

гиперпараметров может быть достаточно значительной, привел к возникновению класса задач настройки гиперпараметров (НРО — hyperparameter optimization). С математической точки зрения такие задачи соответствуют задачам глобальной оптимизации с фиксированными границами изменения переменных. При этом часть гиперпараметров может быть категориальной, то есть принимать значения из некоторого дискретного множества, что приводит к необходимости использовать методы оптимизации, способные решать задачи с дискретными параметрами.

Основная сложность настройки гиперпараметров состоит в том, что поиск их оптимального сочетания требует (для каждого выбранного варианта) решения исходной задачи машинного обучения, а значит, может быть довольно длительным по времени. Таким образом, любой подход будет ограничен в числе сочетаний гиперпараметров, которое метод глобальной оптимизации сможет проверить, прежде чем будет исчерпан доступный вычислительный ресурс.

Весьма распространенными методами настройки параметров являются метаэвристические (генетические, имитационные и аналогичные им) алгоритмы [1, 2]. Данные алгоритмы широко применяются при отсутствии формульного описания оптимизируемой функции (функция вида «черный ящик»), что характерно для рассматриваемых задач. Метаэвристические алгоритмы слабо зависят от числа параметров задачи, используют информацию с предыдущих итераций для выполнения текущей, однако в силу заложенной в методы случайности дают гарантию отыскания глобального оптимума только в вероятностном смысле.

Еще один подход к поиску глобального оптимума — байесовская оптимизация, также применяемая для решения задач с функциями вида «черный ящик» [3, 4]. Для своей работы методы байесовской оптимизации используют стохастическую модель оптимизируемой функции. Модель итерационно обновляется на основе накапливаемой в процессе поиска оптимума информации, позволяя на каждой очередной итерации оценить наиболее вероятное положение глобального оптимума. Методы байесовской оптимизации обладают более высокой эффективностью по сравнению с метаэвристическими алгоритмами, но в значительной степени подвержены влиянию «проклятия размерности».

Хорошие результаты при использовании методов байесовской оптимизации достигаются, когда целевая функция задачи соответствует определенной стохастической модели, например, гауссовскому процессу. Однако в глобальной оптимизации существуют и иные предположения о виде функции, которые дают замечательные результаты. Одним из таких допущений о решаемой задаче является предположение об ограниченности относительных изменений целевой функции. В этом случае говорят, что функция удовлетворяет условию Липшица, а решаемая задача называется задачей липшицевой глобальной оптимизации. Для класса задач липшицевой глобальной оптимизации разработан целый ряд эффективных алгоритмов [5–8], которые превосходят многие другие методы глобальной оптимизации [9, 10].

В рассматриваемых задачах настройки гиперпараметров актуальной является проблема некорректной работы исследуемых алгоритмов ИИ и МО в некоторых (заранее неизвестных) подобластях области поиска. При этом программные реализации настраиваемых методов могут вести себя абсолютно по-разному: в лучшем случае информируют исследователя о том, что задача не может быть решена, в худшем — возвращают недопустимое значение (NaN, inf) в качестве найденного решения. Проблема возникновения недопусти-

мых комбинаций, которая не имеет особого значения в случае «ручной» настройки, существенно осложняет работу фреймворков автоматической настройки гиперпараметров.

Указанное свойство с математической точки зрения можно интерпретировать либо как наличие в задаче некоторых скрытых ограничений [11], либо как наличие неизвестных областей, в которых целевая функция не является непрерывной [12], либо как частичную вычислимость целевой функции в области поиска [13–15]. В такой постановке задача оптимизации существенно усложняется, т. к. область допустимых сочетаний параметров является заранее неопределенной.

Настоящая работа продолжает развитие одного из эффективных детерминированных методов решения задач липшицевой глобальной оптимизации — информационно-статистического алгоритма глобального поиска [16, 17]. В статье приведено описание нового алгоритма, адаптированного для работы с частично определенной целевой функцией. Данный алгоритм основан на подходе, предложенном авторами ранее в [18]. Проведено исследование поведения других известных алгоритмов при решении задач с не всюду вычислимой целевой функцией. Продемонстрированы результаты численных экспериментов как с тестовыми функциями, так и с модельными задачами настройки гиперпараметров, в которых возникают неопределенные значения целевой функции. А именно, проведена настройка гиперпараметров метода LinearSVC при решении задачи классификации на наборе данных Iris, а также метода предсказания значений временного ряда для датасета monthly beer production с платформы Kaggle.

1. Постановка задачи настройки гиперпараметров. В общем виде задача настройки гиперпараметров (которой соответствует задача глобальной оптимизации) может быть сформулирована следующим образом:

$$\phi^* = \phi(y^*) = \min_{y \in D} \phi(y), \quad (1)$$

$$D = \{y \in R^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N\},$$

где $y = (y_1, y_2, \dots, y_N)$ — вектор варьируемых параметров, D — N -мерный гиперкуб, N — размерность решаемой задачи. О целевой функции $\phi(y)$ мы делаем следующие предположения.

1) Целевая функция может быть многоэкстремальной, недифференцируемой и, более того, заданной в форме «черного ящика» (т. е. в виде некоторой подпрограммы, на вход которой подается аргумент, а выходом является соответствующее значение функции).

2) Каждое вычисление функции в некоторой точке допустимой области может требовать значительных вычислительных ресурсов.

3) Целевая функция удовлетворяет условию Липшица

$$|\phi(y') - \phi(y'')| \leq L \|y' - y''\|, \quad y', y'' \in D, \quad (2)$$

где $0 < L < \infty$ — константа Липшица.

4) В некоторой подобласти I области поиска D (в частном случае, в одной или нескольких ее точках) целевая функция может быть не определена. Тогда функция $\phi(y)$ определена и вычислима лишь в подобласти $Q = D \setminus I$ (положительного объема). Отметим, что исходя из опыта решения прикладных оптимизационных задач [19, 20], суммарный объем области невычислимости I составляет небольшую долю объема области поиска D .

Последнее предположение делает невозможными применение известного информационно-статистического алгоритма глобального поиска [17] или других методов липшицевой

оптимизации [6, 8]. Для решения таких задач нами была разработана модификация алгоритма глобального поиска, основанная на предложенном ранее подходе из [18].

1.1. *Редукция размерности.* Используя кривые типа развертки Пеано, однозначно отображающие отрезок $[0,1]$ на N -мерный единичный гиперкуб

$$D = \{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\}, \quad (3)$$

исходную задачу (1) можно редуцировать к одномерной задаче

$$f^*(x) = \phi(y(x^*)) = \min_{x \in [0,1]} \{\phi(y(x))\}, \quad (4)$$

что позволяет применить для ее решения эффективные алгоритмы одномерной оптимизации.

Известно, что схема редукции размерности с использованием кривых Пеано сопоставляет многомерной задаче с липшицевой целевой функцией (1) задачу (4) с одномерной целевой функцией, удовлетворяющей условию Гельдера

$$|f(x') - f(x'')| \leq K\rho(x', x''), \quad x', x'' \in [0,1], \quad (5)$$

где $\rho(x', x'') = |x' - x''|^{1/N}$ — метрика Гельдера, N — размерность исходной задачи, а коэффициент K связан с константой Липшица L соотношением $K \leq 2L\sqrt{N+3}$ [17].

2. Алгоритм решения задач с частично определенной целевой функцией.

Алгоритм глобального поиска предполагает построение последовательности точек *поисковых испытаний* $y^i \in D$, в которых вычисляются значения целевой функции $z^i = \varphi(y^i)$. Согласно используемой схеме редукции размерности, проведение испытания предполагает вычисление значения $y^i = y(x^i)$, $x^i \in [0,1]$. При этом каждой точке испытания x^i ставится в соответствие индекс v^i , определяемый по правилу

$$v(x^i) = \begin{cases} -1, & \text{если } x^i \text{ — граничная точка,} \\ 0, & \text{если } x^i \text{ — невычислимая точка,} \\ 1, & \text{если } x^i \text{ — внутренняя точка.} \end{cases} \quad (6)$$

Результатом испытания является набор значений $(x^i, y^i = y(x^i), z^i = \varphi(y^i), v^i = v(x^i))$.

Накопленная после проведения k испытаний поисковая информация хранится в порядке возрастания значений координаты x^i в множестве Ω_k , в котором соответствующие записи для удобства перенумерованы нижним индексом.

На каждом шаге алгоритма производятся вычисление оценки константы Гельдера, определение характеристик поисковых интервалов, выбор наиболее перспективного интервала и вычисление точки y^{k+1} для проведения очередного испытания. Правила вычисления характеристики интервала для задач с частично определенной целевой функцией представлены в виде Алгоритма 1.

Алгоритм 1. Вычисление характеристики i -го интервала.

Require: $\Omega_k, i, z^*, \mu, r > 1, 0 < \alpha \leq 1$

Ensure: R

1: **if** $v(x_{i-1}) = 1$ and $v(x_i) = 1$ **then**

▷ обе точки являются внутренними

2: $R \leftarrow \Delta_i + \frac{(z_i - z_{i-1})^2}{(r\mu)^2 \Delta_i} - 2 \frac{(z_i + z_{i-1} - 2z^*)}{r\mu}$

```

3: else if  $v(x_i) = 1$  then                                 $\triangleright$  точка  $x_i$  является внутренней
4:    $R \leftarrow 2\Delta_i - 4\frac{(z_i - z^*)}{r\mu}$ 
5: else if  $v(x_{i-1}) = 1$  then                                 $\triangleright$  точка  $x_{i-1}$  является внутренней
6:    $R \leftarrow 2\Delta_i - 4\frac{(z_{i-1} - z^*)}{r\mu}$ 
7: else                                                        $\triangleright$  обе точки являются невычислимыми или граничными
8:    $R \leftarrow \alpha(1 - \frac{1}{r})^2 \Delta_i$ 
9: end if
10: return R

```

Значения r и α являются параметрами алгоритма (их смысл пояснен ниже), а Δ_i обозначает длину i -го интервала в новой метрике,

$$\Delta_i = (x_i - x_{i-1})^{1/N}.$$

Подробное описание правил алгоритма глобального поиска на случай не всюду вычислимой целевой функции для решения задачи (4) приведено в виде Алгоритма 2.

Алгоритм 2. Алгоритм глобального поиска для не всюду вычислимой целевой функции (АГП-Н).

Require: $\varphi(y), D, \varepsilon, K_{max}, r > 1, 0 < \alpha \leq 1$

Ensure: y^{min}

```

1:  $\Omega_k \leftarrow \{(x^1 = 0.0, y^1 = y(x^1), v^1 = v(x^1)), (x^2 = 1.0, y^2 = y(x^2), v^2 = v(x^2))\}$ 
2:  $\Omega_k \leftarrow \Omega_k \cup \{(x^3 = 0.5, y^3 = y(x^3), z^3 = \varphi(y^3), v^3 = v(x^3))\}$ 
3:  $z^* \leftarrow z^3, k \leftarrow 3; t \leftarrow 2$ 
4: while  $\Delta_t \geq \varepsilon$  and  $k \leq K_{max}$  or  $v(x_t) \neq 1$  and  $v(x_{t-1}) \neq 1$  do
5:    $\mu \leftarrow \max \left\{ \frac{|z_i - z_{i-1}|}{\Delta_i}, v(x_{i-1}) = 1 \text{ and } v(x_i) = 1, 1 \leq i \leq k \right\}$ 
6:   if  $\mu = 0$  then
7:      $\mu = 1$ 
8:   end if
9:   for  $i = 1$  to  $k$  do
10:     $R_i \leftarrow R(\Omega_k, i, z^*, \mu, r, \alpha)$ 
11:   end for
12:    $t \leftarrow \min \{\arg \max \{R_i, 1 \leq i \leq k\}\}$ 
13:   if  $v(x_{t-1}) = 1$  and  $v(x_t) = 1$  then
14:      $x^{k+1} \leftarrow \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{1}{2r} \left[ \frac{|z_t - z_{t-1}|}{\mu} \right]^N$ 
15:   else
16:      $x^{k+1} \leftarrow \frac{x_t + x_{t-1}}{2}$ 
17:   end if
18:    $y^{k+1} \leftarrow y(x^{k+1}); z^{k+1} \leftarrow \varphi(y^{k+1}); v^{k+1} \leftarrow v(x^{k+1})$ 
19:    $z^* = \min \{z^*, z^{k+1}\}$ 
20:    $\Omega_k \leftarrow \Omega_k \cup \{(x^{k+1}, y^{k+1}, z^{k+1}, v^{k+1})\}$ 
21:    $k \leftarrow k + 1$ 
22: end while
23:  $min \leftarrow \arg \min \{z^i, 1 < i < k\}$ 
24: return  $y^{min}$ 

```

Замечание 1 (о связи с АГП). В случаях, когда задача не содержит скрытых ограничений или проводимые алгоритмом испытания не попадают в существующие области

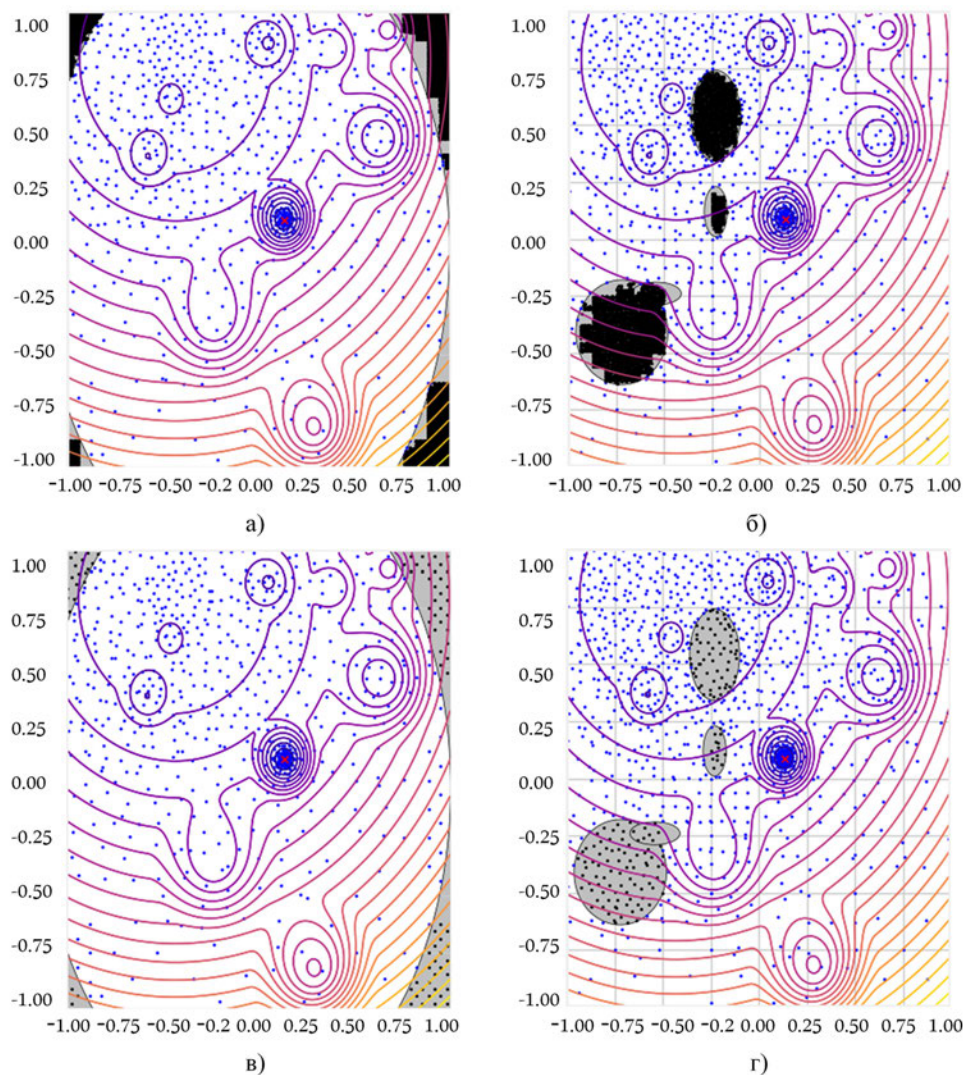


Рис. 1. Распределение точек испытаний с различными видами невычислимых областей и настройкой параметра α : (а) область невычислимости на границе области поиска, $\alpha = 1.0$; (б) случайно расположенные области невычислимости, $\alpha = 1.0$; (в) область невычислимости на границе области поиска, $\alpha = 0.08$; (г) случайно расположенные области невычислимости, $\alpha = 0.08$

невычислимости, АПП-Н полностью повторяет работу своего прототипа — алгоритма глобального поиска.

Замечание 2 (о параметре α). Параметр $0 < \alpha \leq 1.0$ позволяет регулировать плотность сетки испытаний в области невычислимости I , $I \subset D$. Чем меньше параметр, тем менее перспективными для алгоритма будут интервалы с граничными точками в областях невычислимости. При максимально допустимом параметре $\alpha = 1.0$ метод густо усеивает точками испытаний подобласти, в которых функция не вычислима, возникает эффект «черного пятна» (точки испытаний в таких под областях на рисунке обозначены черным цветом), см. рис. 1, а, и рис. 1, б. При уменьшении параметра α точек этих испытаний становится значительно меньше, что наглядно видно на рис. 1, в, и рис. 1, г.

Замечание 3 (о существовании вычислимого отрезка на кривой Пеано). В случае попадания точек испытаний в область невычислимости на первых итерациях, ал-

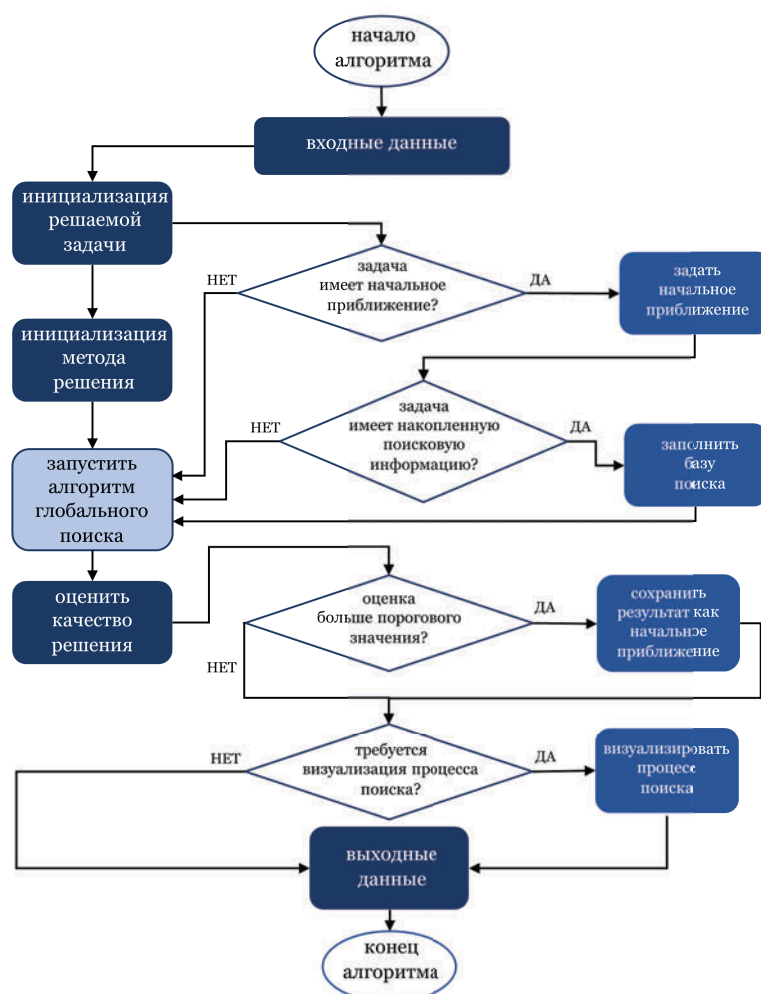


Рис. 2. Блок-схема общего алгоритма функционирования фреймворка iOpt

горитм будет выбирать лучший интервал на основе длин интервалов (т. е. будет строиться равномерная сетка). Так как мы предполагаем, что область вычислимости целевой функции $Q = D \setminus I$ — положительного объема, то последующее сгущение сетки приведет к попаданию очередной точки испытания в область Q .

Замечание 4 (о решении задач с дискретными параметрами). Алгоритм глобального поиска допускает обобщение для решения задач, в которых часть переменных являются непрерывными, а часть может принимать только дискретные значения [21]. Это позволяет применять АГП-Н в задачах, где часть параметров объекта оптимизации может принадлежать некоторому дискретному множеству. Подобные задачи часто возникают при настройке гиперпараметров методов машинного обучения.

3. Программная реализация. Алгоритм глобального поиска, предназначенный для решения задач с не всюду вычислимой целевой функцией (АГП-Н), был реализован на базе открытого фреймворка методов интеллектуальной оптимизации iOpt [22, 23].

Фреймворк позволяет проводить точную настройку параметров моделей и методов, используемых в прикладных исследованиях в различных научных областях. Характерными примерами задач, решаемых фреймворком iOpt, являются задачи настройки гиперпараметров методов машинного обучения. В таких задачах часто возникает рассматриваемая

в данном исследовании проблема частичной определенности критериев в отсутствии формульного описания исследуемой модели вида «черный ящик».

На рис. 2 приведена блок-схема общего алгоритма функционирования фреймворка.

Фреймворк iOpt написан на Python в виде системы взаимосвязанных классов. Реализация описанного в разделе 2 алгоритма содержится в базовом классе Method. Описанный далее в разделе 4.1.1 генератор задач GKLS со скрытыми ограничениями (GKLS-НС) реализован в виде класса GKLSHiddenConstraint — наследника интерфейса Problem. В фреймворк добавлена возможность визуализации процесса решения задач со скрытыми ограничениями с использованием StaticPainterNDListener. Доступны различные способы отрисовки линий уровня целевой функции: по равномерной сетке, только по точкам поисковых испытаний, по точкам испытаний с использованием интерполяции. Подробное описание возможностей фреймворка представлено в документации [24].

4. Результаты вычислительных экспериментов.

4.1. *Решение серий тестовых задач.* Решение серии задач с известными свойствами является одним из традиционных способов оценки качества работы методов глобального поиска. Однако доступные наборы тестовых задач не содержат областей невычислимости целевой функции, поэтому для тестирования разработанного алгоритма АГП-Н был предложен способ генерации тестовых задач со скрытыми ограничениями.

В данном разделе дано краткое описание тестовых задач, а также приведены результаты экспериментов, демонстрирующие надежность глобального поиска и эффективность разработанного алгоритма оптимизации.

4.1.1. *Описание тестовых задач.* Для создания тестовых задач GKLS-НС использовался генератор GKLS [25], позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами.

Генерация задачи состоит в определении выпуклой квадратичной функции, дополненной полиномами более высокого порядка для введения локальных минимумов. Каждый тестовый класс, предоставляемый генератором GKLS, состоит из 100 функций, построенных случайным образом, и определяется размером области поиска, количеством локальных минимумов, значением глобального минимума, радиусом области притяжения глобального минимума, расстоянием от глобального минимума до вершины квадратичной функции. Другие необходимые параметры, такие как точки всех локальных минимумов, их области притяжения и значения, выбираются генератором случайным образом.

Данные функции были дополнены областями невычислимости в форме эллипсоидов (координаты центров и длины радиусов генерировались случайно), при попадании в которые функция возвращает исключение, сообщающее о невозможности вычислить ее значение.

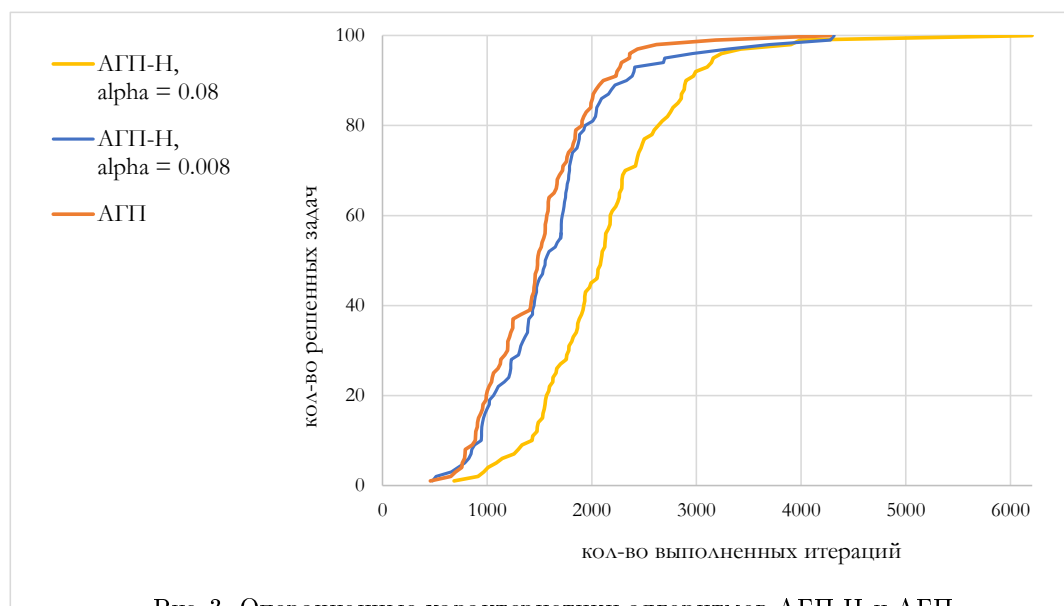
В текущей версии фреймворк iOpt использует генератор GKLS для формирования задач с размерностью от 2 до 5, числом локальных минимумов, равным 10, и областью поиска от -1 до 1 по каждой переменной. Радиус области притяжения глобального минимума и расстояние от глобального минимума до вершины квадратичной функции варьируется в зависимости от размерности задачи. Количество генерируемых областей невычислимости задавалось равным 4. Области невычислимости могут пересекаться, при этом не включают точку известного глобального минимума функций GKLS по построению. Радиусы областей невычислимости выбираются случайно из диапазона $[0.05, 0.25]$.

Если говорить о программной реализации, то генератор задач реализован в виде класса GKLSHiddenConstraint, конструктор которого принимает размерность генерируемой

Таблица 1

Результаты решения задач GKLS и GKLS-НС средствами фреймворка iOpt

класс задач	α	среднее число итераций
GKLS-НС	0.08	2151
	0.008	1635
GKLS	–	1510



задачи (от 2 до 5), номер функции (от 1 до 100), а также параметр для настройки генератора случайных чисел для воспроизведения экспериментов, после чего автоматически устанавливает остальные параметры генератора.

4.1.2. *Результаты решения тестовых задач с помощью АГП-Н.* Вычислительный эксперимент был проведен с использованием фреймворка iOpt, описанного в разделе 3. АГП-Н запускался на серии задач GKLS-НС с точностью поиска минимума $\varepsilon = 0.001$, параметром надежности $r = 4.2$ и двумя значениями параметра α : $\alpha = 0.08$ и $\alpha = 0.008$. Для полноценного сравнения алгоритм АГП-Н также был запущен на серии задач GKSL (т. е. на задачах без подобластей, в которых целевая функция является неопределенной), что соответствует поведению исходного АГП.

Результаты эксперимента приведены в табл. 1.

На рис. 3 показаны операционные характеристики, которые демонстрируют для каждого количества итераций число задач из тестовой выборки, решенное с заданной точностью $\varepsilon = 0.001$.

Примеры визуализации процесса решения задачи GKLS-НС № 38 представлены на рис. 4. Черными точками отмечены точки испытаний, попавшие в подобласти неопределенности функции. Красной точкой отмечено найденное алгоритмом решение.

Результаты проведенных экспериментов показывают, что после выполнения 2000 итераций АГП-Н с параметром $\alpha = 0.008$ будет получена оценка оптимума с требуемой точностью для 80 % задач. Для решения оставшихся 20 % задач требуется (в худшем случае)

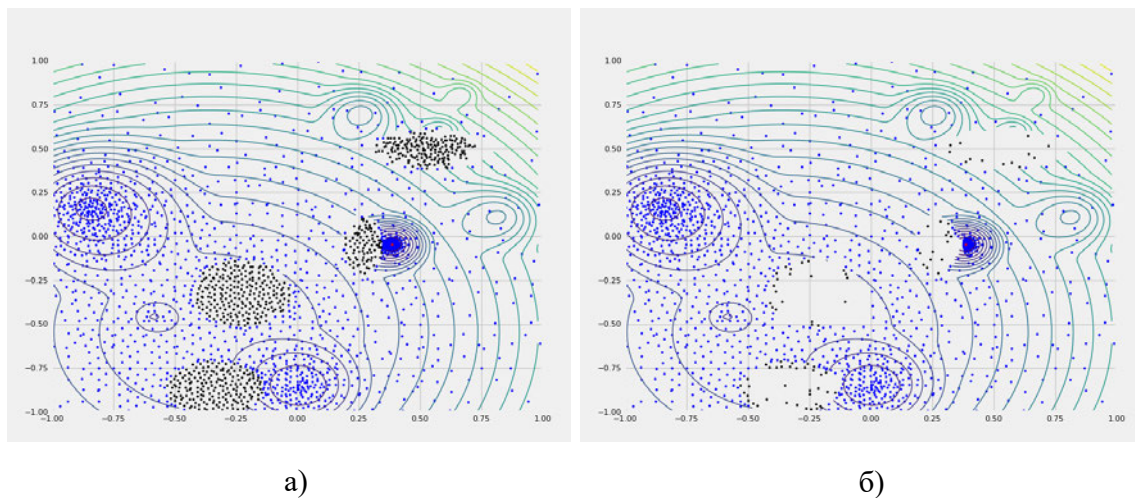


Рис. 4. Графики линий уровня задачи GKLS-НС № 38 с точками проводимых испытаний, полученные средствами фреймворка iOpt: (а) $\alpha = 0.08$, (б) $\alpha = 0.008$

4315 поисковых испытаний. Отметим, что за счет регулировки параметра α получилось добиться такой же скорости работы, как у АГП при решении серии задач GKLS (в терминах числа испытаний, требующихся для решения задач с заданной точностью).

4.2. *Сравнение АГП-Н с известными алгоритмами оптимизации.* АГП-Н может быть использован не только в задаче настройки гиперпараметров различных методов, но и для решения рядовых оптимизационных задач из различных прикладных областей. В связи с этим имеет смысл привести сравнение реализованного метода с существующими оптимизаторами на задачах с не всюду вычислимой целевой функцией.

4.2.1. *Описание алгоритмов оптимизации.* В данной статье сравнение проводилось с алгоритмами глобальной оптимизации библиотеки *scipy.optimize*, а именно с методами *differential_evolution* [26], *dual_annealing* [27], *direct* [28], *basinhopping* [29], *shgo* [30], *brute*.

Метод дифференциальной эволюции (differential evolution) — один из самых известных генетических алгоритмов вещественной оптимизации. Во многих практических задачах метод обеспечивает приемлемое решение, т.к. не использует информацию о градиенте функции, а схемы мутации и рекомбинации позволяют ему «выходить» из локальных минимумов.

Двойной отжиг (dual annealing) — алгоритм глобальной оптимизации, основанный на комбинировании алгоритмов классического и быстрого имитационного отжига. Алгоритм имитирует процесс отжига в металлургии. Отжиг заключается в нагревании и контролируемом охлаждении металла для формирования устойчивой кристаллической решетки: при высокой температуре системы алгоритм использует более широкий диапазон решений, при снижении температуры диапазон поиска становится все меньше, пока не будет найден глобальный оптимум.

Direct (Dividing RECTangles), как и АГП, предназначен для решения задач липшицевой глобальной оптимизации. Особенностью метода является отказ от оценивания константы Липшица. В процессе своей работы алгоритм разбивает область поиска на гиперинтервалы, среди которых выбираются потенциально оптимальные (подлежащие дальнейшему разбиению), после чего в их центрах проводится вычисление значений целевой функции.

Перепрыгивание через бассейн (basin-hopping) — алгоритм глобальной оптимизации, осуществляющий поиск оптимума путем последовательного выполнения локальной оптимизации (по умолчанию используя метод Монте-Карло) с последующим принятием или отклонением новых координат на основе текущего лучшего значения функции.

SHGO (simplicial homology global optimization) — сравнительно новый алгоритм глобальной оптимизации, основанный на применении симплициальной гомологии. Алгоритм использует концепции из комбинаторной интегральной теории гомологии для поиска под областей, которые приблизительно локально выпуклы. В отличие от многих других алгоритмов глобальной оптимизации, использующих теорию графов и методы кластеризации, SHGO создает начальные точки, которые сходятся к уникальным локальным минимумам, что улучшает его производительность.

Метод brute реализует метод «грубой силы» (brute force), т. е. полный перебор.

Алгоритмы *differential_evolution*, *dual_annealing*, *basinhopping* относятся к классу стохастических алгоритмов (т. е. каждый запуск поиска может найти другое решение), потому поиск глобального минимума с использованием данных алгоритмов требует многократных запусков.

4.2.2. *Результаты проведенного сравнения.* В рамках проведенного эксперимента были получены следующие результаты.

При наличии у задачи скрытых ограничений худшее поведение демонстрируют алгоритмы *differential_evolution* и *brute*. Алгоритм *differential_evolution* полностью исчерпывает все выделенные ему итерации, не находя при этом корректный ответ. Полный перебор в свою очередь в качестве результата работы может вернуть что угодно.

Алгоритмы *dual_annealing* и *basinhopping* в ряде случаев справляются с поставленной оптимизационной задачей. При этом в рамках эксперимента количество вычислений целевой функции алгоритмом *dual_annealing* составило порядка 4000, в то время как *basinhopping* мог выполнять и 140000 вычислений для одной задачи.

Алгоритмы *direct* и *shgo* демонстрируют хорошие результаты, хоть и не всегда обладают сходимостью. Среднее число итераций алгоритма *direct* составило 94 (соответствует 1981 вычислениям целевой функции). Алгоритм *shgo* в каждой задаче полностью исчерпал выделенные ему 200 итерации, выполняя в среднем 4482 вычисления целевой функции. В сравнении с данными методами АПП-Н работает эффективнее (отметим, что одна итерация АПП-Н соответствует одному вычислению значения целевой функции) и обладает лучшей сходимостью к глобальному оптимуму.

В табл. 2 собрана информация о возможном поведении исследованных алгоритмов с задачами с не всюду определенной целевой функцией, обнаруженная при решении задач GKLS-НС. Алгоритм АПП-Н всегда находит глобальный минимум, тогда как *differential_evolution* всегда возвращает значение nan. Остальные алгоритмы могут вернуть как глобальный, так и локальный минимум, при этом алгоритмы *dual_annealing* и *basinhopping* в ряде случаев могут вернуть значения nan и $1e+100$ соответственно, а алгоритм *brute*, как уже было отмечено, и вовсе может вернуть в качестве ответа что угодно.

4.3. Решение задач настройки методов машинного обучения.

4.3.1. *Настройка гиперпараметров метода LinearSVC.* В качестве первой модельной задачи оптимизации, в которой встречается недопустимое сочетание параметров, рассмотрим настройку гиперпараметров метода LinearSVC из библиотеки *scikit-learn* при решении задачи классификации.

Таблица 2

Возможные выявленные результаты работы алгоритмов глобальной оптимизации библиотеки *scipy.optimize* на задачах GKLS-НС

алгоритм	возможный результат поиска			
	глобальный минимум	локальный минимум	nan	1e+100
АПП-Н	●			
differential_evolution			●	
dual_annealing	●		●	
basinhopping	●	●		●
direct	●	●		
shgo	●	●		
brute	●	●	●	●

Исследование проводилось на классическом датасете Iris, который содержит характеристики трех различных видов ириса и часто используется в качестве тестового примера для алгоритмов машинного обучения. Набор данных Iris состоит из 150 записей (по 50 записей на каждый вид ириса), содержащих по 5 атрибутов. Датасет включен в библиотеку машинного обучения *scikit-learn*.

Задача заключалась в максимизации метрики $F1$, характеризующей качество построения классификатора. Для LinearSVC варьировались 2 дискретных и 1 непрерывный параметр: тип функции потерь $loss$, $loss \in \{hinge, squared_hinge\}$; параметр $dual$, $dual \in \{true, false\}$, который позволяет выбирать между решением двойственной или основной задачи оптимизации; коэффициент регуляризации C , $C \in [1, 6]$.

Для возможности воспроизведения результатов параметр $random_state$ был зафиксирован в значении 10. Остальные параметры метода LinearSVC рассматривались заданными по умолчанию.

Отметим, что сочетание гиперпараметров $loss = hinge$ и $dual = False$ приводит к «падению» метода LinearSVC (метод выбрасывает исключение и не выдает значения целевой метрики).

Поставленная задача решалась в области изменения гиперпараметров с использованием следующих алгоритмов: АПП-Н (предложенный в данной статье) с использованием параметра надежности $r = 3.5$ и ограничением на число испытаний $K_{max} = 100$; GridSearchCV (метод полного перебора по равномерной сетке); RandomizedSearchCV (метод случайного поиска).

Полученные результаты представлены в табл. 3. Отметим, что в процессе своей работы оба алгоритма библиотеки *scikit-learn* решили задачу с выдачей предупреждения о возникновении недопустимого сочетания параметров. Алгоритм АПП-Н нашел лучшее значение $F1 = 0.973$ при сочетании значений гиперпараметров $C = 1.3125$, $loss = squared_hinge$, $dual = False$.

4.3.2. Настройка гиперпараметров метода предсказания значений временного ряда. В качестве второй модельной задачи рассмотрим задачу настройки гиперпараметров методов предсказания значений временного ряда (time series forecasting). Предсказание значений производилось с использованием фреймворка FEDOT [31, 32] (фреймворк с открытым исходным кодом для задач автоматизированного моделирования и машинного обучения, AutoML). FEDOT поддерживает полный цикл решения задачи, включающий препроцес-

Таблица 3

Результаты настройки гиперпараметров метода LinearSVC
при решении задачи классификации на датасете Iris

метод	финальная F1
АГП-Н	0.973
GridSearchCV	0.960
RandomizedSearchCV	0.960

синг, выбор модели, настройку, кроссвалидацию и т. д. Фреймворк использует ML-модели, в основном из стандартных библиотек *sklearn*, *statsmodels* и *keras*.

Исследование проводилось на датасете *monthly beer production*, который содержит 476 записей. Указанный датасет представлен на платформе Kaggle и часто используется для тестирования алгоритмов машинного обучения (см., например, [33]). Построенный для решения поставленной задачи в FEDOT пайплайн включает в себя комбинацию из двух преобразований: *lagged* и *cgru*. Lagged-преобразование выполняется путем взятия значения переменной в предыдущий момент времени и включения его в модель в качестве признака в текущий момент времени. Для этого данные временного ряда сдвигаются на определенное количество шагов, которые называются лагом или временной задержкой. В качестве используемой нейронной сети выступает рекуррентная сеть (GRU), использующая свертку (CGRU).

У lagged-преобразования имеется один параметр *window_size*, который изменяется от 5 до 250 с шагом 1. Поскольку этот параметр имеет большое количество значений, при оптимизации будем рассматривать его как непрерывный (с округлением до единиц). У CGRU варьировались 4 дискретных и 2 непрерывных параметра: *cnn1_kernel_size* $\in \{4, 5\}$, *cnn2_kernel_size* $\in \{5, 6\}$, *cnn1_output_size* $\in \{32, 64\}$, *cnn2_output_size* $\in \{32, 64\}$; *hidden_size* $\in [20, 200]$, *learning_rate* $\in [0.0005, 0.005]$.

Остальные параметры были зафиксированы в значениях по умолчанию: *batch_size* = 64, *num_epochs* = 50, *optimizer* = 'adamw', *loss* = 'mse'.

Таким образом, фреймворк iOpt решал задачу минимизации метрики *MSE*, варьируя 3 непрерывных и 4 дискретных параметра, каждый из которых принимал 2 значения (всего 16 комбинаций). Отметим, что при некоторых сочетаниях параметров *window_size*, *cnn1_kernel_size*, *cnn1_output_size*, *cnn2_kernel_size* и *cnn2_output_size* невозможно вычислить значение критерия (метод возвращает бесконечное значение функции). Такие точки рассматривались как невычислимые.

Для сравнения поставленная задача также была решена при помощи известного фреймворка оптимизации гиперпараметров Optuna [34]. Оба решателя были интегрированы в FEDOT и задействованы в вычислительном эксперименте.

Для решения поставленной задачи было выставлено ограничение на 1000 поисковых испытаний. В iOpt после 950 «глобальных» итераций глобального метода запускалось локальное уточнение с использованием метода Хука-Дживса. Результаты проведенного эксперимента отражены в табл. 4.

Число точек, в которых не получилось вычислить критерий оптимизации, составило порядка 10 % от общего числа точек поисковых испытаний. Кривые, соответствующие полученному предсказанию, приведены на рис. 5. Темная линия соответствует истинным значениям временного ряда, голубая — предсказанным значениям, полученным после ис-

Таблица 4

Результаты решения задачи настройки гиперпараметров метода предсказания значений временного ряда на датасете *monthly beer production* с использованием фреймворков *iOpt* и *Optuna*

фреймворк	время решения, сек.	финальная MSE
iOpt	6574.2	13.5
Optuna	9554.1	13.4

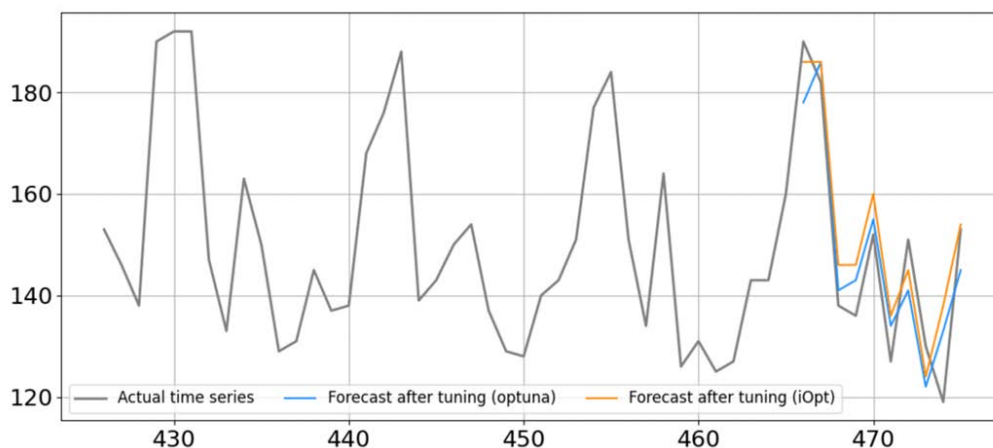


Рис. 5. Визуализация предсказания и истинного значения временного ряда

пользования *Optuna*, желтая линия — предсказанным значениям, полученным после использования *iOpt*.

Закключение. В настоящей работе рассматривалась актуальная проблема решения задач настройки гиперпараметров методов искусственного интеллекта (ИИ) и машинного обучения (МО). Сложность таких задач заключается в большой трудоемкости оценки метрик качества методов ИИ и МО для заданного сочетания гиперпараметров, что ограничивает возможное число проверяемых комбинаций. Одновременно с этим исследуемые методы ИИ и МО могут работать некорректно в некоторых заранее неизвестных подобластях всей области изменения гиперпараметров; фактически, область допустимых сочетаний гиперпараметров является неопределенной. Проблема возникновения некорректных сочетаний гиперпараметров, которая ранее не имела особого значения при «ручной» настройке, при использовании автоматической настройки стала весьма актуальной.

Для решения указанного класса задач была разработана модификация одного из эффективных детерминированных методов решения задач глобальной оптимизации — информационно-статистического алгоритма глобального поиска. Предложенный алгоритм глобального поиска для не всюду вычислимых функций (АГП-Н) реализован на базе фреймворка методов интеллектуальной оптимизации с открытым исходным кодом *iOpt*.

В статье подробно описаны вычислительные правила и приведена схема работы АГП-Н. Представлены результаты экспериментов на нескольких сотнях тестовых задач, демонстрирующие надежность предложенного подхода к обработке неопределенных значений оптимизируемой функции. Также приведены результаты сравнения работы АГП-Н и стандартных алгоритмов оптимизации из библиотеки *SciPy*, которые подтверждают эффективность разработанного метода при решении тестового класса задач.

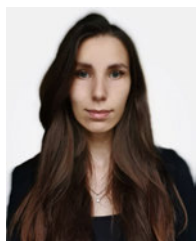
В качестве модельных примеров были решены несколько задач настройки гиперпараметров, где возникает проблема возникновения недопустимых сочетаний гиперпараметров, при которых невозможно оценить целевую метрику качества. В первом примере была проведена настройка гиперпараметров классического алгоритма классификации LinearSVC. АГП-Н успешно решил поставленную задачу, найдя лучшее сочетание гиперпараметров по сравнению со стандартными алгоритмами из библиотеки scikit-learn. В качестве второй модельной задачи была рассмотрена настройка гиперпараметров метода предсказания значений временного ряда. В ходе эксперимента было проведено сравнение АГП-Н с алгоритмами настройки из широко известного фреймворка Optuna. АГП-Н показал сравнимый результат (в терминах полученного значения целевой метрики), значительно обогнав Optuna по времени, затраченному на решение задачи.

Список литературы

1. Zhou J. , Qiu Y., Zhu S., Armaghani D. J. , Li C., Nguyen H. , Yagiz S. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate // Eng. Appl. Artif. Intell. 2021. V. 97. P. 104015. DOI: 10.1016/j.engappai.2020.104015.
2. Yang W., Xia K., Fan S., Wang L., Li T., Zhang J., Feng Y. A Multi-Strategy Whale Optimization Algorithm and Its Application // Eng. Appl. Artif. Intell. 2022. V. 108. P. 104558. DOI: 10.1016/j.engappai.2021.104558.
3. Frazier P.I. A Tutorial on Bayesian Optimization // arXiv. 2018. DOI: 10.48550/arXiv.1807.02811.
4. Archetti F., Candelieri A. Bayesian Optimization and Data Science. Cham: Springer Briefs in Optimization, 2019. DOI: 10.1007/978-3-030-24494-1.
5. Jones D., Martins J. The direct algorithm: 25 years later // J. Glob. Optim. 2021. V. 79, № 3. P. 521–566. DOI: 10.1007/s10898-020-00952-6.
6. Paulavicius R. and Zilinskas J. Simplicial Global Optimization. New York: Springer, 2014. DOI: 10.1007/978-1-4614-9093-7.
7. Paulavicius R., Sergeyev Y.D., Kvasov D.E., Zilinskas J. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization // Expert Syst. Appl. 2020. V. 144. P. 113052. DOI: 10.1016/j.eswa.2019.113052.
8. Сергеев Я. Д., Квасов Д. Е. Диагональные методы глобальной оптимизации. М.: Физматлит, 2008.
9. Liberti L., Kucherenko S. Comparison of deterministic and stochastic approaches to global optimization // Int. Trans. Oper. Res. 2005. V. 12. P. 263–285.
10. Sergeyev Y.D., Kvasov D.E., Mukhametzhanov M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // Sci. Rep. 2018. V. 8, № 1. P. 435.
11. Stripinis L., Paulavicius R. A new DIRECT-GLh algorithm for global optimization with hidden constraints // Optim. Lett. 2021. V. 15, № 6. P. 1865–1884. DOI: 10.1007/s11590-021-01726-z.
12. Audet C., Batailly A., Kojtych S. Escaping unknown discontinuous regions in blackbox optimization // SIAM J. Optim. 2022. V. 32, № 3. P. 1843–1870. DOI: 10.1137/21m1420915.
13. Candelieri A. Sequential model based optimization of partially defined functions under unknown constraints // J. Glob. Optim. 2019. V. 79, № 2. P. 281–303. DOI: 10.1007/s10898-019-00860-4.
14. Баркалов К. А., Стронгин Р. Г. Метод глобальной оптимизации с адаптивным порядком проверки ограничений // Журн. вычисл. матем. и матем. физ. 2002. Т. 42, № 9. С. 1338–1350.

15. Strongin R. G., Barkalov K. A., Bevzuk S. A. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints // *Soft Comput.* 2020. V. 24, № 16. P. 11853–11865. DOI: 10.1007/s00500-020-05078-1.
16. Sergeyev Y. D., Strongin R. G., Lera D. *Introduction to Global Optimization Exploiting Space-Filling Curves*. New York: Springer Briefs in Optimization, 2013. DOI: 10.1007/978-1-4614-8042-6.
17. Strongin R. G., Sergeyev Y. D. *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Dordrecht: Kluwer Academic Publishers, 2000.
18. Usova M. A., Barkalov K. A. An Algorithm for Finding the Global Extremum of a Partially Defined Function // *Communications in Computer and Information Science*. 2024. V. 1914. P. 147–161. DOI: 10.1007/978-3-031-52470-7_13.
19. Barkalov K. A., et al. On solving the problem of finding kinetic parameters of catalytic isomerization of the pentane-hexane fraction using a parallel global search algorithm // *Mathematics*. 2022. V. 10, № 19. P. 3665. DOI: 10.3390/math10193665.
20. Gubaydullin I. M., Enikeeva L. V., Barkalov K. A., Lebedev I. G., Silenko D. G. Kinetic modeling of isobutane alkylation with mixed C4 olefins and sulfuric acid as a catalyst using the asynchronous global optimization algorithm // *Commun. Comput. Inf. Sci.* 2022. V. 1618. P. 293–306. DOI: 10.1007/978-3-031-11623-0_20.
21. Barkalov K. A., Lebedev I. G., Gergel V. P. Parallel Global Search Algorithm with Local Tuning for Solving Mixed-Integer Global Optimization Problems // *Lobachevskii Journal of Mathematics*. V. 7. № 42. 2021. P. 1492–1503.
22. Сысоев А. В., Козинев Е. А., Баркалов К. А., Лебедев И. Г., Карчков Д. А., Родионов Д. М. Фреймворк методов интеллектуальной эвристической оптимизации iOpt // В кн.: *Суперкомпьютерные дни в России: Труды международной конференции*. 2023. С. 179–185.
23. Исходный код фреймворка iOpt. [Электрон. рес.]: <https://github.com/aimclub/iOpt> (дата обращения: 26.01.2025).
24. Документация iOpt. [Электрон. рес.]: <https://iopt.readthedocs.io/ru/latest/> (дата обращения: 26.01.2025).
25. Gaviano M., Kvasov D. E., Lera D., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // *ACM Trans. Math. Softw.* 2003. V. 29, № 4. P. 469–480.
26. Storn R., Price K., Differential Evolution — a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces // *Journal of Global Optimization*. 1997. V. 11. P. 341–359.
27. Xiang Y., Gubian S., Suomela B., Hoeng J. Generalized Simulated Annealing for Efficient Global Optimization: the GenSA Package for R // *The R Journal*. 2013. V. 5, № 1.
28. Gablonsky J., Kelley C. A Locally-Biased form of the DIRECT Algorithm // *Journal of Global Optimization*. 2001. V. 21. P. 27–37.
29. Wales D. J., Doye J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms // *Journal of Physical Chemistry A*. 1997. V. 101. P. 5111.
30. Endres S. C., Sandrock C., Focke W. W. A simplicial homology algorithm for lipschitz optimisation // *Journal of Global Optimization*. 2018.
31. Filippou K., Aifantis G., Papakostas G. A., Tsekouras G. E. Structure learning and hyperparameter optimization using an automated machine learning (AutoML) pipeline // *Information*. 2023. V. 14, № 4. P. 232.
32. Automated modeling and machine learning framework FEDOT. [Электрон. рес.]: <https://github.com/aimclub/FEDOT> (дата обращения: 25.07.2025).
33. Xu N. Time Series Analysis on Monthly Beer Production in Australia // *Highlights in Science, Engineering and Technology*. 2024. V. 94. P. 392–401. DOI: 10.54097/4z3krj13.

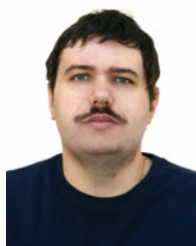
34. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-Generation Hyperparameter Optimization Framework // In Proceedings: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019. P. 2623–2631. DOI: 10.1145/3292500.3330701.



Усова Марина Андреевна — ассистент кафедры Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики ННГУ им. Н. И. Лобачевского.

Область научных интересов: модели и методы решения задач глобальной оптимизации, разработка средств визуализации данных. E-mail: usova@itmm.unn.ru.

Marina Usova — Assistant of the Department of Mathematical Software and Supercomputer Technologies of the Institute of Information Technologies, Mathematics and Mechanics, Lobachevsky State University of Nizhny Novgorod. Research interests: models and methods for solving global optimization problems, development of data visualization tools. E-mail: usova@itmm.unn.ru.

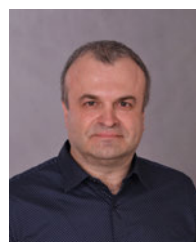


Лебедев Илья Геннадьевич — заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений кафедры Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики

Национального исследовательского нижегородского государственного университета им. Н. И. Лобачевского, Область научных интересов: методы глобальной и локальной оптимизации, параллельные алгоритмы, программирование для графических процессоров, CUDA. Является автором и соавтором 61 научной работы. E-mail: ilya.lebedev@itmm.unn.ru.

Ilya Lebedev — PhD., Head of the Laboratory of Supercomputer Technologies and High-Performance Computing, Department of Mathematical Software and Supercomputing Technologies, Institute of Information Technology, Mathematics and Mechanics, Nizhny Novgorod

State University. N. I. Lobachevsky. His research interests include algorithms of global and local optimization, parallel algorithms, GPU programming, CUDA. He is the author or coauthor of more than 30 papers in these areas. E-mail: ilya.lebedev@itmm.unn.ru.



Штанюк Антон Александрович — кандидат технических наук, доцент, доцент кафедры информатики и автоматизации научных исследований Национального исследовательского нижегородского государственного университета им. Н. И. Лобачевского.

Количество печатных работ 53. Область научных интересов: моделирование и проектирование сложных систем, языки и технология программирования, обучение программированию, алгоритмы и структуры данных. E-mail: anton.shtanyuk@itmm.unn.ru.

Anton Shtanyuk — PhD., Associate Professor at the Department of Informatics and Research Automation of the Institute of Information Technology, mathematics and mechanics at Lobachevsky State University of Nizhny Novgorod. The number of publications: 50. Research interests: modelling and design of complex systems, programming languages and technology, programming teaching, algorithms and data structures. E-mail: anton.shtanyuk@itmm.unn.ru.



Баркалов Константин Александрович — доктор технических наук, доцент, заведующий кафедрой Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики

ННГУ им. Н. И. Лобачевского. Область научных интересов: математические модели, методы и программные средства для решения задач гло-

бальной оптимизации; параллельные вычисления. Является автором и соавтором более 140 научных работ в указанных областях. E-mail: konstantin.barkalov@itmm.unn.ru.

Konstantin Barkalov — Doctor of Technical Sciences, Professor, Head of the Department of Mathematical Software and Supercomputer Technologies, Institute of

Information Technologies, Mathematics and Mechanics, Lobachevsky State University of Nizhny Novgorod. Research interests: mathematical models, methods and software for solving global optimization problems; parallel computing. E-mail: konstantin.barkalov@itmm.unn.ru.

Дата поступления — 04.04.2025

AUTOMATIC PROGRAM CONSTRUCTION WITH ACCELERATORS USAGE BASED ON THE ACTIVE KNOWLEDGE CONCEPT IN LUNA SYSTEM

V. E. Malyshkin^{*,**,***}, V. A. Perepelkin^{*,**,***}, V. A. Spirin^{*,**}

^{*}Novosibirsk State University,
630090, Novosibirsk, Russia

^{**}Institute of computational mathematics and mathematical geophysics SB RAS,
630090, Novosibirsk, Russia

^{***}Novosibirsk State Technical University,
630073, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-73-88

EDN: ZCPPTC

As accelerators are designed for specific types of computations (e. g., GPUs, NPU, FPGAs), they can significantly improve the performance of corresponding programs. In fact, most of the computing power in modern multicomputers is provided by accelerators. However, using accelerators in programs requires knowledge of system programming, which makes development more difficult for experts in other domains. For example, it is necessary to manage data transfers between CPUs and accelerators, implement load balancing, and schedule computations on accelerators. Achieving good efficiency requires different optimization methods in different subject domains.

One way to reduce the labor costs of developing programs that use accelerators is to offload some complex routine tasks from humans to an automation system. Such systems can lower the required level of competence for developing efficient programs by performing automatic program construction. Potentially, these automation systems can also produce programs with higher efficiency than average hand-coded implementations. Since no universal automation approach exists for such systems, it is necessary to develop various approaches for particular classes of applied problems and types of accelerators. One method that can be applied here is the active knowledge concept [1].

Automatic construction of parallel programs using the active knowledge concept is performed based on computational models (CMs). CMs are designed to represent knowledge in a certain subject domain. For simplicity, a CM can be viewed as a bipartite directed graph consisting of operations and variables. Each variable represents a certain value within the subject domain. Each operation represents the derivation of its output variables (on outgoing arcs) from its input variables (on incoming arcs). Operations are supplied with code fragments (CFs) in the form of conventional subroutines (procedures). To solve a problem in this CM's subject domain, we define a VW-task on the CM, where V is the set of input variables and W is the set of output variables to be computed. A VW-plan is then derived from the VW-task (if it exists). A VW-plan is a subgraph of the CM containing the necessary operations and variables to compute the variables in W from the variables in V. Using the VW-plan, a program can be constructed that takes values for the variables in V and computes values for the intermediate variables and the variables in W. Such programs can be parallel if different variables can be evaluated independently.

This research was carried out under the state contract with ICM&MG SB RAS FWNM-2025-0005.

The actual program construction (VW-plan derivation and/or program generation) is performed by an active knowledge system. An example of such a system, explored in this paper, is the LuNA system [2, 3]. The LuNA system takes a prepared VW-plan and performs program generation; the generated program is then executed by the runtime system. The runtime system uses a thread pool to schedule and execute operations. The thread pool consists of one queue and multiple worker threads that take available tasks from the queue and execute them. In a distributed system, multiple instances of the runtime system can execute the generated program. These instances communicate to decide which operation will be executed on which node.

In this paper, we consider an extension of the LuNA runtime system to support accelerators. In particular, we examine the Huawei Ascend neural processor [4]. We propose a high-level specification called CoFaNA (Code Fragment Notation for Accelerators) for describing computations to be executed on the Ascend accelerator. We also propose an extension of the thread pool to perform computations executed on both CPUs and Ascends. The thread pool was extended to have three queues: one for tasks to be executed on CPUs, another for tasks to be executed on Ascends, and a third for heterogeneous tasks that can be executed on either CPUs or Ascends. We also split the worker threads into two groups: one group can execute tasks on CPUs (including heterogeneous tasks), and the other group can execute tasks on Ascends.

Next, based on the implementation of support for the Ascend accelerator, we propose a subsystem that can potentially use any accelerator. By implementing support for a particular accelerator in a separate extension module (plugin), the subsystem can use subprograms from this module during program generation and execution in the runtime system. An extension module implements interfaces for the code generator, context, and thread pool, and also provides metainformation. This metainformation is used during program generation to link necessary shared libraries. The context is used to initialize or deinitialize libraries used in the extension module. The code generator can parse various high-level specifications (for example, the CoFaNA format for the Ascend accelerator). The thread pool can be implemented in various ways to use accelerators more efficiently.

An experimental study of using the Ascend accelerator was carried out on an Atlas 800 server [11]. We considered the problems of block multiplication of dense matrices and convolution of seismic traces [10]. The results show that LuNA implementations have execution times comparable to hand-coded C++ programs. The CoFaNA format helped to reduce the labor costs required to develop programs that use the Ascend accelerator.

In conclusion, this research explored the automatic construction of programs that use accelerators, based on the active knowledge concept. As a result of the work, the LuNA system was extended to support the Huawei Ascend accelerator and potentially other accelerators.

In the future, we plan to implement support for other accelerators and, if necessary, expand the subsystem. We also plan to research improvements for using the Huawei Ascend processor in the LuNA system: expanding the CoFaNA format, considering alternative implementations of the thread pool, and automatically selecting the precision of floating-point numbers based on the properties of the applied problem being solved.

Key words: active knowledge, LuNA system, accelerator, Huawei Ascend processor, automatic program construction, high-level specification, subsystem for accelerators support.

References

1. Malyshkin V. E. Active Knowledge, LuNA and Literacy for Oncoming Centuries // LNCS, V. 9465. 2015. P. 292–303. DOI: 10.1007/978-3-319-25527-9_19.
2. Malyshkin V. E., Perepelkin V. A. LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem // Proceedings of the 11th International Conference on Parallel

Computing Technologies (PaCT-2011), LNCS, 2011. V. 6873, P. 53–61. DOI: 10.1007/978-3-642-23178-0_5.

3. Malyshkin V. E., Perepelkin V. A. Postroenie baz aktivnyh znaniy dlya avtomaticheskogo konstruirovaniya reshenij prikladnyh zadach na osnove sistemy LuNA // *Parallel'nye vychislitel'nye tekhnologii — XVIII vserossiyskaya nauchnaya konferenciya s mezhdunarodnym uchastiem, PaVT'2024*, g. Chelyabinsk, 2–4 aprelya 2024 g. Korotkie stat'i i opisaniya plakatov. Chelyabinsk: Izdatel'skij centr YUUrGU, 2024. P. 57–68. DOI: 10.14529/pct2024 (In Russian).

4. Liang X. Ascend AI Processor Architecture and Programming: Principles and Applications of CANN. Elsevier, 2020. DOI: 10.1016/C2020-0-00270-7.

5. Malyshkin V., Perepelkin V., Schukin G. Scalable Distributed Data Allocation in LuNA Fragmented Programming System // *The Journal of Supercomputing*, 2017. V. 73, Iss. 2, Springer. P. 726–732. DOI: 10.1007/s11227-016-1781-0.

6. Malyshkin V. E., Perepelkin V. A., Schukin G. A. Distributed Algorithm for Data Allocation in Luna Fragmented Programming System // *“Problems of informatics”*. 2017, N 1. P. 74–88.

7. Using Operator Samples [Electron. Res.]: https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/82RC1alpha001/opdevg/tbeaicpudev/atlasopdev_10_0025.html.

8. AscendCL architecture and basic concepts [Electron. Res.]: https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/82RC1alpha001/appdevg/aclcppdevg/aclcppdevg_000004.html.

9. Spirin V. A. Razrabotka algoritmov avtomatizacii primeneniya NPU v sisteme LuNA // *Parallel'nye vychislitel'nye tekhnologii — XVIII vserossiyskaya nauchnaya konferenciya s mezhdunarodnym uchastiem, PaVT'2024*, g. Chelyabinsk, 2–4 aprelya 2024 g. Korotkie stat'i i opisaniya plakatov. Chelyabinsk: Izdatel'skij centr YUUrGU, 2024. P. 165–176. DOI: 10.14529/pct2024 (In Russian).

10. Khairtdinov M. S. Algoritmy potочноj svertki v zadachah aktivnogo vibrosejsmoakusticheskogo monitoringa / M. S. Khairtdinov, G. M. Voskoboynikova, G. S. Seduhina // *Geosibir'*, 2017 (In Russian).

11. Laboratoriya iskusstvennogo intellekta i informacionnyh tehnologiy (In Russian) [Electron. Res.]: <https://icmmg.nsc.ru/ru/content/pages/laboratoriya-iskusstvennogo-intellekta-i-informacionnyh-tehnologiy>.

12. Chen L. Deep learning and practice with mindspore. Springer Nature, 2021. DOI: 10.1007/978-981-16-2233-5.

13. Feng W., Maghareh R., Wang K. T. A. Extending DPC++ with Support for Huawei Ascend AI Chipset // *International Workshop on OpenCL*. 2021. P. 1–4. DOI: 10.1145/3456669.3456684.

14. Gu R., Becchi M. A comparative study of parallel programming frameworks for distributed GPU applications // *Proceedings of the 16th ACM International Conference on Computing Frontiers*. 2019. P. 268–273. DOI: 10.1145/3310273.3323071.

15. Robson M. P., Buch R., Kale L. V. Runtime coordinated heterogeneous tasks in Charm++ // *2016 Second International Workshop on Extreme Scale Programming Models and Middlewar (ESPM2)*. IEEE, 2016. P. 40–43. DOI: 10.1109/ESPM2.2016.011.

16. Bauer M. et al. Legion: Expressing locality and independence with logical regions // *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2012. P. 1–11. DOI: 10.1109/SC.2012.71.

17. Augonnet C. et al. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures // *Euro-Par 2009 Parallel Processing: 15th International Euro-Par Conference, Delft, The Netherlands, August 25–28, 2009. Proceedings 15*. Springer Berlin Heidelberg, 2009. P. 863–874. DOI: 10.1007/978-3-642-03869-3_80.

18. Ayguadé E. et al. An extension of the StarSs programming model for platforms with multiple GPUs // *Euro-Par 2009 Parallel Processing: 15th International Euro-Par Conference, Delft, The Netherlands, August 25–28, 2009. Proceedings 15*. Springer Berlin Heidelberg, 2009. P. 851–862. DOI: 10.1007/978-3-642-03869-3_79.

ОБЕСПЕЧЕНИЕ АВТОМАТИЧЕСКОГО КОНСТРУИРОВАНИЯ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ СПЕЦВЫЧИСЛИТЕЛЕЙ НА ОСНОВЕ КОНЦЕПЦИИ АКТИВНЫХ ЗНАНИЙ В СИСТЕМЕ LUNA

В. Э. Малышкин^{*,**,***}, В. А. Перепелкин^{*,**,***}, В. А. Спирин^{*,**}

^{*}Новосибирский национальный исследовательский государственный университет,
630090, Новосибирск, Россия

^{**}Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

^{***}Новосибирский государственный технический университет,
630073, Новосибирск, Россия

УДК 004.4'242

DOI: 10.24412/2073-0667-2025-4-73-88

EDN: ZCPRTC

Спецвычислители, такие как GPU или NPU, ориентированы на определенный характер вычислительной нагрузки, что позволяет в значительной степени повысить их производительность. Разработка эффективных программ, использующих спецвычислители, является трудоемкой задачей, так как требует знаний и навыков в области системного программирования. В частности, разработка таких программ затруднительна для специалистов других профилей. В работе рассматривается автоматическое конструирование параллельных программ с использованием спецвычислителей. Автоматизация направлена на снижение трудоемкости разработки и исполнения программ с получением приемлемой эффективности. Предлагаемое в работе решение основывается на концепции активных знаний и реализуется в системе активных знаний LuNA. Рассматривается расширение архитектуры исполнительной системы для поддержки спецвычислителей на примере представителя нейронных процессоров Huawei Ascend. Поддержка спецвычислителя обеспечивается отдельным слабосвязанным модулем. Приводятся результаты экспериментального исследования решения задач блочного умножения плотных матриц и корреляционной свертки сейсмограмм с использованием спецвычислителя.

Ключевые слова: концепция активных знаний, система LuNA, спецвычислитель, процессор Huawei Ascend, автоматическое конструирование программ, высокоуровневая спецификация, подсистема поддержки спецвычислителей.

Введение. Важную роль в современных высокопроизводительных вычислениях играют спецвычислители, такие как графические ускорители (GPU), нейронные процессоры (NPU), ПЛИС (FPGA). Спецвычислители позволяют существенно повысить производительность вычислений за счет ориентации на конкретный характер вычислительной нагрузки. В составе суперкомпьютеров именно спецвычислители, как правило, поставляют большую часть вычислительной мощности.

Работа выполнена в рамках государственного задания ИВМиМГ СО РАН FWNM-2025-0005.

Разработка эффективных параллельных программ, использующих спецвычислители, часто сопряжена с дополнительными проблемами по сравнению с использованием обычных процессоров (CPU). Тут и далее эффективность понимается с точки зрения времени выполнения программы, расхода памяти, нагрузки на сеть и т. п. характеристик. Эти проблемы проистекают из того, что эффективное использование спецвычислителя требует знания многих особенностей его работы и соответствующих технических средств, а также умения учитывать эти особенности при решении конкретной задачи. Существенно снизить сложность и трудоемкость разработки программ возможно за счет автоматизации этого процесса, когда часть работы перекладывается с человека на систему автоматизации. Также это позволяет, по крайней мере в перспективе, улучшить качество конструируемых программ по сравнению со средней ручной разработкой.

При всей сложности задачи автоматического конструирования эффективных параллельных программ, существуют подходы к ее решению, по крайней мере в частных случаях. Один из таких подходов — концепция активных знаний [1], определяющая методологию автоматического конструирования программ в конкретных предметных областях. В ее основе лежит идея построения базы активных знаний — машинно-ориентированного представления системы знаний о предметной области, обеспечивающей автоматическое конструирование достаточно эффективных для практического использования параллельных программ решения задач в этой предметной области. База активных знаний включает частичную аксиоматическую теорию предметной области, наработанные при ручном программировании в этой предметной области фрагменты кода и описание особенностей их применения для решения различных задач. Сильной стороной подхода является возможность автоматически учитывать специфичные особенности предметной области для обеспечения требуемой эффективности конструируемых программ.

Концепция активных знаний представляется перспективным подходом и для автоматизации конструирования параллельных программ, использующих спецвычислители, т. к. она позволяет переложить на систему и сложные рутинные технические детали использования спецвычислителя, и многие вопросы системного параллельного программирования. К таким вопросам относятся декомпозиция данных и вычислений, динамическое распределение нагрузки, синхронизация доступа к общим данным, совместное использование ядер центрального процессора и спецвычислителей и пр. Но реализация этой идеи требует развития как подхода, так и инструментария, представленного системой активных знаний LuNA [2, 3].

Целью работы являются исследование и частичная реализация возможностей концепции активных знаний по автоматизации использования специализированных вычислителей в конструируемых параллельных программах. В частности, рассматривается задача обеспечения поддержки спецвычислителей в системе LuNA на примере поддержки спецвычислителя Huawei Ascend [4].

Остальная часть статьи структурирована следующим образом. В разделе 1 приводится описание подхода к автоматическому использованию спецвычислителей на основе концепции активных знаний. В разделе 2 приводятся необходимые определения, связанные с системой LuNA, и описываются предлагаемые решения для поддержки спецвычислителей. В разделе 3+ представлены результаты экспериментального исследования для задач блочного умножения плотных матриц и корреляционной свертки сейсмотрасс с использованием спецвычислителя. В заключении подводятся итоги работы.

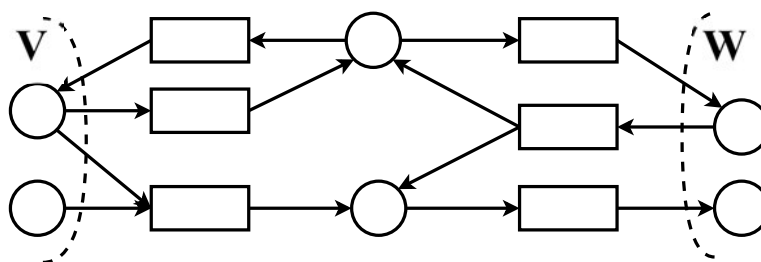


Рис. 1. Пример простой вычислительной модели, где круги представляют данные, а прямоугольники — операции; V и W представляют входные и выходные данные соответственно

1. Подход к автоматизации использования спецвычислителей. В концепции активных знаний автоматическое конструирование программ осуществляется на основе **вычислительных моделей** (ВМ). ВМ можно рассматривать как двудольный орграф, состоящий из *операций* и *переменных*, соединенных дугами (см. рис. 1.). Каждая переменная обозначает некоторую величину предметной области и может иметь значение произвольного типа. Каждая операция обозначает возможность вычислять значения своих выходных переменных (на исходящих дугах) из значений входных (на входящих дугах), и эта возможность реализуется путем предоставления **фрагмента кода** (ФК) — программного модуля (процедуры) определенного вида.

Далее, на ВМ ставится **VW-задача** — пара подмножеств переменных V и W, задающих входные и выходные переменные соответственно. По VW-задаче строится **VW-план** — подграф ВМ, который содержит достаточное количество операций и переменных, чтобы вычислить значения всех переменных из W из значений переменных из V (если такой план существует). По VW-плану строится программа, суть которой сводится к тому, чтобы считать извне значения входных переменных, вызвать в нужном порядке и с нужными аргументами ФК, вычисляя тем самым значения промежуточных и выходных переменных, после чего выдать вовне значения выходных переменных. Конструируемая программа может быть параллельной, если информационные зависимости операций это позволяют. Если конструируется программа для распределенной памяти, то система конструирования должна обеспечить передачу данных по сети тех значений переменных, которые были получены на одном узле мультимикрокомпьютера, а потребляются — на другом (других).

Базой активных знаний называется композиция из описания ВМ и набора фрагментов кода, описания нефункциональных свойств операций, переменных и ФК, а также необходимой сопутствующей технической информации. Таким образом, база активных знаний содержит все необходимое, чтобы на ее основе полностью автоматически конструировать программы. Собственно конструирование и исполнение программ осуществляет **система активных знаний** (система LuNA [2, 3] является примером такой системы). База активных знаний служит машинно-ориентированным способом представления знаний о предметной области, полученных, в первую очередь, в результате ручного программирования специалистов в этой предметной области. Именно знания, заложенные в базу активных знаний, позволяют конструировать программы удовлетворительного качества в этой предметной области.

В качестве примера удобно рассмотреть предметную область «треугольник». Переменными обозначаются такие величины как длины сторон, углы, площадь, периметр и т. п.,

а операции соответствуют формулам вычисления одних величин из других. Если каждую формулу реализовать в виде ФК, то это позволит автоматически конструировать программы решения задач вида «дано — требуется» в этой области. Задача может решаться в несколько действий на графе ВМ, как это показано выше.

Теперь рассмотрим вопрос поддержки спецвычислителей. Наивным способом поддержать возможность конструировать программы с использованием спецвычислителей является создание таких ФК, которые внутри на основе стандартного инструментария (CUDA, OpenACC и т. п.) осуществляют вычисления на спецвычислителе. В принципе, это будет работать, но у такого подхода есть ряд недостатков. Во-первых, вся техническая работа по использованию спецвычислителя остается на пользователе, так как именно он разрабатывает ФК. Во-вторых, система «не знает» о том, что в том или ином ФК используется спецвычислитель и потому не может учитывать это при создании VW-плана и программы (для планирования ресурсов, оценки эффективности и т. д.). В-третьих, отсутствует возможность оптимизировать перемещение данных между памятью центрального процессора и памятью спецвычислителя в случае, если на спецвычислителе выполняется более одной операции.

Нормальным способом поддержать спецвычислители является добавление в систему поддержки нового вида ФК, исполняемых на спецвычислителе. Тогда система конструирования может взять на себя заботы по инициализации и освобождению ресурсов вычислителя, передаче данных (значений переменных) в память спецвычислителя и обратно, отслеживать размещение значений переменных в памяти центрального процессора и памяти спецвычислителя и оптимизировать их перемещение, планировать распределение вычислительной нагрузки по разным устройствам, выполнять синхронизацию процессов и т. п. Такая автоматизация возможна благодаря тому, что в основе конструирования лежит ВМ, явно описывающая переменные, операции и ФК.

В такой постановке задачи разрабатывать код на спецвычислителе все еще должен пользователь. Но в данном случае пользователь должен разработать только т. н. «ядро» (kernel) — только вычисления, без необходимости синхронизации процессов или передачи данных между памятью компьютера. К тому же, разработка одного модуля для спецвычислителя — это существенно проще, чем его встраивание в обычную программу.

При этом концепция активных знаний обладает преимуществами, которые распространяются и на программы со спецвычислителями. Во-первых, это возможность автоматически обеспечивать динамическую балансировку нагрузки на устройства (ядра центрального процессора и спецвычислитель). В работах [5, 6] показывалось, как на основе концепции активных знаний обеспечивается динамическая балансировка нагрузки на узлы мультимикрокомпьютера. Этот же подход может быть использован и для спецвычислителя. Во-вторых, это дополнительные возможности отладки. Например, пользователь может обратиться к стороннему специалисту или программным инструментам для создания ФК для спецвычислителя из обычного ФК. Тогда в режиме отладки возможно автоматически сравнивать результаты работы обычного и специализированного ФК. В-третьих, это оптимизация программ на основе профилирования. Например, возможно автоматически собирать информацию о временных и прочих характеристиках исполнения ФК, после чего повторно конструировать программу с более эффективным распределением вычислений на центральных процессорах и спецвычислителях. Также в систему возможно встраивать поддержку различных продвинутых техник оптимизации вычислений (асинхронные передачи данных, техники типа CUDA Graph и т. п.) без необходимости пользователю

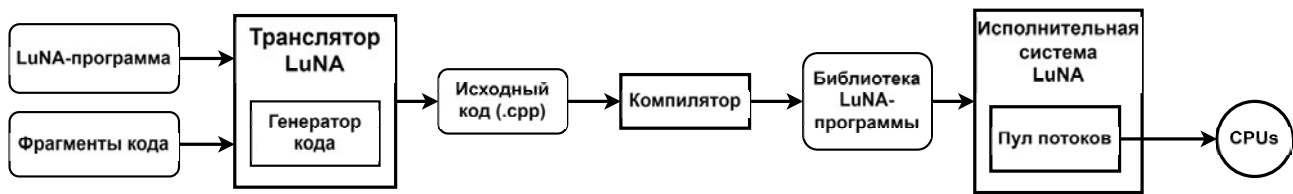


Рис. 2. Схема обработки LuNA-программы системой

даже знать о таких техниках. Эта возможность обеспечивается благодаря тому, что в базе активных знаний содержится достаточно информации для того, чтобы применять эти техники.

2. Система LuNA и поддержка спецвычислителей. Кратко рассмотрим вопрос реализации поддержки спецвычислителей в системе LuNA. Описание ВМ на языке LuNA имеет следующий вид (листинг 1).

Листинг 1. Пример LuNA-программы.

```

01: import init_matrix(name, int);
02: import mat_mat_mul(value, value, name, int)
    : DEV_ID @ {annotations};
03: import print(value, int);
04:
05: sub main(int matSize) {
06:     df matA, matB, matC;
07:     init_matrix(matA, matSize);
08:     init_matrix(matB, matSize);
09:     mat_mat_mul(matA, matB, matC, matSize)
        @ {annotations};
10:     print(matC, matSize);
11: }
  
```

На строках 1–3 объявляются ФК, которые определяются в других файлах в виде C++-процедур. На строках 7–10 определяются операции. Также в строке 9 определяются аннотации (после символа @), которые позволяют задавать нефункциональные параметры операций. Дополнительные параметры, указанные на строке 2, будут рассмотрены позже.

Файл с описанием LuNA-программы вместе с файлами с определениями ФК обрабатываются разными компонентами системы LuNA так, как это показано на рис. 2.

Поданные на вход LuNA-программа и ФК сначала обрабатываются транслятором (ключевым компонентом которого является *генератор кода*), затем компилятором языка программирования C++. В результате генерируется исполняемое представление программы в виде машинного кода, которое содержит фрагменты кода и другие процедуры, которые позволяют управлять исполнением программы.

Далее исполнительная система загружает и исполняет такое представление с использованием *пула потоков*.

В работе в качестве частного случая спецвычислителя рассматривается процессор Huawei Ascend как представитель современных нейронных процессоров. Далее в подразделе под *спецвычислителем (CB)* подразумевается процессор Huawei Ascend.

Принцип работы с СВ. Основным понятием, описывающим вычисления для СВ, является *оператор*. Операторы описываются отдельно от прикладной программы в специальном формате [7]. На основе этого формата генерируется исполняемое представление оператора.

Система Ascend¹ поставляется со *встроенными операторами*, например, оператором GEMM (GEneral Matrix Multiplication) для умножения плотных матриц. Операторы, которые пользователи описывают сами в специальном формате, далее будем называть *собственными*.

Для взаимодействия с СВ в прикладных программах используется библиотека AscendCL [8], которая позволяет управлять памятью СВ, запускать операторы на СВ и пр.

Есть дополнительные нюансы, связанные с использованием СВ в прикладных программах. В частности, для работы со СВ требуется, чтобы поток исполнения был «привязан» к нужному СВ. Это позволяет связать вызовы процедур библиотеки AscendCL в потоке с нужным СВ (т. е. вызовы процедур по управлению памятью, запуску оператора, и пр. будут выполнены на связанном СВ).

За счет автоматизации с пользователя снимаются требования, связанные с знанием такой специфики использования СВ.

Поддержка СВ в системе LuNA. Для поддержки СВ требуется расширить возможности системы LuNA так, чтобы система могла работать со СВ без дополнительного вмешательства пользователя (инициализировать и освобождать ресурсы СВ, осуществлять передачу данных и балансировку вычислительной нагрузки между ядрами центрального процессора и СВ). Для этого был введен новый тип ФК, который описывает вычисления для СВ с соответствующим расширением синтаксиса (см. строку 2 листинга 1). Для автоматизации использования СВ был разработан алгоритм управления очередями задач и потоками в пуле (см. рис. 3). Архитектура системы была расширена (рис. 4), что позволило добавлять поддержку новых видов спецвычислителей путем добавления нового слабосвязанного модуля.

Автоматическая генерация вычислений для СВ. Одна из основных сложностей работы со СВ происходит из процесса разработки ФК для СВ. ФК должен соответствовать требованиям, которые установлены производителями СВ. При этом если требуется использовать другой спецвычислитель, то требования к формату ФК для него будут другими.

В работе предлагается формат описания вычислений CoFaNA (Code Fragment Notation for Accelerator), который позволяет описывать вычисления для СВ на высоком уровне абстракции. На основе этого формата генерируется ФК для СВ в нужном формате. Также формат CoFaNA подходит для генерации ФК для других видов спецвычислителей.

Структура формата CoFaNA представлена на листинге 2 ниже.

Листинг 2. Структура формата CoFaNA.

```
01: name OperatorName
02: types t_floats float16 float double
03: types ts_ints int
04: localvar input t_floats X0local [-1] size=sX0local
```

¹Под системой Ascend подразумевается некоторое программное обеспечение, которое позволяет регулировать работу процессоров Ascend.

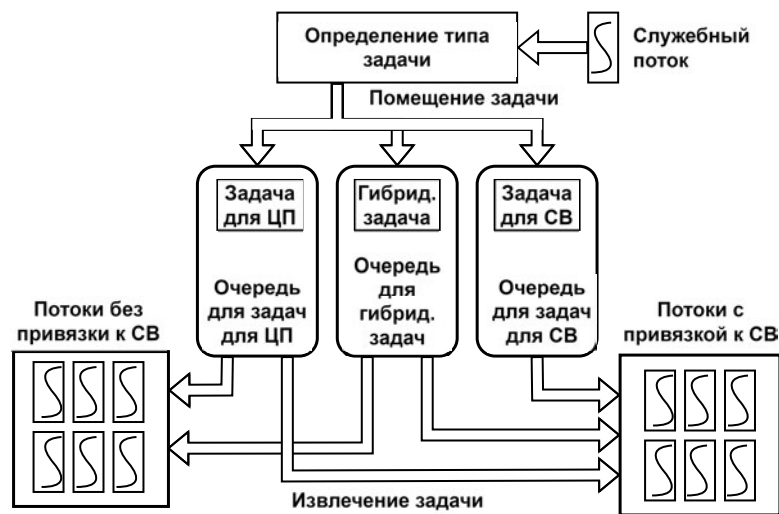


Рис. 3. Схема распределения задач с использованием трех очередей для каждого типа задач

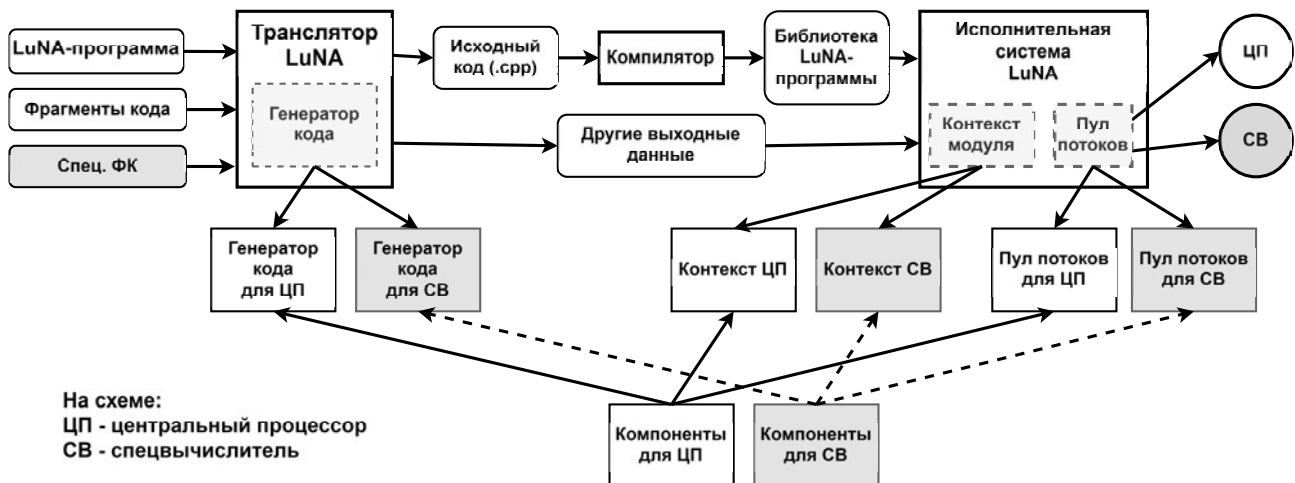


Рис. 4. Схема расширения архитектуры системы LuNA для поддержки спецвычислителей

```

05: opvar input t_floats X0 [-1] size=sX0
06: opvar input t_ints M0 [1] size=sM0
07: opvar output t_floats X implicit [sX0[0]] size=sX
08: opvar output t_floats X1 implicit [sX0[0]] size=sX1
09: rule float float int > float float
10:
11: __header__
12: ...
13: __end__
14:
15: __before__
16: ...
17: __end__
18:

```

```

19: __operator__
20:     __cpu__
21:     ...
22:     __end__
23:     __npu__
24:     ...
25:     __end__
26: __end__
27:
28: __operator__ BuiltinOperatorName
29: var input X implicit [sX0[0] / 2]
30: var input M0 explicit [1]
31: var output X1 implicit [sX0[0]]
32: rule float int float > complex64 int float
33: __start__
34:     ...
35: __end__
36:
37: __after__
38:     ...
39: __end__

```

В строках 1–9 расположена секция с метайнформацией. В строках 11–39 расположена секция с вычислениями. В метайнформации описываются такие компоненты, как название оператора, параметры и их допустимые типы, а также список комбинаций типов, которые будут использоваться в программе. Секция с вычислениями содержит четыре типа вычислений: вычисления перед запуском операторов на СВ, вычисления для собственного оператора, использование встроенных операторов и вычисления после запуска всех операторов на СВ.

Представленные результаты автоматизации использования СВ были частично опубликованы в работе [9].

3. Тестирование программ с использованием процессора Huawei Ascend. Тестирование программ, использующих процессор Ascend, проводилось на задачах блочного умножения плотных матриц и корреляционной свертки сейсмотрасс [10]. Целью тестирования выступает сравнение ручных C++-реализаций и LuNA-реализаций, а также сравнение использования центральных процессоров и процессоров Ascend для решения задач на предмет оценки качества при применении концепции активных знаний. Основной метрикой сравнения выступает время работы программ.

Запуски были выполнены на сервере Atlas 800 (модель 9000), установленном в ИВ-МиМГ СО РАН (см. описание на странице лаборатории ИИИТ [11]).

Задача блочного умножения плотных матриц. Цель тестирования этой задачи заключается в сравнении ручной реализации на языке C++ и автоматически конструируемых реализаций с использованием системы LuNA. Также сравнивается использование разного количества процессоров Ascend.

Для этой задачи матрицы разбиваются на блоки, и умножение двух блоков рассматривается как отдельная планируемая задача. В качестве сравнения выступают три реализации: 1) реализация на C++ только с вычислениями на процессоре Ascend (C++-

NPU реализация); 2) реализация на LuNA только с вычислениями на процессоре Ascend (LuNA-NPU реализация); 3) реализация на LuNA только с вычислениями на центральном процессоре (LuNA-CPU реализация).

Задача корреляционной свертки сейсмотрасс. Цель тестирования этой задачи заключается в применении концепции активных знаний для реальной прикладной задачи. Эта задача, как описано в [10], выполняет вычисление взаимокорреляционной функции и быстрого преобразования Фурье (FFT). В качестве сравнения выступают три LuNA-реализации: 1) С использованием вычислений на центральном процессоре (при этом, эта реализация работает сопоставимо с ручной реализацией на C++). 2) С использованием собственного оператора процессора Ascend, в который были добавлены вычисления из предыдущего способа. 3) С использованием встроенных операторов процессора Ascend для вычисления FFT.

Результаты тестирования. На рис. 5, а, видно, что на больших размерах матриц использование процессора Ascend позволяет снизить время выполнения программы по сравнению с использованием центрального процессора.

На рис. 5, б, показано, что на больших размерах матриц использование большего количества процессоров Ascend позволяет снизить время выполнения программ в большей степени.

На рис. 5, в, показано сравнение реализаций C++-NPU и LuNA-NPU при использовании 4-х процессоров Ascend. По графику видно, что на больших размерах матриц ручная C++-реализация работает сопоставимо с LuNA-реализацией с точки зрения времени выполнения.

На рис. 5, г, видно, вопреки ожиданиям авторов, что вычисление FFT на процессоре Ascend значительно уступает вычислению FFT на центральном процессоре. При этом использование встроенных операторов для FFT уступает использованию собственного оператора.

В целом можно отметить, что разработка LuNA-программ, использующих процессор Ascend, требует меньше трудозатрат, чем разработка ручных реализаций, при этом эти реализации работают сопоставимо. В большей степени этому способствует разработанный формат CoFaNA.

Заключение. В работе рассмотрен подход автоматического использования спецвычислителей на основе концепции активных знаний в системе LuNA. Было выполнено расширение архитектуры системы LuNA для поддержки различных спецвычислителей.

Реализована поддержка спецвычислителя Huawei Ascend в системе LuNA. Для упрощения работы с вычислениями на процессоре Huawei Ascend были предложены формат CoFaNA и пул потоков, поддерживающий такие вычисления.

На этом примере было показано, что на основе концепции активных знаний может быть обеспечено автоматическое конструирование программ с использованием спецвычислителей. При этом с пользователя снимается существенное количество работы, связанной с применением спецвычислителей и требующей детального знания их работы. Это существенно упрощает применение спецвычислителей.

Результаты экспериментального исследования подтвердили, что использование процессора Huawei Ascend при решении задачи блочного умножения плотных матриц позволяет значительно повысить эффективность по сравнению с использованием только CPU. Основным результатом тестирования является то, что программы, построенные автоматически, по эффективности оказались близки к низкоуровневым ручным реализациям на

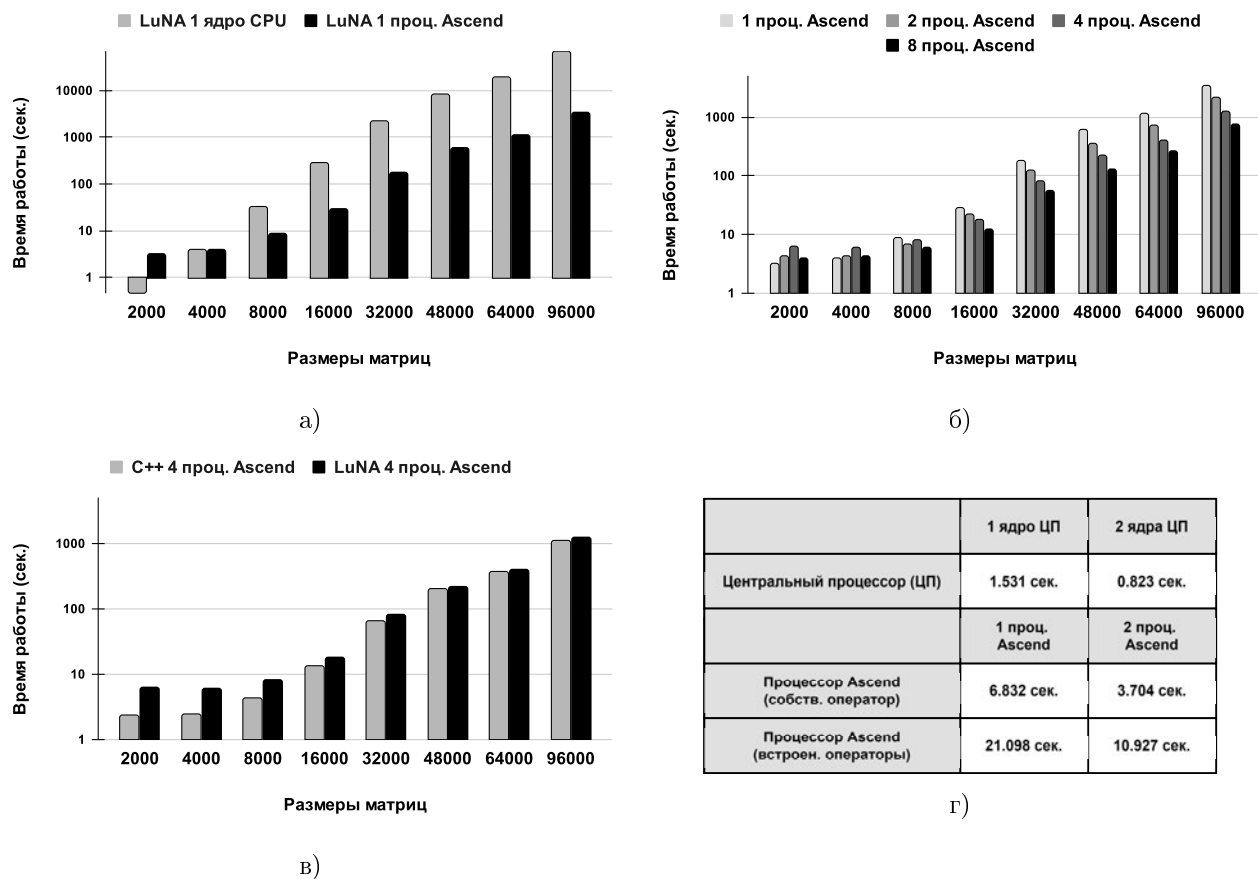


Рис. 5. (а)–(в) — зависимость времени работы программ (логарифмическая шкала) от размеров матриц, (а) — LuNA-CPU на 1 ядре центр. проц. и LuNA-NPU на 1 проц. Ascend, (б) — LuNA-NPU на разном количестве проц. Ascend, (в) — C++-NPU и LuNA-NPU на 4-х проц. Ascend; (г) — зависимость времени работы алгоритма свертки от количества ресурсов и типа вычислений

C++. Это подтверждает применимость концепции активных знаний для решения таких задач автоматического конструирования программ.

Рассмотренный в работе подход на основе активных знаний является не единственным способом реализации поддержки спецвычислителей. Эта проблема исследуется и решается в разной степени в разных системах и компиляторах. В системе Mindspore [12], в которой все программы реализуются на языке Python, реализуется поддержка рассмотренного процессора Huawei Ascend для решения задач машинного обучения. Также поддержку Huawei Ascend реализует компилятор DPC++ [13]. В [14] приводится сравнение различных систем автоматического конструирования параллельных программ, обеспечивающих поддержку графических ускорителей (GPU): Charm++ [15], Legion [16], StarPU [17], StarSS [18].

В отличие от концепции активных знаний, которая позволяет автоматически конструировать эффективные программы в разных предметных областях (за счет разработки разных баз активных знаний), в представленных выше системах упор идет на автоматизацию конструирования эффективных программ ограниченного числа предметных областей (которые определяют область применения системы). Отдельно здесь стоит отметить систему Legion, которая позволяет эффективно работать с разными предметными областями, но с

требованием от пользователя проделывать некоторые этапы конструирования программы вручную.

В дальнейшем планируется реализовывать поддержку других спецвычислителей и при необходимости расширять подсистему поддержки спецвычислителей.

Список литературы

1. Malyshkin V. E. Active Knowledge, LuNA and Literacy for Oncoming Centuries // LNCS, T. 9465. 2015. P. 292–303. DOI: 10.1007/978-3-319-25527-9_19.
2. Malyshkin V. E., Perepelkin V. A. LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem // Proceedings of the 11th International Conference on Parallel Computing Technologies (PaCT-2011), LNCS, 2011. T. 6873, P. 53–61. DOI: 10.1007/978-3-642-23178-0_5.
3. Малышкин В. Э., Перепелкин В. А. Построение баз активных знаний для автоматического конструирования решений прикладных задач на основе системы LuNA // Параллельные вычислительные технологии — XVIII всероссийская научная конференция с международным участием, ПаВТ'2024, г. Челябинск, 2–4 апреля 2024 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2024. С. 57–68. DOI: 10.14529/pct2024.
4. Liang X. Ascend AI Processor Architecture and Programming: Principles and Applications of CANN. Elsevier, 2020. DOI: 10.1016/C2020-0-00270-7.
5. Malyshkin V., Perepelkin V., Schukin G. Scalable Distributed Data Allocation in LuNA Fragmented Programming System // The Journal of Supercomputing, 2017. T. 73, Iss. 2, Springer. P. 726–732. DOI: 10.1007/s11227-016-1781-0.
6. Малышкин В. Э., Перепелкин В. А., Шукин Г. А. Распределенный алгоритм управления данными в системе фрагментированного программирования LuNA // Проблемы информатики, 2017. № 1(34). С. 74–88.
7. Using Operator Samples [Electron. Res.]: https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/82RC1alpha001/opdevg/tbeaicpudev/atlasopdev_10_0025.html.
8. AscendCL architecture and basic concepts [Electron. Res.]: https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/82RC1alpha001/appdevg/aclcppdevg/aclcppdevg_000004.html.
9. Спирин В. А. Разработка алгоритмов автоматизации применения NPU в системе LuNA // Параллельные вычислительные технологии — XVIII всероссийская научная конференция с международным участием, ПаВТ'2024, г. Челябинск, 2–4 апреля 2024 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2024. С. 165–176. DOI: 10.14529/pct2024.
10. Хайретдинов М. С. Алгоритмы поточной свертки в задачах активного виброемкоакустического мониторинга / М. С. Хайретдинов, Г. М. Воскобойникова, Г. С. Седухина // Геосибирь, 2017.
11. Лаборатория искусственного интеллекта и информационных технологий [Электрон. Рес.]: <https://icmmg.nsc.ru/ru/content/pages/laboratoriya-iskusstvennogo-intellekta-i-informacionnyh-tehnologiy>.
12. Chen L. Deep learning and practice with mindspore. Springer Nature, 2021. DOI: 10.1007/978-981-16-2233-5.
13. Feng W., Maghareh R., Wang K. T. A. Extending DPC++ with Support for Huawei Ascend AI Chipset // International Workshop on OpenCL. 2021. P. 1–4. DOI: 10.1145/3456669.3456684.
14. Gu R., Becchi M. A comparative study of parallel programming frameworks for distributed GPU applications // Proceedings of the 16th ACM International Conference on Computing Frontiers. 2019. P. 268–273. DOI: 10.1145/3310273.3323071.

15. Robson M. P., Buch R., Kale L. V. Runtime coordinated heterogeneous tasks in Charm++ // 2016 Second International Workshop on Extreme Scale Programming Models and Middlewar (ESPM2). IEEE, 2016. P. 40–43. DOI: 10.1109/ESPM2.2016.011.
16. Bauer M. et al. Legion: Expressing locality and independence with logical regions // SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE, 2012. P. 1–11. DOI: 10.1109/SC.2012.71.
17. Augonnet C. et al. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures // Euro-Par 2009 Parallel Processing: 15th International Euro-Par Conference, Delft, The Netherlands, August 25–28, 2009. Proceedings 15. Springer Berlin Heidelberg, 2009. P. 863–874. DOI: 10.1007/978-3-642-03869-3_80.
18. Ayguadé E. et al. An extension of the StarSs programming model for platforms with multiple GPUs // Euro-Par 2009 Parallel Processing: 15th International Euro-Par Conference, Delft, The Netherlands, August 25–28, 2009. Proceedings 15. Springer Berlin Heidelberg, 2009. P. 851–862. DOI: 10.1007/978-3-642-03869-3_79.



Виктор Эммануилович

Малышкин — получил степень магистра математики в Томском государственном университете (1970), степень доктора технических наук в Новосибирском государственном

университете (1993). В настоящее время является заведующим лабораторией синтеза параллельных программ в Институте вычислительной математики и математической геофизики СО РАН. Он также основал и в настоящее время возглавляет кафедру параллельных вычислений в Национальном исследовательском университете Новосибирска. Является одним из организаторов международной конференции PaCT (Parallel Computing Technologies), проводимых каждый нечетный год в России.

Имеет более 110 публикаций по параллельным и распределенным вычислениям, синтезу параллельных программ, суперкомпьютерному программному обеспечению и приложениям, параллельной реализации крупномасштабных численных моделей.

В настоящее время область его интересов включает параллельные вычислительные технологии, языки и системы параллельного программирования, методы и средства параллельной реализации крупномасштабных численных моделей, технологию активных знаний.

Victor Emmanuilovich Malyshev graduated from Tomsk State University with the master of sciences degree in 1970, defended his candidate dissertation in physical and

mathematical sciences in Computing Centre of SB RAS in 1984, and defended his doctoral dissertation in technical sciences in Novosibirsk State University (1993). Currently is head of parallel programs synthesis laboratory in the Institute of computational mathematics and mathematical geophysics SB RAS.

Is also a founder and head of the chair of parallel computing in Novosibirsk State University. Is one of organizers of the PaCT (Parallel Computing Technologies) international conference, being held each odd year in Russia. Has more than 110 publications on parallel and distributed computing, parallel programs synthesis, supercomputing software and applications, parallel implementation of large-scale numerical models. Current scientific interests include parallel computing technologies, languages and systems of parallel computing, methods and tools of software implementation of large-scale numerical models, the active knowledge technology.



Перепелкин Владислав
Александрович — кандидат

технических наук, старший научный сотрудник Института вычислительной математики и математической геофизики СО РАН; доцент кафедры парал-

лельных вычислений факультета информационных технологий Новосибирского государственного университета. Тел.: (383) 330-89-94, e-mail: perepelkin@ssd.sscs.ru. В 2008 году

окончил Новосибирский государственный университет с присуждением степени магистра техники и технологии по направлению «Информатика и вычислительная техника». В 2023 году защитил кандидатскую диссертацию. Имеет более 20 опубликованных статей по теме автоматизации конструирования параллельных программ в области численного моделирования. Является одним из основных разработчиков экспериментальной системы автоматизации конструирования численных параллельных программ для мультикомпьютеров LuNA (от Language for Numerical Algorithms). Область профессиональных интересов включает автоматизацию конструирования параллельных программ, языки и системы параллельного программирования, высокопроизводительные вычисления.

Perepelkin Vladislav Aleksandrovich — PhD in Computer Science. Graduated from Novosibirsk State University in 2008 with the master degree in engineering and technology in computer science. Defended his PhD thesis in 2023. Nowadays has the senior researcher position at the Institute of Computational Mathematics and Mathematical Geophysics of the Siberian Branch of Russian Academy of Sciences, and also has a part time associated professor position in the Novosibirsk State University. He is an author of more than 20 papers on automation of numerical parallel programs construction. He is one of

main developers of system LuNA (Language for Numerical Algorithms) for automatic construction of numerical parallel programs. Professional interests include automation of parallel programs construction, languages and systems of parallel programming, high performance computing.



Спирин Виталий Андреевич — студент магистратуры факультета информационных технологий Новосибирского государственного университета и инженер в Институте вычислительной математики и

математической геофизики СО РАН. Получил диплом бакалавра по направлению «Информатика и вычислительная техника» в 2023 году. E-mail: v.spirin@ng.su.ru. Область научных интересов: параллельные алгоритмы, высокопроизводительные вычисления, автоматическое конструирование программ.

Spirin Vitaly Andreevich is a master's student at the Faculty of Information Technology at Novosibirsk State University and an engineer at the Institute of Computational Mathematics and Mathematical Geophysics of the Siberian Branch of Russian Academy of Sciences. Graduated with a bachelor's degree in Computer Science in 2023. E-mail: v.spirin@ng.su.ru. Research interests: parallel algorithms, high performance computing, automatic programs construction.

Дата поступления — 29.05.2025

Алфавитный указатель авторов за 2025 год

АВРАМЕНКО Ю. А. — см. ВИШНЕВСКИЙ В. М.

АЛЕЕВА В. Н. Подход к эффективной реализации численных алгоритмов — № 1 (66).

АЛЕЕВА В. Н., САПОЖНИКОВ А. С. Эффективная реализация алгоритмов обучения нейронных сетей с помощью концепции Q-детерминанта — № 3 (68).

АХАТОВ А. Р., ЭШТЕМИРОВ Б. Ш., НАЗАРОВ Ф. М. Методы определения и анализа индекса движения на городских дорогах — № 1 (66).

БАЙБАРА Б. В. — см. РАХМАНИ Д.

БАКУЛИНА М. П. Эффективный алгоритм сжатия с помощью преобразования данных словарного типа — № 4 (69).

БАРКАЛОВ К. А. — см. УСОВА М. А.

БОБОХОНОВ А., ХУРАМОВ Л., РАШИДОВ А. Detection of Skin Diseases from Images using Machine Learning and Deep Learning Techniques — № 3 (68).

БРЕДИХИН С. В., ЩЕРБАКОВА Н. Г. Выявление сообществ в мультиплексной сети авторов научного журнала — № 4 (69).

БРЕДИХИН С. В., ЩЕРБАКОВА Н. Г. Модель сети соавторства научного журнала — № 1 (66).

БРЕДИХИН С. В., ЩЕРБАКОВА Н. Г. Структурные свойства мультиплексной сети авторов научного журнала — № 2 (67).

ВИШНЕВСКИЙ В. М., АВРАМЕНКО Ю. А., НГУЕН В. Х., КАЛМЫКОВ Н. С. Оценка характеристик производительности беспроводной сети на базе привязных БПЛА — № 1 (66).

ГЕРЬ А. Р., ДЕВЯТЫХ Е. Е., ОМАРОВА Г. А. Сравнение методов графовой редукции первого, второго и третьего поколения в моделях химической кинетики — № 4 (69).

ДЕВЯТЫХ Е. Е. — см. ГЕРЬ А. Р.

ДЕЛЬ И. В., СТАРЧЕНКО А. В. Краткосрочный прогноз скорости ветра на основе искусственных нейронных сетей с применением метода декомпозиции на вариационные моды — № 2 (67).

ИВЛЕВ А. Д., ЛИНЕВ А. В. Реализация алгоритмов подготовки начального состояния системы кубитов с учетом ограничений современных квантовых компьютеров — № 2 (67).

КАЛМЫКОВ Н. С. — см. ВИШНЕВСКИЙ В. М.

ЛЕБЕДЕВ И. Г. — см. УСОВА М. А.

ЛЕОНОВА Ю. Ф. Оптимизация программного кода на примере алгоритма для решения задачи коммивояжера — № 2 (67).

ЛИНЕВ А. В. — см. ИВЛЕВ А. Д.

МАЛЫШКИН В. Э., ПЕРЕПЕЛКИН В. А., НУШТАЕВ Ю. Ю. Уменьшение накладных расходов на вызов модулей в автоматически конструируемых программах на основе концепции активных знаний — № 3 (68).

МАЛЫШКИН В. Э., ПЕРЕПЕЛКИН В. А., СПИРИН. Обеспечение автоматического конструирования программ с использованием спецвычислителей на основе концепции активных знаний в системе LuNA — № 4 (69).

МАТКУРБАНОВ Т. А. — см. РОДИОНОВ А. С.

МОНАХОВ О. Г. — см. МОНАХОВА Э. А.

МОНАХОВА Э. А., МОНАХОВ О. Г. Генерация многоуровневых регулярных сетей на основе операции композиции модифицированных хордальных графов с использованием больших языковых моделей — № 4 (69).

НАЗАРОВ Ф. М. — см. АХАТОВ А. Р.

НГУЕН В. Х. — см. ВИШНЕВСКИЙ В. М.

НУШТАЕВ Ю. Ю. — см. МАЛЫШКИН В. Э.

ОМАРОВА Г. А. — см. ГЕРБ А. Р.

ПЕРЕПЕЛКИН В. А. — см. МАЛЫШКИН В. Э.

РАШИДОВ А. — см. БОБОХОНОВ А., ТУРСУНОВ Ш. А.

РАХМАНИ Д., БАЙБАРА Б. В., ТЕТОВ С. Г. Уязвимости больших языковых моделей: анализ и методы защиты — № 3 (68).

РОДИОНОВ А. С., МАТКУРБАНОВ Т. А., ХАЙРУЛЛАЕВ У. Б. Построение траектории полета БПЛА для мониторинга сельскохозяйственных земель — № 1 (66).

САПОЖНИКОВ А. С. — см. АЛЕЕВА В. Н.

СПИРИН В. А. — см. МАЛЫШКИН В. Э.

СТАРЧЕНКО А. В. — см. ДЕЛЬ И. В.

ТЕТОВ С. Г. — см. РАХМАНИ Д.

ТУРСУНОВ Ш. А., РАШИДОВ А. Э. Анализ алгоритмов кодирования категориальных данных — № 2 (67).

УСОВА М. А., ЛЕБЕДЕВ И. Г., ШТАНЮК А. А., БАРКАЛОВ К. А. Алгоритм глобальной оптимизации для настройки гиперпараметров методов машинного обучения — № 4 (69).

ХАЙРУЛЛАЕВ У. Б. — см. РОДИОНОВ А. С.

ХУРАМОВ Л. — см. БОБОХОНОВ А.

ЩЕРБАКОВА Н. Г. — см. БРЕДИХИН С. В.

ШТАНЮК А. А. — см. УСОВА М. А.

ЭШТЕМИРОВ Б. Ш. — см. АХАТОВ А. Р.

ЮРТИН А. А. Метод прогнозирования ошибки времени обучения нейросетевых моделей восстановления многомерных временных рядов — № 3 (68).

Правила представления и подготовки рукописей для публикации в журнале „ПРОБЛЕМЫ ИНФОРМАТИКИ“

Общие требования.

Редакция принимает к рассмотрению статьи в электронном виде (*исходный файл в L^AT_EX* и файл PDF, с приложением оригиналов рисунков в формате тех программ, в которых они были сделаны, отдельными файлами).

Файлы, содержащие текст статьи, иллюстрации и дополнительные материалы, можно пересылать на электронный адрес редакции: problem-info@sscc.ru.

Принимаются файлы, архивированные архиваторами ZIP/7Z или RAR; применение самораспаковывающихся архивов не допускается.

При повторной отправки материалов, а также при внесении в исходный текст дополнений или исправлений необходимо сообщить об этом в редакцию в тексте электронного письма.

Направляя статью в редакцию журнала, автор (соавторы) на безвозмездной основе передает(ют) издателю на срок действия авторского права по действующему законодательству РФ исключительное право на использование статьи или отдельной ее части (в случае принятия редколлегией Журнала статьи к опубликованию) на территории всех государств, где авторские права в силу международных договоров Российской Федерации являются охраняемыми, в том числе следующие права: на воспроизведение, на распространение, на публичный показ, на доведение до всеобщего сведения, на перевод на иностранные языки и переработку (и исключительное право на использование переведенного и (или) переработанного произведения вышеуказанными способами), на предоставление всех вышеперечисленных прав другим лицам.

Журнал „Проблемы информатики“ является некоммерческим изданием. Плата с авторов за публикацию статей не взимается.

К статье должны быть приложены:

— **разрешение на публикацию** от экспертного совета организации, в которой выполнена работа (для авторов из России);

— **оригинал рецензии;**

портретные фотографии авторов разрешением не менее 300 dpi.

— Блоки информации и на русском, и на английском языках просьба присылать отдельными файлами:

— **Название** статьи;

— **Инициалы и фамилии** авторов;

— **Места работы** авторов: полное наименование организации, почтовый индекс, город, страна;

— **Код(ы) классификации УДК;**

— **Аннотации**, содержащие краткую постановку задачи и описание метода решения: на русском языке объемом не более 1000 знаков, на английском языке расширенную, объемом от 4000 до 8000 знаков, что соответствует требованиям ВАК и Scopus.

— **Ключевые слова;**

— **Списки используемой литературы** в соответствии с ГОСТ Р 7.0.5—2008 (в английской версии необходимо выполнить **транслитерацию** неанглоязычных элементов списка литературы в соответствии с ГОСТ Р 7.0.34-2014) — составляются по ходу упоминания источников в тексте;

— **Краткие биографии (БИО)** авторов с указанием ключевых научных достижений (включая ученую степень, ученое звание — при наличии; основные области научных интересов и формулировку основных результатов, место работы, занимаемую должность, контактные данные — почтовый адрес с индексом, адрес электронной почты, контактный телефон).

Подготовка статьи.

1. Материал статьи должен быть изложен в следующей последовательности:

1.1. название статьи на английском языке;

1.2. инициалы и фамилия автора(ов) на английском языке;

1.3. место работы автора(ов) (на английском языке): полное наименование организации, индекс, город, страна;

1.4. англоязычная аннотация;

1.5. ключевые слова на английском языке;

1.6. references+транслитерация неанглоязычных элементов списка литературы;

1.7. название статьи на русском языке;

1.8. инициалы и фамилии и авторов;

- 1.9. место работы авторов: полное наименование организации, почтовый индекс, город, страна;
- 1.10. индекс УДК;
- 1.11. аннотация на русском языке;
- 1.12. ключевые слова (не более 8);
- 1.13. текст статьи;
- 1.14. список литературы, оформленный в соответствии с требованиями ГОСТ;
- 1.15. краткие биографии авторов на английском и русском языках с указанием ключевых научных достижений (ученую степень, ученое звание — при наличии; место работы, занимаемую должность, контактные данные — почтовый адрес с индексом, адрес электронной почты, контактный телефон, основные области научных интересов и формулировка основных результатов).

2. Требования к формулам:

- Нумерация формул сквозная, выносные формулы центрируются, номер выровнен по правому краю.

3. Требования к рисункам:

- Файлы с рисунками присылаются отдельно в формате программ, в которых они были выполнены: в формате MS Excel (для графиков и диаграмм), eps, pdf, png, tiff, bmp или jpeg (с максимальным качеством).
- Рисунки с подрисуночными подписями завершаются в текст статьи.
- Тексты, являющиеся частью рисунка, выполняются шрифтом TimesNewRoman.
- Фотографии должны иметь разрешение не менее 300 dpi.

4. Дополнительные требования:

- В текст статьи необходимо включать ссылки на рисунки и таблицы, а также подрисуночные подписи и заголовки таблиц. Все буквенные обозначения, приведенные на рисунках, необходимо пояснить в основном тексте или в подрисуночных подписях.
- Сокращения слов не допускаются (кроме общепринятых).
- Векторные переменные обозначаются полужирным шрифтом без курсива.
- Таблицы не должны быть громоздкими. Значения физических величин в таблицах, на графиках и в тексте должны указываться в единицах измерения СИ.
- Графики, если их на рисунке несколько, а также отдельные детали на чертежах, узлы и линии на схемах следует обозначать цифрами, набранными курсивом.
- Нумеровать следует только те формулы и уравнения, на которые имеются ссылки в тексте, нумерация сквозная.
- Ссылки на источники в тексте заключаются в квадратные скобки.
- Иностранные источники приводятся на языке оригинала. Ссылки на неопубликованные работы не допускаются.

Все статьи, опубликованные в журнале «Проблемы информатики», доступны на сайте https://elibrary.ru/title_about.asp?id=30275 и на сайте журнала <http://problem-info.sssc.ru> спустя год после опубликования.

Пример оформления статей можно посмотреть на сайте журнала <http://problem-info.sssc.ru>.