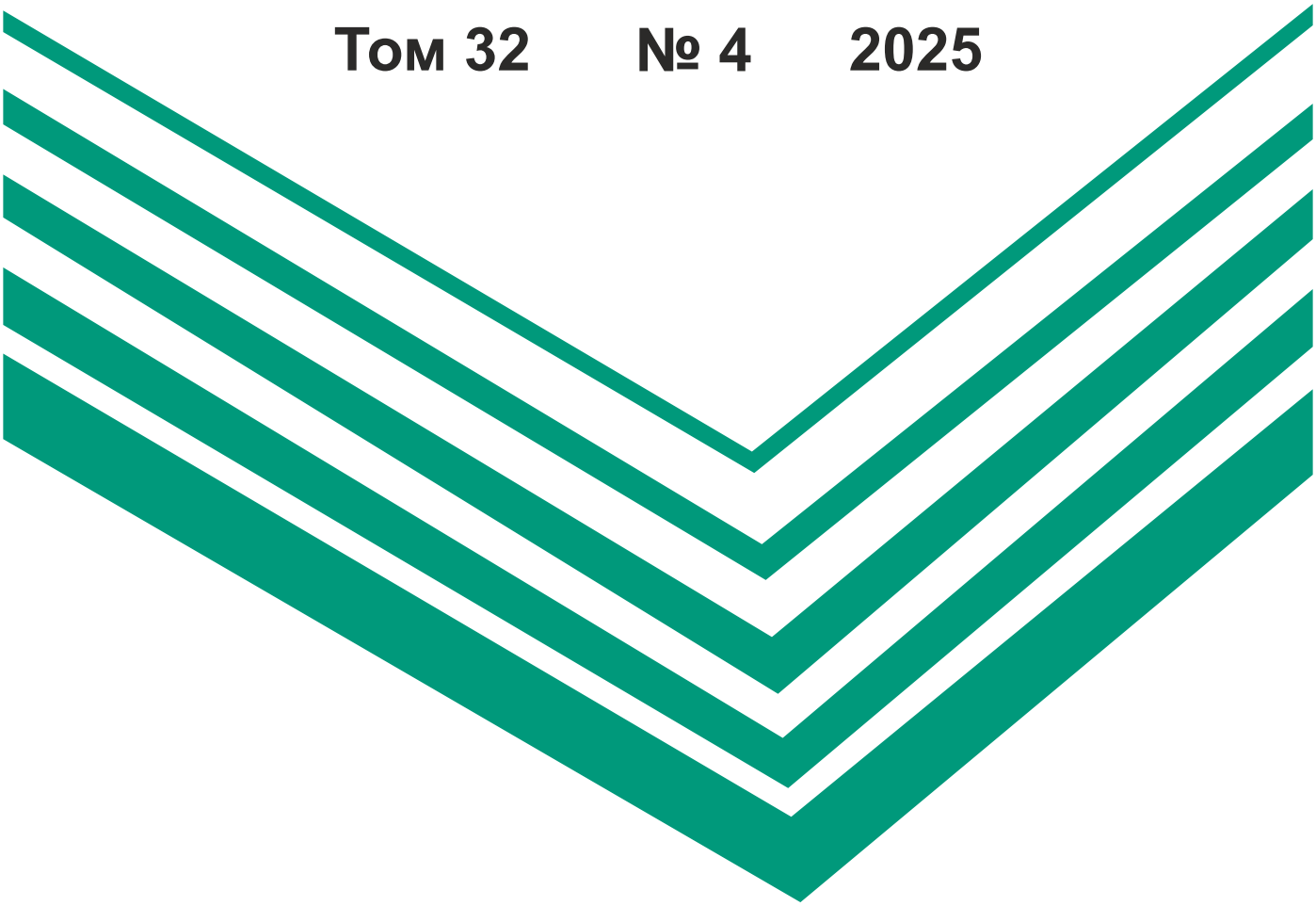
ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 4 2025



Новосибирск
Издательство Института математики

Учредители журнала Сибирское отделение РАН
Институт математики им. С. Л. Соболева СО РАН

Журнал включён в базу данных Russian Science Citation Index (RSCI) на платформе Web of Science. Переводы статей на английский язык публикуются в *Journal of Applied and Industrial Mathematics* и доступны по ссылке `www.springer.com/mathematics/journal/11754`.

СИБИРСКОЕ ОТДЕЛЕНИЕ РОССИЙСКОЙ АКАДЕМИИ НАУК
ИНСТИТУТ МАТЕМАТИКИ им. С. Л. СОБОЛЕВА СО РАН

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Выпускается с 1994 г. Научный журнал 4 номера в год

Том 32, № 4 (166) **Октябрь–декабрь 2025**

СОДЕРЖАНИЕ

(Продолжение на с. 2)

НОВОСИБИРСК
ИЗДАТЕЛЬСТВО ИНСТИТУТА МАТЕМАТИКИ

В журнале публикуются оригинальные научные статьи и обзоры теоретической и прикладной направленности по следующим разделам дискретного анализа, исследования операций и информатики:

- дискретная оптимизация
- комбинаторика
- контроль и надёжность дискретных устройств
- математические модели и методы принятия решений
- математическое программирование
- модели экономики
- моделирование процессов управления
- построение и анализ алгоритмов
- синтез и сложность управляющих систем
- теория автоматов
- теория графов
- теория игр и её приложения
- теория кодирования
- теория расписаний и размещений

Адрес редакции:
Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск, Россия
Телефон: +7 (383) 329–75–79
E-mail: `discopr@math.nsc.ru`

SIBERIAN BRANCH OF THE RUSSIAN ACADEMY OF SCIENCES
SOBOLEV INSTITUTE OF MATHEMATICS

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII

/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/

Published since 1994 | Scientific journal | 4 issues per year

Vol. 32, No. 4 (166) | **October–December, 2025**

CONTENTS

(Continued on Page 4)

NOVOSIBIRSK
SOBOLEV INSTITUTE PRESS

In this journal we publish original research papers and survey papers of both theoretical and practical importance on the following topics of discrete analysis, operations research and informatics:

- discrete optimization
- combinatorics
- control and reliability of discrete devices
- decision making models and methods
- mathematical programming
- economic models
- management modeling
- design and analysis of algorithms
- synthesis and complexity of control systems
- automata theory
- graph theory
- game theory and its applications
- coding theory
- theory of scheduling and facility location

Editorial office address:
Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue,
630090 Novosibirsk, Russia
Phone: +7 (383) 329–75–79
E-mail: discopr@math.nsc.ru

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 5–68

УДК 519.7 DOI: 10.33048/daio.2025.32.837

АТАКИ ПО ПОБОЧНЫМ КАНАЛАМ НА ТЕОРЕТИКО-КОДОВЫЕ ПОСТКВАНТОВЫЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ: ОБЗОР. ЧАСТЬ 1

А. О. Бахарев[1,2,a], *Д. М. Воронов*[1,2,b], *Н. А. Коломеец*[1,2,c], *Н. Н. Токарева*[1,2,3,d], *И. С. Хильчук*[1,2,e], *А. С. Шапоренко*[1,2,f]

[1] Национальный технологический центр цифровой криптографии,
Раменский б-р, 1, 119607 Москва, Россия
[2] Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
[3] Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: [a] a.bakharev@g.nsu.ru, [b] d.voronov2@g.nsu.ru,
[c] n.kolomeets@g.nsu.ru, [d] crypto1127@mail.ru,
[e] i.khilchuk@g.nsu.ru, [f] a.shaporenko@g.nsu.ru

Аннотация. В работе, состоящей из двух частей, приводится структурированный аналитический обзор, посвящённый атакам, использующим информацию, полученную по побочным каналам, на постквантовые криптосистемы, основанные на методах и конструкциях теории помехоустойчивого кодирования. В первой части обзора представлено описание основных криптографических примитивов и алгоритмов, применяемых в теоретико-кодовых криптосистемах, а также приведены описания наиболее значимых современных теоретико-кодовых схем: Classic McEliece, «Кодиеум», «Шиповник», BIKE и HQC. Представленное исследование выполнено в рамках НИР «Кульминация», проведённой в АНО «Национальный технологический центр цифровой криптографии». Табл. 5, ил. 14, библиогр. 111.

Ключевые слова: постквантовая криптография, атака по побочным каналам, теоретико-кодовая криптографическая система.

Введение

Криптография с открытым ключом является важнейшей компонентой современной цифровой связи. Протоколы с открытым ключом применяются для выполнения трёх основных криптографических функций:

шифрование с открытым ключом, цифровые подписи и выработка общего секретного ключа. Большинство современных протоколов с открытым ключом основаны на сложности задач факторизации и дискретного логарифмирования. В работе [1] 1997 г. Шор показал, что квантовые компьютеры могут эффективно решать каждую из них, потенциально делая все криптосистемы с открытым ключом, основанные на сложности этих двух задач, небезопасными.

Постквантовая криптография основана на других сложных задачах, для которых неизвестны эффективные алгоритмы решения с помощью квантового компьютера. Одним из основных типов таких задач являются задачи, относящиеся к области помехоустойчивого кодирования. Криптосистемы, основанные на задачах такого типа, называют *теоретико-кодовыми*.

При этом важную роль для оценки защищённости криптографических систем на практике играет учёт методов криптоанализа, которые основаны на особенностях реализаций теоретико-кодовых систем и используют информацию, получаемую злоумышленником по побочным каналам, например время выполнения криптосистемой тех или иных операций, электромагнитное излучение или энергопотребление. Таким образом, оценка устойчивости потенциальных реализаций теоретико-кодовых криптосистем относительно методов этого типа криптоанализа является актуальной задачей.

В работе, состоящей из двух частей, приводится структурированный аналитический обзор, посвящённый атакам по побочным каналам, опубликованным в открытых источниках, на современные постквантовые теоретико-кодовые криптосистемы. В разд. 1 ч. 1 приводится терминология и основные определения, которые используются в данном обзоре. Разд. 2 посвящён описанию криптографических примитивов и алгоритмов, применяемых в теоретико-кодовых криптосистемах. Введению в теоретико-кодовые криптосистемы посвящён разд. 3. В разд. 4–7 приводится описание наиболее значимых современных теоретико-кодовых схем: Classic McEliece, «Кодиеум», «Шиповник», BIKE и HQC. Разд. 8 посвящён основам атак по побочным каналам. В разд. 9 приводится обзор общей представленности работ по рассматриваемой тематике в трудах ведущих конференций.

В ч. 2 обзора проводится детальный анализ наиболее значимых работ последних лет, посвящённых атакам по побочным каналам на современные теоретико-кодовые криптосистемы, а также применимости рассмотренных атак к криптосистемам Classic McEliece, «Кодиеум» и «Шиповник».

1. Терминология и основные определения

1.1. Обозначения. Будем использовать следующие обозначения:

- $\mathbb{F}_q$ — конечное поле из $q = p^n$ элементов для простого p и натурального n; обычно $p = 2$, т. е. рассматривается $\mathbb{F}_{2^n}$;
- $\mathbb{F}_q^n$ — векторное пространство размерности n над $\mathbb{F}_q$, наиболее часто используется $\mathbb{F}_2^n$;
- координаты q-ичного вектора $x \in \mathbb{F}_q^n$ могут нумероваться как с нуля, так и с единицы в зависимости от контекста;
- $\langle A \rangle_{\mathbb{F}_q} = \langle \alpha_1, \ldots, \alpha_m \rangle_{\mathbb{F}_q}$ — линейная оболочка над полем $\mathbb{F}_q$ множества векторов $A = \{\alpha_1, \ldots, \alpha_m\} \subseteq \mathbb{F}_q^n$;
- $\mathbb{F}_q[x]$ — кольцо многочленов переменной x над полем $\mathbb{F}_q$;
- $\operatorname{rk} H$ — ранг матрицы H;
- $\operatorname{wt}(x)$ — вес Хэмминга вектора $x \in \mathbb{F}_q^n$, равный числу его ненулевых координат;
- $d(x, y)$ — расстояние Хэмминга между векторами $x, y \in \mathbb{F}_q^n$, равное числу координат, в которых x и y различаются;
- $[n, k, d]_q$-код $\mathcal{C}$ — линейное подпространство $\mathcal{C} \leqslant \mathbb{F}_q^n$ размерности k такое, что $\min\limits_{x \in \mathcal{C} \setminus \{0\}} \operatorname{wt}(x) = d$.

Обозначения в алгоритмах и протоколах:

- $\perp$ — символ ошибки;
- $[\,]$ — пустой массив;
- $x \leftarrow y$ — операция присвоения переменной x значения y;
- $a \xleftarrow{\$} \mathrm{Alg}$ — операция запуска вероятностного алгоритма Alg с последующим присвоением результата работы Alg переменной a;
- $a \xleftarrow{\mathcal{U}} A$ — операция присвоения переменной a случайного значения в соответствии с равномерным распределением на множестве A;
- $a \parallel b$ — результат конкатенации строк a и b.

Основное внимание будет обращено на следующие криптосистемы:

- NIED — криптосистема Нидеррайтера [2];
- ME — оригинальная криптосистема Мак-Элиса [3];
- CME — криптосистема Мак-Элиса, поданная на конкурс NIST под названием Classic McEliece [4];
- CODI — криптосистема «Кодиеум» [5];
- SHIP — криптосистема «Шиповник» [6].

1.2. Базовые определения. Напомним, что любую матрицу H размера $m \times n$, $m \leqslant n$, над полем $\mathbb{F}_2$ можно привести к приведённому ступенчатому виду (reduced row-echelon form) элементарными преобразованиями строк. Обозначим такую матрицу через H', будем также считать, что $\operatorname{rk} H = m$. Привести матрицу H к указанной форме можно с помощью метода Гаусса, при этом H' определяется однозначно.

Далее будем нумеровать строки, столбцы и координаты с нуля. Обозначим через c_i номер столбца, в котором стоит *ведущий элемент* i-й строки матрицы H' — первая единица слева в i-й строке и единственная в столбце c_i в силу вида H'. Ясно, что

$$0 \leqslant c_0 < c_1 < \cdots < c_{m-1} < n.$$

Напомним, что матрица H' называется *систематической*, если $c_i = i$ для всех $i = 0, \ldots, m-1$. Будем называть H' (μ, ν)-*полусистематической*, $\mu \leqslant \nu$, если

- $c_i = i$ для всех $0 \leqslant i < m - \mu$,
- $c_i \leqslant i - \mu + \nu$ для всех $0 \leqslant i < m$.

При $\mu = \nu$ любая (μ, ν)-полусистематическая матрица H' является систематической. Если $\nu > \mu$, то (μ, ν)-полусистематическая форма допускает большую свободу. В спецификации криптосистемы CME используется как $(\mu, \nu) = (0, 0)$, так и $(\mu, \nu) = (32, 64)$ (см. табл. 1: параметры с обозначением f отвечают использованию полусистематической формы).

Теория кодирования. Опишем основные определения теории кодирования. *Весом Хэмминга* $\mathrm{wt}(x)$ вектора $x \in \mathbb{F}_q^n$ называется число ненулевых координат данного вектора. *Расстоянием Хэмминга* $d(x, y)$ для векторов $x, y \in \mathbb{F}_q^n$ будем называть число координат, в которых они отличаются. Вес Хэмминга и расстояние Хэмминга связаны отношением

$$d(x, y) = \mathrm{wt}(x - y).$$

Произвольное подмножество $\mathcal{C} \subseteq \mathbb{F}_q^n$ называется *кодом длины* n, а элементы $\mathcal{C}$ — кодовыми словами. Отметим, что мы рассматриваем элементы пространства $\mathbb{F}_q^n$ как вектор-строки. *Кодовым расстоянием* d кода $\mathcal{C}$ называется величина, равная минимальному расстоянию Хэмминга между двумя различными кодовыми словами, т. е.

$$d = \min\{d(x, y) \mid x, y \in \mathcal{C},\ x \neq y\}.$$

Код *линейный*, если он образует подпространство $\mathbb{F}_q^n$. Запись $[n, k, d]_q$-код $\mathcal{C}$ означает, что $\mathcal{C} \leqslant \mathbb{F}_q^n$ — линейный код размерности k длины n с кодовым расстоянием d. Отметим, что для линейного кода справедливо равенство

$$d = \min_{x \in \mathcal{C} \setminus \{0\}} \mathrm{wt}(x).$$

Порождающей матрицей $G \in \mathbb{F}_q^{k \times n}$ линейного кода $\mathcal{C}$ называется матрица, строки которой образуют базис $\mathcal{C}$. Матрица $H \in \mathbb{F}_q^{(n-k) \times n}$ называется *проверочной*, если $x \in \mathcal{C}$ в том и только в том случае, когда

$$Hx^\top = 0.$$

Синдромом вектора $x \in \mathbb{F}_q^n$ относительно линейного кода $\mathcal{C}$ называется вектор $Hx^\top$, где H — проверочная матрица $\mathcal{C}$. *Ортогональным кодом* для $[n, k, d]_q$-кода $\mathcal{C}$ называется множество $\mathcal{C}^\perp \subseteq \mathbb{F}_q^n$, для элементов $x \in \mathcal{C}^\perp$ которого выполняется равенство

$$Gx^\top = 0.$$

Если $t \in \mathbb{N}$ — максимальное такое число, что для любых $e \in \mathbb{F}_q^n$, $\mathrm{wt}(e) \leqslant t$, и кодовых слов $x, y \in \mathcal{C}$, $x \neq y$, выполнено неравенство

$$d(x, x + e) < d(y, x + e),$$

то будем говорить, что код $\mathcal{C}$ *исправляет* t *ошибок*. Отметим, что для кода с кодовым расстоянием d имеет место равенство

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Определим задачу синдромного декодирования, на которой основывается стойкость криптосистем, построенных на кодах, исправляющих ошибки. В общем случае эта задача NP-трудна [7].

Задача 1 (задача синдромного декодирования при ровно t ошибках). *Даны проверочная матрица $H \in \mathbb{F}_2^{(n-k)\times n}$ линейного кода, целое $t > 0$ и синдром $s = He^\top \in \mathbb{F}_2^{n-k} \setminus \{0\}$, где $e \in \mathbb{F}_2^n$ и $wt(e) = t$. Найти вектор $e' \in \mathbb{F}_2^n$ такой, что $wt(e') = t$ и $He'^\top = s$.*

Пусть $g(x)$ — многочлен степени t над $\mathbb{F}_{q^m}$ и $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_{q^m}$ — различные элементы, не являющиеся корнями g. *Кодом Гоппы* длины n над полем $\mathbb{F}_q$ называется код, предложенный в [8]:

$$\Gamma(g, \alpha_0, \ldots, \alpha_{n-1}) = \left\{ (u_0, \ldots, u_{n-1}) \in \mathbb{F}_q^n \mid \sum_{i=0}^{n-1} \frac{u_i}{x - \alpha_i} \equiv 0 \pmod{g(x)} \right\}.$$

Код Гоппы представляет собой $[n, k, d]_q$-код, для которого справедливы оценки

$$k \geqslant n - m \deg g, \quad d \geqslant \deg g + 1.$$

Такие коды можно задать также с помощью обобщённых кодов Рида — Соломона. Отметим, что на практике в криптосистемах используются коды размерности $k = n - m \deg g$, (см., например, п. 4.4 с описанием процедуры CME.Kgen(): если код не удовлетворяет нужным параметрам, ключи генерируются заново).

Если выбран неприводимый многочлен, то используют термин *неприводимый* код Гоппы. Важным частным случаем таких кодов является двоичный код Гоппы ($q = 2$). Двоичный код Гоппы называется *сепарабельным*, если выбранный многочлен Гоппы не имеет кратных корней.

Кодовое расстояние двоичного сепарабельного кода Гоппы удовлетворяет неравенству

$$d \geqslant 2 \deg g + 1,$$

т. е. указанный код исправляет не менее $\deg g$ ошибок.

Шифрование с открытым ключом и инкапсуляция ключа. Здесь рассматриваются схемы шифрования с открытым ключом, схемы инкапсуляции ключа и то, как получить одно из другого.

Определение 1. Для заданного пространства сообщений $\mathcal{M}$ *схемой шифрования с открытым ключом* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{PKE} = (\mathsf{Kgen}, \mathsf{Enc}, \mathsf{Dec})$:

- $(pk, sk) \xleftarrow{\$} \mathsf{PKE.Kgen}()$ — алгоритм генерации ключевой пары, возвращает открытый ключ pk и секретный ключ sk;
- $c \xleftarrow{\$} \mathsf{PKE.Enc}(pk, m)$ — алгоритм шифрования, принимает на вход сообщение m и открытый ключ pk и возвращает шифртекст c;
- $b \leftarrow \mathsf{PKE.Dec}(sk, c)$ — алгоритм расшифрования, принимает на вход секретный ключ sk и шифртекст c и возвращает m, если c — корректный шифртекст сообщения m, и $\perp$ иначе.

Для схемы шифрования с открытым ключом PKE должно выполняться стандартное требование корректности зашифрования и расшифрования сообщения:

$$(pk, sk) \xleftarrow{\$} \mathsf{PKE.Kgen}() \Rightarrow \mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, m)) = m.$$

Здесь и далее требование корректности может не выполняться с пренебрежимо малой вероятностью вне зависимости от входных и генерируемых данных алгоритма.

Определение 2. Для заданного пространства ключей $\mathcal{K}$ *схемой инкапсуляции ключа* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{KEM} = (\mathsf{Kgen}, \mathsf{Encaps}, \mathsf{Decaps})$:

- $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kgen}()$ — алгоритм генерации ключевой пары, возвращает открытый ключ pk и секретный ключ sk;
- $(K, c) \xleftarrow{\$} \mathsf{KEM.Encaps}(pk)$ — алгоритм инкапсуляции, на вход принимает открытый ключ pk и возвращает ключ K и его инкапсуляцию c;
- $K \leftarrow \mathsf{KEM.Decaps}(sk, c)$ — алгоритм декапсуляции, на вход принимает секретный ключ sk и инкапсуляцию c и возвращает K, если c — корректная инкапсуляция ключа K, и $\perp$ иначе.

Для схемы инкапсуляции ключа KEM должно выполняться стандартное требование корректности алгоритмов инкапсуляции и декапсуляции:

$$(pk, sk) \xleftarrow{\$} \mathsf{Kgen}(),\ (K, c) \leftarrow \mathsf{Encaps}(pk) \Rightarrow K = \mathsf{Decaps}(sk, c).$$

После небольшого введения в терминологию теоретической (доказуемой) стойкости определим модели угроз, относительно которых принято рассматривать стойкость представленных криптосистем.

Противник $\mathcal{A}$ — некоторый вероятностный алгоритм (вероятностная машина Тьюринга). Под вычислительными ресурсами противника будем понимать величину, ограничивающую сумму времени работы противника (например число тактов вычислений) и размера его программы (эта оговорка необходима для исключения ситуаций, в которых в код противника прописываются некоторые предвычисленные таблицы, упрощающие перебор). Запись $\mathcal{A}^{\mathcal{O}}$ означает противника (вероятностный алгоритм) $\mathcal{A}$, имеющего доступ к оракулу $\mathcal{O}$. Под записью $\mathbb{P}[\mathcal{A} \to a]$ понимаем вероятность того, что вероятностный алгоритм $\mathcal{A}$ выдаст a. Противник взаимодействует с окружением (экспериментом) посредством обращения к набору оракулов, которые формализуют возможности противника по взаимодействию с некоторой реальной системой.

Через $\mathrm{Exp}^{M}_{\mathcal{K}}$ обозначим эксперимент — вероятностный алгоритм, моделирующий для противника условия (модель) M, в рамках которых он взаимодействует с криптосистемой $\mathcal{K}$. Будем рассматривать два типа экспериментов.

- В экспериментах первого типа противнику необходимо реализовать угрозу, определив некоторую искомую величину, например найти дискретный логарифм или подделать подпись (задача поиска).
- В экспериментах второго типа перед началом взаимодействия с противником случайно равновероятно выбирается один из двух сценариев воспроизведения, а противнику необходимо определить, по какому сценарию воспроизводится эксперимент (задача различения). В этом случае для модели используются обозначения M-0 или M-1 в зависимости от номера сценария воспроизведения. Примером эксперимента второго типа может служить задача различения генерируемых криптосистемой ключей с одной стороны и случайных строк с другой, или задача различения Диффи — Хеллмана.

Для противника $\mathcal{A}$ определим его *преимущество* $\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A})$ в модели M для криптосистемы $\mathcal{K}$ как вероятность успешного прохождения эксперимента:

$$\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{M}_{\mathcal{K}}(\mathcal{A}) \to 1\big]$$

для экспериментов первого типа и

$$\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{M\text{-}1}_{\mathcal{K}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{M\text{-}0}_{\mathcal{K}}(\mathcal{A}) \to 1\big]\big|$$

для экспериментов второго типа. В настоящей работе через $\mathrm{InSec}(\mathrm{par})$ будем обозначать максимальное преимущество среди всех противников

с ограничением par на вычислительные и иные ресурсы (количество запросов к оракулам, их максимальные и суммарные длины и т. д.), т. е.

$$\mathrm{InSec}(\mathrm{par}) = \max_{\mathcal{A} \in A(\mathrm{par})} \mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}),$$

где $A(\mathrm{par})$ — множество всех противников с ограничением par на вычислительные и иные ресурсы.

Определение 3. *Преимуществом* противника $\mathcal{A}$ в модели OW-CPA для схемы шифрования PKE назовём величину

$$\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big],$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}$ определён на рис. 1.

$\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $c^* \leftarrow \mathsf{Enc}(pk, m)$
4 : $m' \leftarrow \mathcal{A}(pk, c^*)$
5 : **if** $m' = m$ **then**
6 : **return** 1
7 : **else**
8 : **return** 0

Рис. 1

$\mathrm{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\mathsf{PKE}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $(m_0, m_1) \leftarrow \mathcal{A}(pk)$
3 : $c \leftarrow \mathsf{Enc}(pk, m_b)$
4 : $b' \leftarrow \mathcal{A}(pk, c)$
5 : **if** $b' = b$ **then**
6 : **return** 1
7 : **else**
8 : **return** 0

Рис. 2

Через $\mathrm{InSec}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(t)$ будем обозначать максимум среди преимуществ вида $\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы.

Определение 4. *Преимуществом* противника $\mathcal{A}$ в модели IND-CPA для схемы шифрования PKE назовём величину

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CPA\text{-}1}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CPA\text{-}0}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big]\big|,$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\mathsf{PKE}}$, $b \in \{0, 1\}$, определён на рис. 2.

Определение 5. *Преимуществом* противника в модели IND-CCA для механизма инкапсуляции ключа KEM назовём величину

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CCA\text{-}1}}_{\mathsf{KEM}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CCA\text{-}0}}_{\mathsf{KEM}}(\mathcal{A}) \to 1\big]\big|,$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{KEM}}, b \in \{0, 1\}$, определён на рис. 3.

$\mathrm{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{KEM}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $(K_0^*, c^*) \xleftarrow{\$} \mathsf{Encaps}(pk)$
3 : $K_1^* \xleftarrow{\mathcal{U}} \mathcal{K}$
4 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{decaps}}}(c^*, K_b^*)$
5 : **return** b'

$\mathcal{O}_{\mathrm{decaps}}(c)$

1 : **if** $c = c^*$ **then**
2 : **return** $\perp$
3 : **else**
4 : $K \leftarrow \mathsf{Decaps}(sk, c)$
5 : **return** K

Рис. 3

Через $\mathrm{InSec}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(t, q_{\mathrm{decaps}})$ будем обозначать максимум среди преимуществ вида $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы, выполняющим не более q_{decaps} запросов к оракулу $\mathcal{O}_{\mathrm{decaps}}$.

Используя преобразование Фуджисаки — Окамото [9] (FO-преобразование), из стойкой в модели OW-CPA или в модели IND-CPA схемы шифрования с открытым ключом PKE можно получить стойкую в модели IND-CCA схему инкапсуляции ключа KEM. В общем случае модель стойкости рассматриваемого шифрования с открытым ключом может быть другой, но в случае кодовых криптосистем интересны модели OW-CPA и IND-CPA. В настоящей работе будем рассматривать два вида FO-преобразования: $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$. Пусть схема шифрования с открытым ключом

KEM.Kgen()

1 : $(pk, sk') \xleftarrow{\$} \mathsf{PKE.Kgen}()$
2 : $s \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $sk \leftarrow (sk', s)$
4 : **return** (sk, pk)

KEM.Encaps(pk)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $c \xleftarrow{\$} \mathsf{PKE.Enc}(pk, m)$
3 : $K \leftarrow H(m, c)$ // $\mathsf{U}^{\not\perp}$
4 : $K \leftarrow H(m)$ // $\mathsf{U}^{\not\perp}_m$
5 : **return** (K, c)

KEM.Decaps(sk, c)

1 : $(sk', s) \leftarrow sk$
2 : $m' \leftarrow \mathsf{PKE.Dec}(sk', c)$
3 : **if** $m' = \perp$ **then**
4 : $K \leftarrow H(s, c)$
5 : **else**
6 : $K \leftarrow H(m', c)$ // $\mathsf{U}^{\not\perp}$
7 : $K \leftarrow H(m')$ // $\mathsf{U}^{\not\perp}_m$
8 : **return** K

Рис. 4. Преобразования $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$

PKE стойкая в модели OW-CPA или IND-CPA. Тогда для некоторой хэш-функции H, действующей в пространство ключей $\mathcal{K}$, способ получения стойкой в модели IND-CCA схемы инкапсуляции ключа KEM с использованием преобразований $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$ представлен на рис. 4.

Отметим, что в [10] показана эквивалентность преобразований $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$ в модели QROM, представляющей собой модель безопасности над моделью M. В модели QROM в качетсве противника выступает полиномиальный квантовый алгоритм, имеющий классический доступ к оракулам модели M и квантовый доступ к случайной функции, моделирующей хэш-функцию в рассматриваемом криптомеханизме.

Схемы подписи. Здесь рассматриваются схемы подписи, их модели угроз и один из способов их построения.

Определение 6. *Схемой подписи* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{SS} = (\mathsf{Kgen}, \mathsf{Sign}, \mathsf{Verify})$:

• $(sk, vk) \xleftarrow{\$} \mathsf{SS.Kgen}()$ — алгоритм генерации ключевой пары, возвращает ключ подписи sk и ключ проверки подписи vk;

• $\sigma \xleftarrow{\$} \mathsf{SS.Sign}(sk, m)$ — алгоритм формирования подписи, принимает на вход ключ подписи sk и сообщение m и возвращает подпись σ для сообщения m;

• $b \xleftarrow{\$} \mathsf{SS.Verify}(vk, m, \sigma)$ — алгоритм проверки подписи, принимает на вход ключ проверки подписи vk, сообщение m и подпись σ и возвращает 1, если подпись верна, и 0 иначе.

Для схемы подписи SS должно выполняться стандартное требование корректности формирования и проверки подписи:

$$(sk, vk) \xleftarrow{\$} \mathsf{SS.Kgen}() \Rightarrow \mathsf{SS.Verify}(vk, m, \mathsf{SS.Sign}(sk, m)) = 1.$$

Определение 7. *Преимуществом* противника $\mathcal{A}$ в модели SUF-CMA для схемы подписи SS назовём величину

$$\mathrm{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A}) \to 1\big],$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}$ определён на рис. 5.

Через $\mathrm{InSec}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(t, q_{\mathrm{sign}})$ будем обозначать максимум среди всех преимуществ вида $\mathrm{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы и делающим не более q_{sign} запросов к оракулу $\mathcal{O}_{\mathrm{sig}}$.

Аналогичным образом можно определить модель UF-CMA, единственное отличие которой заключается в том, что в списке L хранятся только

$\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $L \leftarrow [\,]$
3 : $(m, \sigma) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{sig}}}(pk)$
4 : **if** $(m, \sigma) \notin L \wedge \mathsf{Verify}(vk, m, \sigma)$ **then**
5 : **return** 1
6 : **else**
7 : **return** 0

$\mathcal{O}_{\mathrm{sig}}(m)$

1 : $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$
2 : $L \leftarrow L \cup (m, \sigma)$
3 : **return** σ

Рис. 5

сообщения m. Иными словами, даже верная новая подпись σ для сообщения m, которое уже было подано на вход оракулу $\mathcal{O}_{\mathrm{sig}}$, будет не принята экспериментом.

В настоящей работе будут рассматриваться схемы подписи, полученные из Σ-протоколов. Σ-*протоколом* называется схема идентификации, в которой один из участников (доказывающий) должен доказать другой стороне (проверяющему) знание некоторого секрета без его раскрытия. Σ-протокол состоит из трёх сообщений — обязательство, вызов, ответ — и имеет вид, представленный на рис. 6.

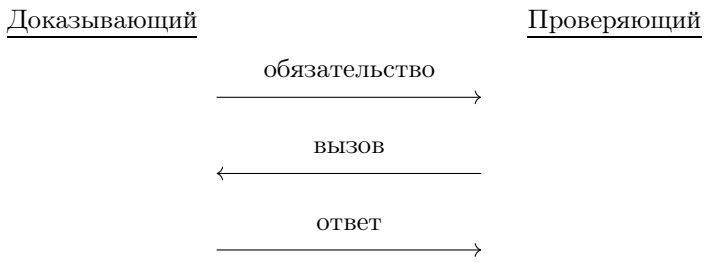


Рис. 6. Σ-протокол

В сообщении «обязательство» доказывающий отправляет результат некоторой (односторонней) функции от выбранных значений, что позволяет проверяющей стороне убедиться в том, что соответствующие значения доказывающего были выбраны до сообщения «вызов», не зная соответствующих значений до сообщения «ответ». В этом сообщении проверяющий выбирает некоторые значения, в соответствии с которыми доказывающий должен сформировать сообщение «ответ».

Используя преобразование Фиата — Шамира [11], можно получить схему подписи из Σ-протокола. Основная идея преобразования Фиата — Шамира состоит в отсутствии отправки первых двух сообщений и замене сообщения «вызов» значением $H(\text{«обязательство»} \parallel m)$, где H — это некоторая криптографическая хэш-функция, действующая в множество сообщений «вызов», а m — подписываемое сообщение.

2. Криптографические примитивы и базовые определения

2.1. Аддитивное быстрое преобразование Фурье. В криптосистемах, рассматриваемых в настоящей работе, используются многочлены над полем $\mathbb{F}_{2^m}$ характеристики 2. При этом часто требуется найти значения таких многочленов на множестве точек или во всех точках поля $\mathbb{F}_{2^m}$ (см., например, [4]).

Другими словами, для многочлена $f \in \mathbb{F}_{2^m}[x]$ требуется вычислить значения $f(\alpha_1), \ldots, f(\alpha_{2^m})$, где $\{\alpha_1, \ldots, \alpha_{2^m}\} = \mathbb{F}_{2^m}$. Для этого можно использовать алгоритм из [12], который называется *аддитивным быстрым преобразованием Фурье* и обозначается FFT (алгоритм 1). Стоит отметить, что в литературе имеются и другие вариации алгоритма FFT.

Алгоритм 1. Аддитивное быстрое преобразование Фурье (FFT)

Вход: многочлен $f \in \mathbb{F}_{2^m}[x]$, $\deg f < 2^m$, $m \geqslant 1$ целое, линейно независимые над $\mathbb{F}_2$ элементы $\beta_1, \ldots, \beta_m \in \mathbb{F}_{2^m}$, массив $B = \langle \beta_1, \ldots, \beta_m \rangle_{\mathbb{F}_2}$.

Выход: $\mathrm{FFT}(f, m, B) = (f(B[0]), \ldots, f(B[2^m - 1]))$, где $B[i] = c_1\beta_1 + \cdots + c_m\beta_m$ при $c_1 2^0 + c_2 2^1 + \cdots + c_m 2^{m-1} = i$.

if $m = 1$ **then**
  return $(f(0), f(\beta_1))$
$g(x) \leftarrow f(\beta_m x)$
найти $g_{0,i}, g_{1,i} \in \mathbb{F}_{2^m}$: $g(x) = \sum\limits_{i=0}^{2^{m-1}-1} (g_{0,i} + g_{1,i}x)(x^2 - x)^i$ ▷ [12, алгоритм 1]
$g_0(x) \leftarrow \sum\limits_{i=0}^{2^{m-1}-1} g_{0,i}x^i$, $g_1(x) \leftarrow \sum\limits_{i=0}^{2^{m-1}-1} g_{1,i}x^i$
for $i = 1, \ldots, m-1$ **do**
  $\gamma_i \leftarrow \beta_i\beta_m^{-1}$, $\delta_i \leftarrow \gamma_i^2 - \gamma_i$
$\Gamma \leftarrow \langle \gamma_1, \ldots, \gamma_{m-1} \rangle_{\mathbb{F}_2}$
$\Delta \leftarrow \langle \delta_1, \ldots, \delta_{m-1} \rangle_{\mathbb{F}_2}$
$(u_0, \ldots, u_{2^{m-1}-1}) \leftarrow \mathrm{FFT}(g_0, m-1, \Delta)$
$(v_0, \ldots, v_{2^{m-1}-1}) \leftarrow \mathrm{FFT}(g_1, m-1, \Delta)$
for $i = 0, \ldots, 2^{m-1} - 1$ **do**
  $w_i = u_i + v_i\Gamma[i]$
  $w_{i+2^{m-1}} = w_i + v_i$
return $(w_0, \ldots, w_{2^m-1})$

2.2. Методы декодирования кодов Гоппы.

Алгоритм Берлекэмпа — Мэсси. Будем использовать интерпретацию алгоритма Берлекэмпа — Мэсси, представленную здесь в виде алгоритма 2.

Алгоритм 2. Алгоритм Берлекэмпа — Мэсси

Вход: последовательность $a_1, \ldots, a_n \in \mathbb{F}_q$.
Выход: многочлен $f(x) = 1 + f_1 x + \cdots + f_L x^L$ минимальной степени такой, что $a_j + f_1 a_{j-1} + \cdots + f_L a_{j-L} = 0$, $j = L+1, L+2, \ldots, n$.

$f(x) \leftarrow 1$, $b(x) \leftarrow 1$, $L \leftarrow 0$
for $r = 1, \ldots, n$ **do**
  $\Delta \leftarrow a_r + f_1 a_{r-1} + \cdots + f_L a_{r-L}$
  if $\Delta = 0$ **then**
    $b(x) \leftarrow xb(x)$.
  else
    $b'(x) \leftarrow f(x) - \Delta x b(x)$
    if $2L < r$ **then**
      $b(x) \leftarrow \Delta^{-1} f(x)$
      $f(x) \leftarrow b'(x)$
      $L \leftarrow r - L$
    else
      $b(x) \leftarrow xb(x)$
      $f(x) \leftarrow b'(x)$
return $f(x)$

Существуют вариации алгоритма, минимизирующие число обращений элементов поля, а также вариации, использующие альтернативное описание. Возможно сделать реализацию для декодирования кодов, ориентированную на операцию расшифрования в кодовых криптосистемах, в которой искомый многочлен находится за постоянное время.

Алгоритм Берлекэмпа — Мэсси применим для решения системы уравнений

$$\begin{bmatrix} a_n & a_{n-1} & \ldots & a_1 \\ a_{n+1} & a_n & \ldots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n-1} & a_{2n-2} & \ldots & a_n \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} -a_{n+1} \\ -a_{n+2} \\ \vdots \\ -a_{2n} \end{bmatrix}.$$

Декодирование двоичного кода Гоппы на основе алгоритма Берлекэмпа — Мэсси.

Вход: синдром $s = Hy^\top \in \mathbb{F}_2^t$ принятого сообщения $y \in \mathbb{F}_2^n$, многочлен $g \in \mathbb{F}_{2^m}[x]$, $\deg g = t$, и элементы $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_{2^m}$, определяющие код Гоппы $\Gamma(g, \alpha_0, \ldots, \alpha_{n-1}) \subseteq \mathbb{F}_2^n$.

Выход: вектор $e \in \mathbb{F}_2^n$ такой, что $He^\top = 0$ и $\mathrm{wt}(y-e) \leqslant t$, при этом $H(y-e)^\top = s$

Шаг 1. Вычислить проверочную матрицу двойного размера $2t \times n$:

$$H_{(2)} = \begin{bmatrix} \frac{1}{g^2(\alpha_0)} & \frac{1}{g^2(\alpha_1)} & \cdots & \frac{1}{g^2(\alpha_{n-1})} \\ \frac{\alpha_0}{g^2(\alpha_0)} & \frac{\alpha_1}{g^2(\alpha_1)} & \cdots & \frac{\alpha_{n-1}}{g^2(\alpha_{n-1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_0^{2t-1}}{g^2(\alpha_0)} & \frac{\alpha_1^{2t-1}}{g^2(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^{2t-1}}{g^2(\alpha_{n-1})} \end{bmatrix}.$$

Шаг 2. Привести матрицу $H_{(2)}$ к двоичному виду $H'_{(2)}$, заменив каждый элемент двоичным столбцом длины m согласно используемому представлению поля (неприводимый многочлен задан в спецификациях криптосистем).

Шаг 3. Вычислить синдром $s_{(2)} = H'_{(2)}(s, 0, \ldots, 0)^\top$ длины $2t$.

Шаг 4. С помощью алгоритма Берлекэмпа — Мэсси по синдрому $s_{(2)}$ найти многочлен — локатор ошибок $\sigma(x)$ такой, что $y_i - e_i = 1$ тогда и только тогда, когда $\sigma(\alpha_i) = 0$ при $i = 0, \ldots, n-1$.

Шаг 5. Найти $\deg\sigma$ корней многочлена $\sigma(x)$, например, последовательной подстановкой в него ненулевых элементов поля $\mathbb{F}_{2^m}$.

Шаг 6. Учитывая систематическую форму используемых в криптосистемах в качестве открытого ключа матриц, корней многочлена $\sigma(x)$ достаточно, чтобы подсчитать e, так как в качестве y подходит вектор $(s, 0, \ldots, 0)$ размерности n.

Заметим, что в криптосистеме CME шаг 5 выполняется с использованием аддитивного быстрого преобразования Фурье (см. алгоритм 1).

2.3. Расширенный алгоритм Евклида. Представляет собой естественное обобщение обычного алгоритма Евклида и изложен в виде алгоритма 3.

2.4. Метод Гаусса. Используется для приведения матрицы к систематическому виду, что требуется, например, для определения открытых ключей в системе CME. Соответствующий алгоритм представлен в виде алгоритмов 4 и 5: исключение столбца и собственно метод Гаусса.

2.5. Алгоритм расщепления носителя. Предложенный Сандрие в работе [13] алгоритм позволяет для линейного $[n, k, d]_q$-кода $\mathcal{C}$ с тривиальной группой автоморфизмов и перестановочно эквивалентного ему кода $\mathcal{D}$ эффективно определять перестановку $\pi \in S_n$, переводящую $\mathcal{C}$ в $\mathcal{D}$. Хорошо работает на линейных кодах, похожих на случайные. Многие рассматриваемые далее атаки используют этот алгоритм как одну

Алгоритм 3. Расширенный алгоритм Евклида (EEA)

Вход: многочлены $a(z)$, $b(z)$: $\deg a \geqslant \deg b$, d_{fin}.
Выход: многочлены $u(z)$, $r(z)$: $r(z) = b(z)u(z) \bmod a(z)$, $\deg r \leqslant d_{\text{fin}}$.

$r_{-1}(z) \leftarrow a(z)$, $r_0(z) \leftarrow b(z)$
$u_{-1}(z) \leftarrow 1$, $u_0(z) \leftarrow 0$
$i \leftarrow 0$
while $\deg r_i(z) > d_{\text{fin}}$ **do**
    $i \leftarrow i + 1$
    $q_i(z) \leftarrow r_{i-2}(z) \operatorname{div} r_{i-1}(z)$
    $r_i(z) \leftarrow r_{i-2}(z) \bmod r_{i-1}(z)$ ▷ $r_{i-2}(z) = q_i(z)r_{i-1}(z) + r_i(z)$
    $u_i(z) \leftarrow u_{i-2}(z) - q_i(z)u_{i-1}(z)$
$N \leftarrow i$
return $u_N(z)$, $r_N(z)$

Алгоритм 4. Исключение столбца EliminateColumn(H_j, j)

Вход: матрица $H_j \in \mathbb{F}_2^{tm \times n}$, номер столбца $j \in \{1, \ldots, tm\}$.
Выход: матрица $H_{j+1} \in \mathbb{F}_2^{tm \times n}$: $H_{j+1}[:, j] = e_j^\top$, $H_{j+1} = G_j H_j$ для некоторой обратимой матрицы $G_j \in \mathbb{F}_2^{tm \times tm}$, или $\perp$.

$H_{j+1} \leftarrow H_j$
if $H_{j+1}[j, j] \neq 1$ **then** ▷ убедиться, что $H_{j+1}[j, j] = 1$
    $k \leftarrow \min\{j + 1 \leqslant r \leqslant tm \mid H_{j+1}[r, j] = 1\}$
    if такое k не существует **then**
        return $\perp$
    $H_{j+1}[j, :] \leftarrow H_{j+1}[j, :] + H_{j+1}[k, :]$ ▷ прибавить k-ю строку к j-й
for $i \in \{1, \ldots, tm\} \setminus \{j\}$ **do**
    if $H_{j+1}[i, j] = 1$ **then** ▷ убедиться, что $H_{j+1}[i, j] = 0$ при $i \neq j$
        $H_{j+1}[i, :] \leftarrow H_{j+1}[i, :] + H_{j+1}[j, :]$ ▷ прибавить j-ю строку к i-й
return H_{j+1}

Алгоритм 5. Метод Гаусса

Вход: матрица $H \in \mathbb{F}_2^{tm \times n}$: $\operatorname{rk} H[1{:}tm, 1{:}tm] = tm$.
Выход: систематическая форма $H' = (I_{tm} \mid A) \in \mathbb{F}_2^{tm \times n}$ матрицы H.

$H_1 \leftarrow H$
for $j = 1, \ldots, tm$ **do**
    $H_{j+1} \leftarrow \text{EliminateColumn}(H_j, j)$
return $H' \leftarrow H_{tm+1}$

из своих составных частей. В рамках настоящей работы нет необходимости в его детальном описании.

2.6. Декодирование по информационным совокупностям. Алгоритм, обозначаемый ISD и представленный в виде алгоритма 6 в соответствии с работой Штерна [14], находит кодовое слово веса w в $[n,k]$-коде $\mathcal{C}$ с порождающей матрицей $G \in \mathbb{F}_2^{k\times n}$. В дополнение на вход алгоритма подаются параметры $p \in \{0,1,\ldots,w\}$ и $\ell \in \{0,1,\ldots,n-k\}$. Также фиксируется маскирующая функция $\varphi(x) = x_{k+1}x_{k+2}\cdots x_{k+\ell}$.

Алгоритм 6. Алгоритм Штерна

Вход: порождающая матрица $G \in \mathbb{F}_2^{k\times n}$, параметры $w, p, \ell \in \mathbb{N}$.
Выход: кодовое слово $c \in \mathcal{C}$ веса w.

$L \leftarrow \varnothing$
repeat
  применить случайную перестановку столбцов π к матрице G
  привести полученную матрицу $\pi(G)$ к виду $G' = (I_k \mid A)$
  $L_1 \leftarrow \varnothing$, $L_2 \leftarrow \varnothing$
  for $u \in \{v \in \mathbb{F}_2^{k/2} \mid \mathrm{wt}(v) = p\}$ **do**
    добавить $x = (u \parallel 0)G'$ в список L_1
    добавить $x' = (0 \parallel u)G'$ в список L_2
  отсортировать список L_1 согласно функции $\varphi(x)$
  отсортировать список L_2 согласно функции $\varphi(x)$
  for $x \in L_1$ **do**
    for $x' \in L_2$ **do**
      if $\varphi(x) = \varphi(x')$ **then**
        добавить вектор $(x \parallel x')$ в список L
until существует $(x \parallel x') \in L$ такой, что $\mathrm{wt}(x) = w - 2p$
if $x = (u \parallel 0 \parallel x_{k+1}\ldots x_n) \wedge x' = (0 \parallel u' \parallel x'_{k+1}\ldots x'_n)$ **then**
  return $c \leftarrow (u \parallel u')G$

3. Теоретико-кодовые криптосистемы

Теоретико-кодовыми криптосистемами обычно называют системы, которые основаны на задачах помехоустойчивого кодирования. В настоящее время криптосистемы такого типа представляют повышенный интерес в связи с развитием постквантовой криптографии.

Первой криптосистемой кодового типа является криптосистема Мак-Элиса [3], разработанная в 1978 г. Основная идея данной криптосистемы состоит в маскировке некоторого линейного кода под код, не обладающий видимой алгебраической и комбинаторной структурой. Как известно, общая задача декодирования линейного кода NP-трудна [7], однако, зная структуру такого кода, можно легко расшифровать сообщение. В оригинальном варианте криптосистемы в качестве кодов, исправляющих ошибки, используются двоичные коды Гоппы [8].

Далее, Нидеррайтер в 1986 г. предложил свой вариант кодовой криптосистемы [2], основанный на обобщённых кодах Рида — Соломона. Этот вариант оказался нестойким: в 1992 г. В. М. Сидельников и С. О. Шестаков опубликовали атаку на криптосистему Нидеррайтера [15]. Авторы полагали, что их атака применима и к двоичным кодам Гоппы, но это оказалось не так, и вариант криптосистемы Нидеррайтера на двоичных кодах Гоппы всё ещё не взломан [16].

Криптосистема Мак-Элиса, как и многие другие кодовые криптосистемы, обладает значимым преимуществом — высокой скоростью зашифрования и расшифрования. Однако её основной недостаток — большой размер открытого ключа. В связи с этим предпринято множество попыток модифицировать криптосистему с применением разных семейств кодов. Абсолютное большинство таких попыток не увенчалось успехом. Такие системы обычно взламывают, опираясь на структуру используемого кода, или их стойкость не превосходит стойкости криптосистем на кодах Гоппы. Так, криптосистема Сидельникова на основе кодов Рида — Маллера, предложенная в 1994 г. [17], взломана в 2007 г. [18]. Различные вскрытые модификации системы Мак-Элиса описаны в монографии [19].

В настоящее время Национальный институт стандартов и технологий США (NIST) проводит ряд конкурсов по выбору квантово-устойчивых алгоритмов с целью стандартизации набора схем инкапсуляции ключа и цифровой подписи. В четвёртом раунде конкурса NIST [21] участвовали четыре криптосистемы KEM, три из которых кодовые — CME, BIKE и HQC, а последняя признана ненадёжной. CME использует идею Нидеррайтера, при этом названа в честь Мак-Элиса в силу использования кодов Гоппы; BIKE основан на квазициклических кодах (MDPC); в различных же модификациях HQC могут использоваться также квазициклические коды. В работе [20] приведён анализ производительности данных криптосистем, из которого становится ясно, что наиболее производительной с точки зрения реализации является HQC, демонстрирующая наименьшее время выполнения и наименьший размер ключа среди трёх систем. CME наименее производительная, вместе с тем наиболее изученная. Однако, вопреки ожиданиям многих специалистов, в качестве стандарта была выбрана криптосистема HQC. Обратим внимание, что обзор третьего раунда конкурса NIST представлен в [22], а в обзоре второго раунда [23] отмечается особая важность исследования стойкости систем к атакам по побочным каналам.

Отметим, что атаки по побочным каналам на теоретико-кодовые криптосистемы принципиально не отличаются от атак на любые другие. Идея заключается в выявлении алгоритма, допускающего утечки по побочным каналам, и исследовании возможности использования полученной информации для восстановления секрета.

ME.Kgen()

$g(x) \xleftarrow{\mathcal{U}} \{g \in \mathbb{F}_{2^m}[x] \mid g \text{ неприводимый}, \deg g = t\}$

$(\alpha_0, \ldots, \alpha_{n-1}) \xleftarrow{\mathcal{U}} \{\alpha \in \mathbb{F}_{2^m}^n \mid \alpha_i \neq \alpha_j \text{ при } i \neq j\}$

$G \leftarrow$ порождающая матрица кода $\Gamma(g, \alpha_0, \ldots, \alpha_{n-1})$

$k \leftarrow \dim \Gamma =$ длина сообщения m

$S \xleftarrow{\mathcal{U}} \mathrm{GL}_k(2)$

$P \xleftarrow{\mathcal{U}}$ множество перестановочных матриц порядка n

$G_{\mathrm{pub}} \leftarrow SGP$

return $(pk = G_{\mathrm{pub}},\ sk = (g, \alpha_0, \ldots, \alpha_{n-1}, S, P))$

ME.Enc(pk, m)

$e \xleftarrow{\mathcal{U}} \{e \in \mathbb{F}_2^n \mid \mathrm{wt}(e) = t\}$

$c \leftarrow mG_{\mathrm{pub}} + e$

return c

ME.Dec(sk, c)

$c' \leftarrow cP^{-1}$

найти m': $m'G + e' = c', \mathrm{wt}(e') \leqslant t$

// алгоритм декодирования кода Гоппы Γ

if m' найдено $\wedge\ \mathrm{wt}(e') = t$ **then**

  $m \leftarrow m'S^{-1}$

  return m

else

  return $\perp$

Рис. 7. Схема шифрования Мак-Элиса [3]

4. Криптосистема Classic McEliece

4.1. Обозначения:

- n — длина кода, количество точек для кода Гоппы;
- $q = 2^m$ — размер поля;
- через m в криптосистемах также обозначено сообщение, но из контекста всегда ясно, что подразумевается;
- t — степень многочлена Гоппы и вес вектора ошибок; необходимо, чтобы $mt < n$;
- $k = n - mt$ — размерность кода Гоппы для используемых параметров;
- $f(z) \in \mathbb{F}_2[z]$ — неприводимый нормированный многочлен степени m, определяющий поле $\mathbb{F}_q$ в виде элементов кольца $\mathbb{F}_2[z]/(f(z))$ и двоичных векторов из $\mathbb{F}_2^m$, образованных коэффициентами остатков от деления на многочлен $f(z)$; в большинстве описаний будет опущен;

NIED.Kgen()

1 : $g(x) \xleftarrow{\mathcal{U}} \{g \in \mathbb{F}_{2^m}[x] \mid g \text{ неприводимый}, \deg g = t\}$
2 : $(\alpha_0, \ldots, \alpha_{n-1}) \xleftarrow{\mathcal{U}} \left\{\alpha \in \mathbb{F}_{2^m}^n \mid \alpha_i \neq \alpha_j \text{ при } i \neq j\right\}$
3 : $h_{i,j} \leftarrow \alpha_j^i / g(\alpha_j)$, $i \in [0, t-1]$
4 : $(\beta_{i,j,0}, \ldots, \beta_{i,j,m-1}) \leftarrow h_{i,j}$ // $\beta_{i,j,l} \in \mathbb{F}_2$
5 : $\widehat{h}_{im+l,j} \leftarrow \beta_{i,j,l}$
6 : $\widehat{H} \leftarrow (\widehat{h}_{i,j}) \in \mathbb{F}_2^{mt \times n}$
7 : представить $\widehat{H}$ в виде $(I_{n-k} \mid H')$, $H' \in \mathbb{F}_2^{mt \times mt}$
8 : **return** $(pk = H', sk = (g, \alpha_0, \ldots, \alpha_{n-1}))$

NIED.Enc(pk, m)

1 : $m \in \mathcal{M}_{n,t}$
2 : $c \leftarrow (I_{n-k} \mid H')m^\top$
3 : **return** $c^\top$

NIED.Dec(sk, c)

1 : $H \leftarrow (I_{n-k} \mid H')$
2 : $c' = c \parallel 0^k$
3 : найти e: $He^\top = 0 \wedge \mathrm{wt}(c' \oplus e) \leqslant t$
4 : // алгоритм Берлекэмпа — Мэсси
5 : **if** e найден $\wedge\ \mathrm{wt}(c' + e) = t$ **then**
6 : **return** $m \leftarrow c' + e$
7 : **else**
8 : **return** $\perp$

Рис. 8. Схема шифрования Нидеррайтера [2]

• $\mathcal{M}_{n,t} = \left\{x \in \mathbb{F}_2^n \mid \mathrm{wt}(x) = t\right\}$ — множество сообщений для криптосистем NIED, CME, CODI.

4.2. Оригинальная криптосистема Мак-Элиса (ME). Криптосистема с открытым ключом (PKE), представленная на рис. 7, предложена Мак-Элисом в 1978 г. [3].

Однако обратим внимание, что криптосистема, поданная на конкурс NIST [4] под названием Classic McEliece, построена на базе криптосистемы Нидеррайтера, при этом она также использует коды Гоппы.

4.3. Криптосистема Нидеррайтера. На рис. 8 приведено описание криптосистемы Нидеррайтера [2], которая, как и оригинальная криптосистема Мак-Элиса, относится к классу PKE. Данное описание согласовано с описанием криптосистемы CME, в том числе в нём используется код Гоппы.

Обратим внимание, что сообщениями в данной криптосистеме являются элементы $\mathcal{M}_{n,t}$, т. е. двоичные векторы размерности n и веса t.

4.4. Криптосистема CME. Перейдём к описанию криптосистемы, получившей название Classic McEliece. Её спецификация [4] датируется 23.10.2022 г. Согласно заявке авторами криптосистемы являются исследователи из более чем 10 организаций.

В режиме PKE её процедуры Enc и Dec совпадают с описанными ранее NIED.Enc и NIED.Dec. Далее опишем возможные параметры криптосистемы, CME.Kgen() и функции KEM.

Заданные в спецификации значения параметров отражены в табл. 1 (их описание см. в п. 4.1). Пустые μ, ν можно интерпретировать как $(\mu, \nu) = (0, 0)$ (или (i, i) для любого $0 \leqslant i < n$).

Таблица 1

Параметры криптосистемы CME

Бит. стой-кость	Название	m	t	n	k	μ, ν	Откр. ключ, МБ	Секр. ключ, КБ	Шифр-текст, Б
128	mceliece348864	12	64	3488	2720	—	0,25	6,34	96
128	mceliece348864f	12	64	3488	2720	32, 64	0,25	6,34	96
192	mceliece460896	13	96	4608	3360	—	0,5	13,29	156
192	mceliece460896f	13	96	4608	3360	32, 64	0,5	13,29	156
256	mceliece6688128	13	128	6688	5024	—	1	13,61	208
256	mceliece6688128f	13	128	6688	5024	32, 64	1	13,61	208
256	mceliece6960119	13	119	6960	5413	—	1	13,63	194
256	mceliece6960119f	13	119	6960	5413	32, 64	1	13,63	194
256	mceliece8192128	13	128	8192	6528	—	1,3	13,79	208
256	mceliece8192128f	13	128	8192	6528	32, 64	1,3	13,79	208

Для описания процедуры генерации ключей CME.Kgen() определим следующие дополнительные параметры.

- H — криптографическая хэш-функция, возвращающая ℓ битов.
- Входы функции H будем записывать следующим образом: $\mathsf{H}(b, v, c)$, где $b \in \mathbb{F}_2$, $v \in \mathbb{F}_2^n$ и $c \in \mathbb{F}_2^{mt}$ — шифртекст. Кодирование параметров в единую строку для хэширования происходит следующим образом: число b представляется 1 байтом, вектор v — $\lceil n/8 \rceil$ байтами, вектор c — $\lceil mt/8 \rceil$ байтами.
- G — криптографический генератор псевдослучайных битов, принимающий на вход двоичную строку длины ℓ (т. е. имеет seed размера ℓ).

В спецификации предлагается использовать $\ell = 256$, а в качестве выхода H брать первые ℓ бит выхода функции SHAKE256. В качестве G также используется SHAKE256: на вход ей подаётся байт, в котором записано число 64, за которым следуют ещё $\lceil \ell/8 \rceil$ байтов, содержащих ℓ-битный вход генератора G (итого 33 байта).

Процедура CME.$\mathsf{Kgen}(\delta)$

Вход: $\delta \in \mathbb{F}_2^\ell$.

Выход: (pk, sk) — ключевая пара.

Шаг 1. Получить $(s, r, \delta') = \mathsf{G}(\delta)$, где $\delta' \in \mathbb{F}_2^\ell$, $s \in \mathbb{F}_2^n$, а длина двоичной строки r достаточна для генерации остальных параметров (конкретное её значение несущественно в контексте данной работы).

Шаг 2. Определить порядок $\alpha_0, \ldots, \alpha_{q-1}$ элементов поля $\mathbb{F}_q$ и неприводимый многочлен g при помощи строки r. Если что-либо не удалось сгенерировать корректно, то вся процедура CME.Kgen перезапускается с $\delta = \delta'$.

Шаг 3. Построить для кода Гоппы $\Gamma(\alpha_0, \ldots, \alpha_{n-1})$ проверочную матрицу H.

Шаг 4. Привести матрицу H к систематическому или (μ, ν)-полусистематическому виду (в зависимости от выбранных параметров криптосистемы). Результатом шага 4 является набор $(T, c_{mt-\mu}, \ldots, c_{mt-1}, \Gamma')$ такой, что

- $c_{mt-\mu}, \ldots, c_{mt-1}$ — номера ведущих столбцов получившейся (μ, ν)-полусистематической матрицы ($c_i = i$ для систематической матрицы);
- Γ' — код Гоппы, полученный из Γ перестановкой столбцов с номерами $c_{mt-\mu}, \ldots, c_{mt-1}$ в позиции $mt-\mu, \ldots, mt-1$, для этого достаточно перенумеровать используемые точки $\alpha_0, \ldots, \alpha_{n-1}$ в $\alpha'_0, \ldots, \alpha'_{n-1}$;
- проверочная матрица кода Γ' представима в систематическом виде $(I_{mt} \mid T)$, где I_{mt} — единичная матрица порядка mt.

Аналогично шагу 2 вся процедура CME.Kgen перезапускается с $\delta = \delta'$, если проверочную матрицу кода Γ' не удалось привести к нужному виду. На шаге 4 используется метод Гаусса (см. п. 2.4), и на его реализацию направлена одна из атак по сторонним каналам, рассматриваемых ниже.

Шаг 5. $pk \leftarrow T$.

Шаг 6. $sk \leftarrow (\delta, c, g, \alpha', s)$, где

- $c = (c_{mt-\mu}, \ldots, c_{mt-1})$,
- $\alpha' = (\alpha'_0, \ldots, \alpha'_{n-1}, \alpha_n, \alpha_{q-1})$ с учётом возможной перенумерации $\alpha_0, \ldots, \alpha_{n-1}$ в $\alpha'_0, \ldots, \alpha'_{n-1}$ на шаге 4.

CME.$\mathsf{Encaps}(pk)$

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$
2 : $c \leftarrow \text{NIED.}\mathsf{Enc}(pk, m)$
3 : $K \leftarrow \mathsf{H}(1, m, c)$
4 : **return** (K, c)

CME.$\mathsf{Decaps}(sk, c)$

1 : $(sk_{\text{NIED}}, s) \leftarrow sk$
2 : $m \leftarrow \text{NIED.}\mathsf{Dec}(sk_{\text{NIED}}, c)$
3 : **if** $m \neq \perp$ **then**
4 : **return** $K \leftarrow \mathsf{H}(1, m, c)$
5 : **else**
6 : **return** $K \leftarrow \mathsf{H}(0, s, c)$

Рис. 9. Схема Classic McEliece [4]

Сообщения в криптосистеме CME представляются векторами из $\mathcal{M}_{n,t}$. На рис. 9 приведены её KEM-функции CME.Encaps и CME.Decaps. Отметим, что в актуальной версии криптосистемы [4] эти функции упрощены по сравнению со спецификацией 2020 г., представленной на третий раунд конкурса NIST [24]. Один из аргументов, обозначенных в рамках четвёртого раунда конкурса NIST [25], — сделать отличия от существующего патента U.S. 9912479 ещё более явными.

4.5. Известные реализации.

- Официальная реализация [4] и несколько неофициальных, ссылки на которые размещены на странице [26].
- Наработки в рамках библиотеки CIRCL [27, 28].
- Реализация для OpenSSH, включённая в библиотеку liboqs [29] и основанная на программном коде PQClean [30].
- Реализация для VPN-протокола WireGuard [31].
- Свободные реализации [32–34].
- Много информации представлено в презентации Бернштейна [35].

5. Криптосистема «Кодиеум»

Схема инкапсуляции ключа «Кодиеум» была разработана в рамках деятельности рабочей группы 2.5 «Постквантовые криптографические механизмы» Технического комитета по стандартизации «Криптографическая защита информации» и представлена на XXVI Международной научно-практической конференции «РусКрипто — 2024» [5]. Как и в Classic McEliece (см. рис. 9), в Кодиеуме используется схема шифрования Нидеррайтера (см. рис. 8). Для получения итоговой схемы используется FO-преобразование $\mathsf{U}^{\not\perp}_m$ в отличие от используемого в CME преобразования $\mathsf{U}^{\not\perp}$. Схема криптосистемы «Кодиеум» представлена на рис. 10. В этой криптосистеме применяется хэш-функция $h\colon \{0,1\}^* \to \{0,1\}^{512}$ — Стрибог-512 [36].

CODI.Kgen()

1 : $(pk, sk) \leftarrow \text{NIED.Kgen}()$
2 : $s \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $sk' \leftarrow (sk, s)$
4 : **return** (pk, sk')

CODI.Encaps(pk)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $c \leftarrow \text{NIED.Enc}(pk, m)$
3 : $K \leftarrow h(0 \parallel m)$
4 : **return** (K, c)

CODI.Decaps(sk')

1 : $(sk, c) \leftarrow sk'$
2 : $m' \leftarrow \text{NIED.Dec}(sk, c)$
3 : **if** $m' \neq \perp$ **then**
4 : **return** $K \leftarrow h(0 \parallel m')$
5 : **else**
6 : **return** $K \leftarrow h(1 \parallel s \parallel c)$

Рис. 10. Схема «Кодиеум» [5]

В [37] с помощью результатов из [10] получены оценки стойкости схемы инкапсуляции ключа «Кодиеум». Однако, как и для криптосистемы Classic McEliece, оценки стойкости могут быть улучшены, что следует из недавних результатов [38]. Данная криптосистема находится на начальной стадии стандартизации, из чего следует, что возможны изменения в строении схемы, в том числе в выборе класса используемых кодов. Авторам настоящей работы не удалось найти реализации криптосистемы «Кодиеум», что свойственно для недавно представленных криптосистем.

В табл. 2 представлены параметры схемы «Кодиеум» и соответствующие им размеры ключей и шифртекстов для различных уровней битовой стойкости в модели QROM. Отметим, что как указано ранее, данные параметры могут быть улучшены в свете недавней работы [38].

Таблица 2

Параметры схемы «Кодиеум»

Битовая стойкость	m	n	k	t	Открытый ключ, МБ	Секретный ключ, КБ	Шифртекст, Б
128	13	16960	14230	210	4,63	29,32	341
192	13	31620	27902	286	12,37	54,49	465
256	13	51980	47404	352	25,75	95,12	574

Секретная информация: $s \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$.
Открытая информация: $H \xleftarrow{\mathcal{U}} \mathbb{F}_2^{(n-k)\times n}$, $y^\top \leftarrow Hs^\top$.

	Доказывающий		Проверяющий
1 :	$u \xleftarrow{\mathcal{U}} \mathbb{F}_2^n$, $\sigma \xleftarrow{\mathcal{U}} S_n$		
2 :	$c_0 \leftarrow h(\sigma \parallel Hu^\top)$		
3 :	$c_1 \leftarrow h(\sigma(u))$		
4 :	$c_2 \leftarrow h(\sigma(u \oplus s))$		
5 :		$\xrightarrow{c_0, c_1, c_2}$	
6 :			$b \xleftarrow{\mathcal{U}} \{0, 1, 2\}$
7 :		$\xleftarrow{b}$	
8 :	**if** $b = 0$ **then**		
9 :	$r_0 \leftarrow \sigma$, $r_1 \leftarrow u$		
10 :	**if** $b = 1$ **then**		
11 :	$r_0 \leftarrow \sigma$, $r_1 \leftarrow u \oplus s$		
12 :	**if** $b = 2$ **then**		
13 :	$r_0 \leftarrow \sigma(u)$, $r_1 \leftarrow \sigma(s)$		
14 :		$\xrightarrow{r_0, r_1}$	
15 :			**if** $b = 0$ **then**
16 :			// проверить
17 :			$c_0 = h(r_0 \parallel Hr_1^\top)$
18 :			$c_1 = h(r_0(r_1))$
19 :			**if** $b = 1$ **then**
20 :			// проверить
21 :			$c_0 = h(r_0 \parallel (Hr_1^\top \oplus y))$
22 :			$c_2 = h(r_0(r_1))$
23 :			**if** $b = 2$ **then**
24 :			// проверить
25 :			$c_1 = h(r_0)$
26 :			$c_2 = h(r_0 \oplus r_1)$
27 :			$\mathrm{wt}(r_1) = t$

Рис. 11. Схема идентификации Штерна [39]

6. Криптосистема «Шиповник»

Так же, как и схема инкапсуляции ключа «Кодиеум», схема подписи «Шиповник» была разработана в рамках деятельности рабочей группы 2.5 «Постквантовые криптографические механизмы» Технического комитета по стандартизации «Криптографическая защита информации». Впервые эта схема представлена без названия на XXIII Международной научно-практической конференции «РусКрипто — 2021» [6]. В основе схемы подписи «Шиповник» лежит схема идентификации Штерна [39] (рис. 11), в которой в качестве хэш-функции $h\colon \{0,1\}^* \to \{0,1\}^{512}$ применяется хэш-функция Стрибог-512 [36].

В [39] представлена стратегия, при которой противник, не знающий секретную информацию, может успешно пройти схему идентификации с вероятностью $2/3$. Тогда для уменьшения вероятности успешной атаки

SHIP.Kgen()

$s \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$
$y \leftarrow Hs^\top$
return $(pk = y,\ sk = s)$

SHIP.Verify$(pk, m, (c \parallel r))$

$b \leftarrow h'(m \parallel c)$
for $i = 0, \ldots, \delta - 1$ **do**
  if $b_i = 0 \wedge$
    $\wedge\, (c_{i,0} \neq h(r_{i,0} \parallel Hr_{i,1}^\top) \vee$
    $\vee\, c_{i,1} \neq h(r_{i,0}(r_{i,1})))$ **then**
    return 0
  if $b_i = 1 \wedge$
    $\wedge\, (c_{i,0} \neq h(r_{i,0} \parallel (Hr_{i,1}^\top \oplus y)) \vee$
    $\vee\, c_{i,2} \neq h(r_{i,0}(r_{i,1})))$ **then**
    return 0
  if $b_i = 2 \wedge$
    $\wedge\, (c_{i,1} \neq h(r_{i,0}) \vee$
    $\vee\, c_{i,2} \neq h(r_{i,0} \oplus r_{i,1}) \vee$
    $\vee\, \mathrm{wt}(r_{i,1}) \neq t)$ **then**
    return 0
return 1

SHIP.Sign(sk, m)

for $i = 0, \ldots, \delta - 1$ **do**
  $u_i \xleftarrow{\mathcal{U}} \mathbb{F}_2^n$
  $\sigma_i \xleftarrow{\mathcal{U}} S_n$
  $c_{i,0} \leftarrow h(\sigma_i \parallel Hu_i^\top)$
  $c_{i,1} \leftarrow h(\sigma(u_i))$
  $c_{i,2} \leftarrow h(\sigma(u_i \oplus s))$
  $c_i \leftarrow c_{i,0} \parallel c_{i,1} \parallel c_{i,2}$
$c \leftarrow c_0 \parallel \cdots \parallel c_{\delta-1}$
$b \leftarrow h'(m \parallel c)$
for $i = 0, \ldots, \delta - 1$ **do**
  if $b_i = 0$ **then**
    $r_i \leftarrow \sigma_i \parallel u_i$
  if $b_i = 1$ **then**
    $r_i \leftarrow \sigma_i \parallel (u_i \oplus s)$
  if $b_i = 2$ **then**
    $r_i \leftarrow \sigma_i(u_i) \parallel \sigma_i(s)$
$r \leftarrow r_0 \parallel \cdots \parallel r_{\delta-1}$
return $c \parallel r$

Рис. 12. Схема «Шиповник» [40]

следует повторить протокол δ раз. Для перехода от схемы идентификации к схеме подписи используется преобразование Фиата — Шамира (см. п. 1.2). Для этого необходима троичная хэш-функция $h'\colon \{0,1\}^* \to \{0,1,2\}^\delta$, определённая следующим образом:

$$h'(x) = \left\lfloor \frac{h(x) \cdot 3^\delta}{2^{512}} \right\rfloor,$$

где двоичные и троичные строки естественным образом отождествляются с натуральными числами. Схема подписи «Шиповник» представлена на рис. 12.

Стойкость данной криптосистемы анализируется в [40–42]. В табл. 3 приведены параметры схемы «Шиповник» и соответствующие им размеры открытого ключа и подписи для различных уровней битовой стойкости с учётом результатов доказуемой стойкости согласно работам [6, 40].

Таблица 3

Параметры схемы «Шиповник»

Битовая стойкость	n	k	t	δ	Открытый ключ, МБ	Подпись, МБ
80	2896	1448	318	137	0,25	0,62
128	4841	2421	533	219	0,70	1,75
256	8841	4421	973	438	2,33	6,78
512	16818	8409	1850	876	8,43	27,43

6.1. Известные реализации. Открытая реализация [43] отечественного постквантового алгоритма «Шиповник» компании «Криптонит» выполнена компанией QApp в рамках деятельности в составе рабочей группы «Постквантовые криптографические механизмы» Технического комитета 26 Росстандарта. Проект написан на языке C с оптимизацией под наборы команд SSE4.1, SSE2 и MMX.

7. Криптосистемы HQC и BIKE

Положим $\mathcal{R} = \mathbb{F}_2[X]/(X^r - 1)$ — кольцо циклических многочленов степени не выше $r-1$ с коэффициентами из поля $\mathbb{F}_2$. Простое p называется *примитивным*, если многочлен $(X^p - 1)/(X - 1)$ неприводим над $\mathbb{F}_2$.

Линейный код $\mathcal{C} \leqslant \mathbb{F}_2^n$ *квазициклический* (n_0-квазициклический), если циклический сдвиг произвольного кодового слова из $\mathcal{C}$ на n_0 позиций также является кодовым словом. Проверочная матрица $H \in \mathbb{F}_2^{r \times n}$ такого кода имеет блочный вид

$$H = (H_0 \mid H_1 \mid \cdots \mid H_{n_0-1}),$$

где r простое, $n = n_0 r$. Каждый блок $H_i \in \mathbb{F}_2^{r\times r}$ — циркулянтная матрица, в которой каждая строка получается циклическим сдвигом предыдущей строки на 1 позицию вправо (сдвиг последней строки приводит к первой). Благодаря изоморфизму между кольцом циркулянтных матриц размера $r \times r$ и кольцом многочленов $\mathcal{R}$ сложение и умножение матриц может выполняться в кольце многочленов. Под QC-MDPC-кодом понимается квазициклический код с проверочной матрицей, все строки которой имеют фиксированный вес $w = O(\sqrt{n})$.

Схемы инкапсуляции ключа BIKE и HQC, безопасные в модели IND-CCA, получаются из схем шифрования с открытым ключом, безопасных в модели IND-CPA. Приведём описание задач, на сложности решения которых основана безопасность схем HQC-PKE (задача QCSD) и BIKE-PKE (задачи QCSD и QCCF) в модели IND-CPA.

Задача 2 (задача синдромного декодирования квазициклического кода, QCSD). *Даны проверочная матрица квазициклического кода* $H \in \mathbb{F}_2^{(n-k)\times n}$, *синдром* $s \in \mathbb{F}_2^{n-k}$ *и целое* $t > 0$. *Найти вектор* $e \in \mathbb{F}_2^n$ *такой, что* $wt(e) \leqslant t$ *и* $He^\top = s$.

Задача 3 (задача поиска кодового слова квазициклического кода, QCCF). *Даны проверочная матрица* $H \in \mathbb{F}_2^{(n-k)\times n}$ *квазициклического кода и целое* $t > 0$. *Найти вектор* $c \in \mathbb{F}_2^n$ *такой, что* $wt(c) = t$ *и* $Hc^\top = 0$.

Основные виды атак, направленных на кодовые криптосистемы на основе квазициклических кодов, — декодирование на основе информационных совокупностей [44] (см. также п. 2.6) и его модификации, а также атаки, направленные на структуру кода [45] или вид многочлена, порождающего циклическую структуру [46, 47]. Помимо этого схемы могут быть уязвимы для GJS-атаки, представленной в работе [48] 2016 г., в случае повторного использования пары ключей. GJS, или реакционная атака, использует корреляцию между секретным ключом и паттернами ошибок, вызывающими отказ декодирования. Нескольких таких паттернов достаточно для проведения успешной атаки. В работе [49] показано, что основную вычислительную сложность имеет задача обнаружения первого такого паттерна. Атаки по побочным каналам, направленные на HQC и BIKE, описаны в разд. 9 ниже и в ч. 2 обзора.

7.1. Схема HQC. Hamming Quasi-Cyclic — схема инкапсуляции ключа на основе квазициклических кодов без скрытой структуры, которая была представлена на конкурс NIST [50]. Её основными преимуществами являются малый размер открытого ключа и точная оценка вероятности ошибки расшифрования. Кроме того, в отличие от большинства существующих кодовых криптосистем, основанных на схеме шифрования Мак-Элиса, стойкость схемы HQC опирается на сложность решения

HQC.KeyGen()

1 : $h \xleftarrow{\mathcal{U}} \mathcal{R}$
2 : $G \leftarrow$ порождающая матрица кода $\mathcal{C}$ // $G \in \mathbb{F}_2^{k \times n}$
3 : $sk \leftarrow (x, y) \xleftarrow{\$} \mathcal{R}_w \times \mathcal{R}_w$
4 : $pk \leftarrow (h, s = x + h \cdot y)$
5 : **return** (pk, sk)

HQC.Encrypt(pk, m)

1 : $m \xleftarrow{\mathcal{U}} \mathbb{F}_2^k$
2 : $e \xleftarrow{\mathcal{U}} \mathcal{R}_{w_e}$
3 : $r \leftarrow (r_1, r_2) \xleftarrow{\$} \mathcal{R}_{w_r} \times \mathcal{R}_{w_r}$
4 : $u \leftarrow r_1 + h \cdot r_2$
5 : $v \leftarrow \text{truncate}(mG + s \cdot r_2 + e, \ell)$
6 : $c \leftarrow (u, v)$
7 : **return** c

HQC.Decrypt(sk, c)

1 : $m' \leftarrow \mathcal{C}.\mathsf{Decode}(v - u \cdot y)$
2 : **return** m'

Рис. 13. Схема HQC-PKE

задачи QCSD, а не на скрытую структуру кода, исправляющего ошибки. Актуальная на данный момент спецификация обновлена 19.02.2025 г. [51].

В схеме HQC используются два вида кодов. Первый — открытый известный $[n, k]$-код $\mathcal{C} \in \mathbb{F}_2^n$, исправляющий Δ ошибок и являющийся конкатенацией кодов Рида — Маллера и Рида — Соломона, для которого имеются эффективные алгоритмы кодирования $\mathcal{C}.\mathsf{Encode}$ и декодирования $\mathcal{C}.\mathsf{Decode}$. Второй — случайный квазициклический код длины $2n$ и размерности n с проверочной матрицей $(I, \operatorname{rot} h) \in \mathbb{F}_2^{2n \times n}$, где $\operatorname{rot} h$ — циркулянтная матрица, индуцированная вектором h. В отличие от $\mathcal{C}$, предполагается, что никто не знает эффективного алгоритма декодирования для этого кода. Отметим, что его декодирование не требуется ни для шифрования, ни для расшифрования при работе схемы.

Параметры криптосистемы отражены в табл. 4 и представляют собой набор $(n, k, \Delta, w, w_r, w_e, \ell)$, где n — наименьшее примитивное число, большее $n_1 n_2$ (произведение длин кодов Рида — Соломона и Рида — Маллера соответственно), которое позволяет избежать алгебраической атаки

Таблица 4

Параметры схемы HQC

Битовая стойкость	n_1	n_2	n	w	$w_r = w_e$	Открытый ключ, Б	Секретный ключ, Б	Шифр-текст, Б
128	46	384	17,669	66	75	2,249	56	4,497
192	56	640	35,851	100	114	4,522	64	9,042
256	90	640	57,637	131	149	7,245	72	14,485

факторизацией многочлена. Лишние $\ell = n - n_1 n_2$ битов, появляющиеся в результате, отсекаются операцией truncate$(v, \text{num-bits})$. Далее k — кодовое расстояние кода $\mathcal{C}$, w — вес Хэмминга векторов x, y, а w_r, w_e — веса Хэмминга векторов r_1, r_2 и e.

Безопасность схемы шифрования с открытым ключом, представленной на рис. 13, в модели IND-CPA зависит от сложности решения задачи QCSD. Итоговая схема инкапсуляции ключа получается из схемы HQC-PKE с помощью FO-преобразования, что обеспечивает безопасность в модели IND-CCA.

Создателями приведены как официальная реализация [52], так и оптимизированная для работы за постоянное время [53].

7.2. Схема BIKE. Bit-Flipping Key Encapsulation — схема инкапсуляции ключа на основе двоичных квазициклических кодов, также представленная на конкурс NIST. Актуальная на данный момент спецификация обновлена 10.10.2024 г. [54].

В основе BIKE лежит криптосистема Нидеррайтера на квазициклических кодах [55], безопасность которой в модели IND-CPA зависит от сложности решения задач QCSD и QCCF. Открытый текст представлен вектором (e_0, e_1) веса t, а шифртекст — его синдромом. Итоговая схема инкапсуляции ключа, безопасная в модели IND-CCA, получается с помощью FO-преобразования.

BIKE.Kgen()

1 : $(h_0, h_1) \xleftarrow{\mathcal{U}} \mathcal{R}_w$
2 : $h \leftarrow h_1 h_0^{-1}$
3 : $\mu \leftarrow \pi_\ell(h)$
4 : $\sigma \xleftarrow{\$} \mathcal{M}$
5 : **return** $(pk = h,\ sk = (h_0, h_1, \mu, \sigma))$

BIKE.Encaps(h)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $(e_0, e_1) \leftarrow H(m, \pi_\ell(h))$
3 : $c \leftarrow (e_0 + e_1 h, m \oplus L(e_0, e_1))$
4 : $K \leftarrow K(m, c)$
5 : **return** (K, c)

BIKE.Decaps(h_0, h_1, μ, σ, c)

1 : $e' \leftarrow \mathsf{decoder}(c_0 h_0, h_0, h_1)$,
2 : $e' \in \mathcal{R}^2 \cup \{\perp\}$
3 : $m' \leftarrow c_1 \oplus L(e')$
4 : **if** $e' = H(m', \mu)$ **then**
5 : $\quad K \leftarrow K(m', c)$
6 : **else**
7 : $\quad K \leftarrow K(\sigma, c)$
8 : **return** K

Рис. 14. Схема BIKE-KEM

Таблица 5

Параметры схемы BIKE

Битовая стойкость	r	w	t	Открытый ключ, Б	Секретный ключ, Б	Шифр-текст, Б
128	12,323	142	134	1,540	312	1,572
192	24,659	206	199	3,082	450	3,114
256	40,973	274	264	5,122	612	5,154

В представленной на рис. 14 схеме $\mathcal{M} = \{0,1\}^{\ell}$ — множество сообщений, H, K, L — хэш-функции, а $\pi_{\ell}(h)$ возвращает первые ℓ битов вектора h. Для восстановления вектора e из синдрома используется BIKE-Flip-декодер. Параметры схемы отражены в табл. 5 и представляют собой набор (r, w, t), где r — размер блока, w — вес строки проверочной матрицы, t — вес вектора ошибок.

Создателями приведены две реализации криптосистемы BIKE — официальная [56] и дополнительная для различных архитектур [57].

8. Атаки по побочным каналам

8.1. Основные понятия. Криптографические примитивы — математические преобразования, которые входным данным сопоставляют выходные, возможно, параметризированные ключом. Поскольку криптографические примитивы будут выполняться на некотором устройстве в определённой среде, они будут обладать специальными физическими характеристиками, которые, как и математические характеристики, могут быть подвержены анализу с целью выявления секретных параметров, используемых в вычислениях. Так, например, устройства потребляют энергию и выполняют поставленные задачи за определённое время. Они также обладают электромагнитным полем, рассеивают тепло и создают шум.

Атаками по побочным каналам (side-channel attacks или SCA) называются методы атак на криптографические системы и другие защищённые устройства, при которых помимо уязвимостей в программном обеспечении или алгоритме злоумышленник также использует дополнительную информацию, извлекаемую из физического поведения системы.

По степени влияния атакующего на криптографическую систему различают *пассивные* и *активные* атаки. Первые проводятся путём прослушивания и анализа информации без изменения входных и выходных данных, а вторые — путём анализа информации и входных/выходных данных с изменением данных. К пассивным атакам относятся те, которые не оказывают заметного влияния на работу системы: атакующий получает некоторую информацию о работе системы, но сама система работает

в обычном режиме. С другой стороны, при активной атаке противник оказывает некоторое влияние на поведение системы, например, изменяя входные/выходные данные. Отметим, что различие между активными и пассивными атаками больше связано с влиянием на вычислительные процессы системы, чем с физическим влиянием на устройство.

По характеру воздействия атаки делятся на *инвазивные* — атаки, при которых криптосистема вскрывается и осуществляется прямое воздействие на её компоненты, *полуинвазивные* — атаки, при которых оказывают влияние на элементы криптосистемы, но без непосредственного контакта с ней, а также *неинвазивные* — атаки, не требующие прямого воздействия на саму криптосистему. К полуинвазивным атакам можно отнести, например, использование лазерного луча. Неинвазивные атаки направлены на анализ внешней информации о работе криптосистемы, например замер энергопотребления или времени выполнения алгоритма.

8.2. Измерительная установка и модели утечек. Для практической реализации атаки по побочным каналам первостепенное значение имеет измерительная установка. Её назначение — преобразование физических характеристик наблюдаемого устройства в анализируемые цифровые данные. Далее приведём ряд элементов, из которых обычно состоят такие установки [58]:

- целевое криптографическое устройство, например смарт-карта, программируемая пользователем вентильная матрица или интегральная схема, на которой запущен какой-либо криптографический примитив, например блочный шифр;
- внешний источник питания, тактовый генератор и любые дополнительные схемы, необходимые для правильной работы устройства;
- датчик утечки; так, например, потребление энергии можно контролировать, вставив небольшой резистор в цепь питания целевого устройства; электромагнитное излучение можно улавливать с помощью простых самодельных катушек;
- устройство сбора данных, например цифровой осциллограф, подключённый к компьютеру для статистического анализа набора измерений энергопотребления или электромагнитного излучения. Такие наборы измерений называются *трассами*.

Качество измерительной установки в основном определяется объёмом шума в её трассах. Шум является центральной проблемой для атак по побочным каналам и может влиять на их эффективность. Обычно рассматриваются следующие типы шумов: физический шум, который создаётся транзисторами и их окружением; шум измерения, вызванный процессом измерения и инструментами, которые для него используются; алгоритмический шум, создаваемый некоторыми вычислениями в реализации.

Модель утечки определяет наблюдаемую информацию, которая просачивается через побочные каналы. Моделирование происходит с помощью функции утечки, которая принимает состояние устройства в качестве входных данных и возвращает значение (или набор значений), представляющее информацию, наблюдаемую из побочных каналов. Хорошие модели утечки оказывают сильное влияние на эффективность атаки по побочным каналам. Они используются как для симуляции атак, так и для повышения их эффективности. Примерами самых простых моделей утечки являются *модель расстояния Хэмминга* и *модель веса Хэмминга*. Модель веса Хэмминга предполагает, что когда значение x_0 вычисляется в устройстве, фактические утечки по побочным каналам коррелируют с весом Хэмминга этого значения $\mathrm{wt}(x_0)$. Модель расстояния Хэмминга предполагает, что когда значение x_0 переключается на значение x_1, фактические утечки по побочным каналам коррелируют с расстоянием Хэмминга этих значений, а именно $d(x_0, x_1) = \mathrm{wt}(x_0 \oplus x_1)$. Также модель утечки, используемая для атаки, возможно, не идеально соответствует реальным наблюдениям. В таком случае говорят о шуме соответствия модели. На практике используются более продвинутые модели утечек, о которых можно прочитать подробнее в работах [59–61].

8.3. Простые и разностные атаки. По способу анализа полученных данных различают *простые* атаки, при которых исследуется прямая зависимость между информацией, полученной по побочному каналу, и операциями, выполняемыми устройством, и *разностные* атаки, при которых проводится большое количество измерений с использованием статистических методов для исследования взаимосвязи между входными данными и информацией, полученной по побочному каналу [62].

Напомним, что наборы измерений энергопотребления или электромагнитного излучения называются *трассами*. При выполнении простой атаки с помощью анализа трассы можно определить операции, выполненные устройством, или, например, количество выполненных раундов. Полученная таким образом информация, возможно, не выявляет секретной информации сама по себе. Однако такой анализ трасс утечки может быть предварительным шагом в проведении более мощной атаки. Например, таким образом можно определять части трасс, которые представляют интерес для злоумышленника.

При проведении разностной атаки злоумышленник выполняет ряд следующих шагов. В начале злоумышленник определяет алгоритм и целевую платформу (например смарт-карту), с которой он намерен извлечь секретную информацию. Злоумышленник определяет тип утечки, который он хочет использовать. Также подготавливается измерительная установка. Атаки по побочным каналам обычно основаны на стратегии

«разделяй и властвуй», в которой разные части секрета восстанавливаются по отдельности. Так, например, злоумышленник может выбрать, какая часть секретного ключа является целью его атаки. Если позволено, то злоумышленник случайно выбирает входные данные, которые подаются на целевое устройство. Если нет, то предполагается, что злоумышленник может отслеживать открытые тексты. Для ряда известных входных открытых текстов и возможных значений секрета, например всех возможных заполнений части секретного ключа, злоумышленник предсказывает получаемые в целевом устройстве значения, зависящие от секретного ключа, которые будут вычислены во время выполнения алгоритма. Затем злоумышленник моделирует утечки целевого устройства для тех же наборов возможных заполнений части секретного ключа. Используя измерительную установку, злоумышленник измеряет утечку целевого устройства. Наконец, злоумышленник применяет статистические методы для сравнения предсказанных утечек с проведёнными измерениями. Если атака успешна, то ожидается, что модель, соответствующая правильному ключевому кандидату, даст наилучший результат сравнения. Подробное описание данных атак приведено, например, в [59].

8.4. Профилированные атаки. В отличие от разностных атак, при которых с помощью модели утечки злоумышленник вычисляет гипотетическую потребляемую мощность для предсказанных значений переменных, *профилированные атаки* предполагают, что злоумышленник обладает копией целевого устройства, которую он может использовать для более точной характеризации физических утечек. Наибольшее применение получили *шаблонные атаки*, которые были представлены в [63]. Далее опишем их идею.

Предполагается, что утечки могут быть смоделированы как многомерные гауссовские распределения, называемые шаблонами, которые задаются вектором математических ожиданий и ковариационной матрицей. Эти шаблоны строятся для каждого класса секретной информации на этапе профилирования и используются позже на этапе сопоставления.

Этап профилирования состоит в вычислении параметров многомерных гауссовых распределений для каждого класса секретной информации. Гауссово распределение определяется математическим ожиданием и дисперсией. Для вычисления математического ожидания и дисперсии злоумышленник использует копию целевого устройства, над которым он имеет полный контроль и может отслеживать физические свойства. Затем злоумышленник запускает целевую криптографическую операцию большое число раз с различными значениями секретного параметра, при этом используя случайные значения для других входных данных, классифицируя полученные трассы потребляемой мощности и формируя шаблоны, каждому из которых соответствует трасса энергопотребления.

После того как все шаблоны будут созданы, трасса энергопотребления может быть связана с заданным шаблоном с помощью функции плотности вероятности.

Затем происходит сопоставление зафиксированной утечки и созданных ранее шаблонов.

8.5. Основные виды утечек. Атаки по побочным каналам тесно связаны с существованием физически наблюдаемых явлений, которые вызваны выполнением вычислительных работ электронных устройств. Так, например, микропроцессоры потребляют энергию, требуют некоторого времени вычисления, а также излучают электромагнитные поля, тепло и издают шум.

1. Атаки по времени основаны на анализе времени, которое необходимо для выполнения определённых операций в криптографических системах, с целью извлечения секретной информации. Разные операции могут занимать различное время в зависимости от входных данных, что создаёт уязвимость. Данный вид атак пассивный и неинвазивный и направлен на определение секрета с помощью анализа высокоточных замеров времени выполнения алгоритма при различных входных данных. Например, в алгоритмах шифрования, если выполнение операции отличается по времени в зависимости от значения битов ключа, злоумышленник может попытаться подобрать ключ, анализируя, сколько времени занимает корректная обработка данных.

2. Атаки по потребляемой мощности основаны на том, что во время выполнения криптографических операций потребление энергии может варьироваться в зависимости от выполняемых вычислений и обрабатываемых данных. Так же, как и в случае атаки по времени, выполняется высокоточный замер мощности, потребляемой устройством, после чего проводится анализ полученных данных с целью определения выполняемых в устройстве операций. Относятся к пассивным и неинвазивным атакам.

3. Атаки по электромагнитному излучению представляют собой метод получения секретной информации с использованием анализа электромагнитных сигналов, излучаемых устройством во время его работы. Эти атаки основаны на том факте, что многие электронные устройства испускают электромагнитные волны, которые могут быть зафиксированы и проанализированы злоумышленником. Данный вид атак пассивный и неинвазивный.

4. Акустические атаки представляют собой тип атак, при которых злоумышленник использует звук, генерируемый устройством во время работы, в качестве канала для извлечения секретной информации. Основной

сложностью для проведення данного типа атак является шум. Относятся к пассивным и неинвазивным атакам.

8.6. Методы противодействия атакам по побочным каналам. Приведём общие идеи для защиты криптографических устройств от атак по побочным каналам [59].

Для повышения устойчивости устройства к физическим атакам используют щиты, конформные клеи [64], физически неклонируемые функции [65] и съёмные источники питания [66]. К основным методам противодействия также можно отнести создание постоянных утечек или зависимости утечки от некоторой случайной величины [67], внедрение задержек, а также рандомизация времени [68]. Также для защиты блочных шифров от атак по побочным каналам применяют *маскирование* порядка d — разделение каждой секретной переменной, которая встречается в вычислениях, на $d + 1$ частей [69–71]. Наиболее распространённым является булевое маскирование: представление x в виде $x = x_0 \oplus x_1 \oplus \cdots \oplus x_d$ [69]. При применении маскирования для защиты реализации блочного шифра необходимо разработать схему для работы с масками и замаскированными данными, которая должна гарантировать, что части секрета позволят восстановить ожидаемый шифртекст. Работа [72] посвящена динамической и дифференциальной логике выполнения микросхем для уменьшения зависимости потребления энергии от данных. Для уменьшения количества информации в побочных каналах возможно добавление шума.

8.7. Краткая сводка наиболее значимых работ по теме атак на известные кодовые криптосистемы. Первая кодовая криптосистема с открытым ключом была предложена Мак-Элисом [3] с использованием двоичного кода Гоппы. Наиболее широко используемым алгоритмом декодирования для кодов Гоппы являлся алгоритм Паттерсона [73]. Первая атака по времени против реализации алгоритма Паттерсона на ПК, которая позволяла раскрыть зашифрованное сообщение, была описана в [74]. Атака затем была улучшена в [75, 76] и протестирована на платформах FPGA. Дальнейший анализ алгоритма декодирования Паттерсона привёл к более серьёзным атакам, которые направлены на восстановление секретного ключа [77, 78].

Первая атака по потребляемой мощности на криптосистему Classic McEliece была предложена в [79]. Эта атака могла полностью восстановить секретный ключ. Позже в [80] была предложена ещё одна атака по потребляемой мощности, которая направлена на восстановление зашифрованного сообщения, но не на восстановление секретного ключа. Эта атака была также успешно протестирована против реализации FPGA [81].

Альтернативой кодам Гоппы являются MDPC-коды, которые позволяют использовать открытый ключ меньшего размера [55]. Низкоресурсная реализация криптосистемы Мак-Элиса ME с кодами MDPC была предложена в [82]. Эта реализация была подвержена простым атакам по потребляемой мощности и времени, и в связи с этим в [83] была предложена улучшенная реализация. Реализация на платформах FPGA была предложена в [84]. В работе [85] на эту реализацию была проведена разностная атака по потребляемой мощности.

В [86, 87] представлены атаки по побочным каналам, направленные на систему HQC, но, как отмечается в работе, посвящённой результатам третьего раунда конкурса NIST [22], они не применимы к текущим реализациям системы. В [88, 89] представлены атаки по времени на систему HQC, которые также не применимы к текущей её версии, поскольку они были направлены на класс кодов, которые больше не используются в её построении.

В [90, 91] предложены атаки на алгоритм декапсуляции системы Classic McEliece с восстановлением открытого текста и секретного ключа соответственно. В [92] рассмотрена атака на алгоритм декапсуляции системы Classic McEliece, которая нацелена на шаг вычисления полинома — локатора ошибок с помощью алгоритма Берлекэмпа — Мэсси. Авторы [93] с помощью утечки по потребляемой мощности определяют столбцы проверочной матрицы, которые возможно удалить и таким образом уменьшить длину кода, что влечёт снижение сложности решения задачи синдромного декодирования. В [94] авторы предложили атаку на восстановление ключа системы Classic McEliece, используя утечку по потребляемой мощности во время приведения проверочной матрицы к систематическому виду с помощью метода Гаусса в процессе генерации открытого ключа. В работе [95] авторы представили шаблонную атаку на синдромное декодирование, которую они применили к программной реализации Classic McEliece. В [96] представлена шаблонная атака на алгоритм декапсуляции системы Classic McEliece с восстановлением секретного ключа, а в [97] — атака на механизмы инкапсуляции ключей, основанные на FO-преобразовании и его вариантах, которая использует утечку по побочным каналам во время вычисления псевдослучайной функции при повторном шифровании в алгоритме декапсуляции KEM. Подробное описание этих атак будет приведено во второй части работы.

9. Обзор трудов ведущих конференций

В данном разделе рассмотрены работы, связанные с атаками на кодовые криптосистемы по сторонним каналам, представленные на конференциях PQCrypto, начиная с первой, и CHES с 2000 г. В трудах конференций FSE, IACR PKC, начиная с 2000 г., работ указанной тематики

не обнаружено. Несколько работ конференций ASIACRYPT и CBCrypto не вошли в данный обзор. Отметим, что в ч. 2 данного обзора детально разобраны наиболее значимые работы по теме исследования.

9.1. Международная конференция по постквантовой криптографии. Приведём краткое описание работ по атакам на кодовые криптосистемы, представленных на конференциях PQCrypto.

1. Быстрая атака на криптосистему Мак-Элиса (2008 г.). Авторы статьи [98] отмечают, что наиболее быстрая атака на оригинальную систему Мак-Элиса (из известных на 2008 г.) основана на декодировании по информационным совокупностям. Такая атака реализована в работе Канто и Шабо [99] 1998 г. и подробнее анализировалась в [100].

В [98] авторы возвращаются к исходной атаке Штерна [14] 1988 г., которая предшествовала атаке Канто и Шабо. Авторы модернизируют её и показывают, что их атака самая быстрая из известных. Они отмечают, что для первоначально предложенных параметров криптосистемы Мак-Элиса атаку можно провести на компьютерном кластере средней мощности (1400 дней на одном процессоре Core 2 Quad CPU 2,4 ГГц или 7 дней на кластере с 200 вычислительными модулями). Ранее Канто и Сандрие также указывали на то, что система Мак-Элиса не соответствует современным стандартам безопасности, но реальная атака проведена впервые. Также в статье предлагаются новые параметры для криптосистем Мак-Элиса и Нидеррайтера, которые позволяют повысить их стойкость, в том числе к предложенной авторами атаке.

2. Атаки по побочным каналам на криптосистему Мак-Элиса (2008 г.). В статье [74], по утверждению авторов, предпринята первая попытка применить подобные атаки к криптосистеме Мак-Элиса. Авторы отмечают, что простая реализация криптосистемы Мак-Элиса может иметь слабости относительно нескольких типов атак по побочным каналам. В частности, они рассматривают атаку по времени, которая была успешно применена к программной реализации криптосистемы Мак-Элиса. Предложены некоторые усовершенствования в реализации криптосистемы, чтобы противостоять атакам по энергопотреблению и памяти.

Более детально: атаку по времени авторы предпринимают по отношению к степени полинома — локатора ошибок, который используется на шаге исправления ошибки при декодировании. Проведены теоретические исследования и сама практическая атака. Авторами предложены усовершенствования реализации криптосистемы Мак-Элиса против атаки по энергопотреблению на построение проверочной матрицы кода на этапе генерации ключа, а также относительно атаки по времени доступа к памяти в отношении перестановки кодовых слов во время расшифрования.

3. Практические атаки по мощности на реализации криптосистемы Мак-Элиса (2010 г.). Напомним, что стойкость криптосистемы Мак-Элиса основана на том, что задача о декодировании произвольного линейного двоичного кода NP-трудна. Авторы [79] обращают внимание на то, что интерес к реализации постквантовых криптографических алгоритмов, таких как криптосистема Мак-Элиса, на микропроцессорных платформах существенно возрос из-за увеличения объёма памяти устройств. В связи с этим необходимо изучать их уязвимость и устойчивость к физическим атакам, например к современным атакам по мощности. В работе [79] авторы исследуют две атаки по мощности на различные реализации криптосистемы Мак-Элиса на 8-битном микропроцессоре AVR, при этом они отмечают, что подобные атаки рассмотриваются на практике впервые.

4. Атака по времени на секретную перестановку в криптосистеме Мак-Элиса (2010 г.). В [77] представлена новая атака по времени на криптосистему Мак-Элиса. Автор предлагает использовать уязвимости в алгоритме Паттерсона, которые позволяют злоумышленнику собирать информацию о секретной перестановке по побочному каналу. Как утверждает автор, полученная информация может быть использована для существенного снижения сложности атаки, основанной на полном переборе секретного ключа. Автор также описывает некоторые контрмеры к своей атаке.

5. Декодирование «одного из многих» (2011 г.). Как отмечается в статье [45], одной из самых распространённых атак на кодовые криптосистемы в целом является атака, направленная на декодирование случайного линейного кода, поэтому для выбора секретных параметров кодовой системы необходимо тщательно анализировать и измерять сложность лучших методов декодирования для кодов, которые предполагается в ней использовать. Автор рассматривает ситуацию, в которой злоумышленник имеет доступ к многим шифртекстам, и целью атаки является дешифрование какого-либо одного из них.

6. Атаки по времени на инвертирование синдрома в кодовых криптосистемах (2013 г.). В [78] представлена первая практическая атака по времени на кодовые криптосистемы. Атака основана на уязвимостях, обнаруживающихся при расшифровании, а именно — при инвертировании синдрома с помощью расширенного алгоритма Евклида. При этом для успешной атаки автор комбинирует три типа уязвимостей: восстановление нулевого элемента, уточнение первой уязвимости с получением линейных уравнений, а затем и кубических уравнений. Все подходы вместе позволяют получить дополнительную информацию о носителе — части ключа кодовой криптосистемы.

7. Устойчивые к атакам по побочным каналам реализации криптосистемы QC-MDPC Мак-Элиса на устройствах с ограниченными возможностями (2014 г.). Авторы [83] делают отсылку к работе [55], в которой предложено использование квазициклических кодов (QC-MDPC) для криптосистемы Мак-Элиса. Данные коды могут обеспечивать как относительно малый размер ключа, так и высокую производительность на скоростных вычислительных ресурсах. Однако, как отмечают авторы, для широко распространённых микроконтроллеров ранее были представлены только медленные реализации. Они представляют реализацию криптосистемы QC-MDPC Мак-Элиса, обеспечивающую стойкость на уровне 80 битов (порядка 2^{80} операций) на недорогих микроконтроллерах ARM Cortex-M4 с приемлемой производительностью 42 мс при зашифровании и 251–558 мс при расшифровании. Помимо практических вопросов, таких как генерация случайного вектора ошибок, авторы рассматривают атаки по побочным каналам на простую реализацию предложенной схемы и предлагают контрмеры для её защиты от атак по времени и мощности.

8. QC-MDPC Мак-Элиса: атака по времени и CCA2 KEM (2018 г.). В [101] проводится глубокий разбор первопричин GJS-атаки на криптосистему QC-MDPC Мак-Элиса 2016 г. [48]. Авторы предлагают контрмеры для защиты и отмечают, что вес синдрома является фундаментальной величиной, из-за которой происходит утечка секретной информации. Если по побочному каналу удаётся контролировать вес синдрома, то можно провести атаку с восстановлением ключа.

9. Декодеры QC-MDPC с несколькими «оттенками серого» (2020 г.). Схемы KEM на основе квазициклических кодов задействуют декодеры, имеющие небольшую или пренебрежимо малую частоту отказов при декодировании. Эти декодеры должны быть эффективными и реализуемыми в режиме постоянного времени. Одним из примеров такого подхода является BIKE, кандидат второго раунда конкурса NIST. Авторы [102] продолжают свои исследования по теме Black-Gray декодеров и улучшают предыдущие показатели декапсуляции BIKE.

10. Атака по мощности на реализацию криптосистемы HQC, основанную на комбинации кодов Рида — Маллера и Рида — Соломона (2022 г.). В [103] рассматривается схема HQC, являющаяся кандидатом четвёртого раунда конкурса NIST. Авторы отмечают, что начиная с третьей версии, в алгоритме используется новая комбинация кодов, а именно кода Рида — Маллера и кода Рида — Соломона, которая требует модификации ранее уже опубликованных атак. Авторы утверждают, что атака по мощности, предпринятая Uneo и соавторами на CHES 2021, на практике не работает, поскольку упущен тот факт, что реализованный декодер Рида — Маллера не имеет фиксированной границы декодирования. В своей работе [103]

они предлагают определённую модификацию атаки, что делает её успешной для рассматриваемой версии алгоритма.

11. Новая атака восстановления ключа по сторонним каналам на HQC на основе выбранного шифртекста (2022 г.). Авторы [104] вновь отмечают, что определённые этапы декодирования кодовых криптосистем уязвимы для атак по сторонним каналам, и HQC не является исключением. Авторы предлагают новую атаку по сторонним каналам для восстановления ключа HQC с использованием выбранного шифртекста. Атака опирается на преимущества повторного использования статического секретного ключа на микроконтроллере с физическим доступом. Цель авторов, как они её формулируют, состоит в том, чтобы получить статический секретный ключ, ориентируясь на этап декодирования кода Рида — Маллера при декапсуляции и, более точно, на преобразование Адамара. Информация, полученная через сторонние каналы, используется для построения оракула, который различает несколько схем декодирования кодов Рида — Маллера. Авторы показывают, как сделать запрос к оракулу таким образом, чтобы ответы предоставляли полную информацию о статическом секретном ключе. Авторы провели эксперименты и утверждают, что для извлечения всего статического секретного ключа, используемого для декапсуляции, достаточно менее 20 000 трасс в рамках электромагнитной атаки, при этом они предлагают способы защиты от неё.

9.2. Международная конференция по криптографическому оборудованию и встроенным системам. В этом пункте даётся краткое описание работ по атакам на кодовые криптосистемы, представленных на конференциях CHES.

1. QcBits — кодовая криптосистема с постоянным временем работы (2016 г.). В статье [105] представлена схема QcBits — реализация алгоритма шифрования с открытым ключом на основе схемы Нидеррайтера с квазициклическими кодами, выполняющая соответствующие операции за постоянное время для противостояния атакам по времени.

2. Атака по побочным каналам на криптосистему QcBits (2017 г.). В работе [106] демонстрируется, что QcBits, несмотря на стойкость к атакам по времени, уязвима для разностной атаки по энергопотреблению на вычисление синдрома в алгоритме декодирования. Представленная атака позволила авторам составить систему двоичных линейных уравнений с ошибками. После решения системы был полностью восстановлен ключ. В качестве меры противодействия атаке авторы предложили маскирование кодового слова путём сложения его с другим случайным кодовым словом перед процедурой вычисления синдрома.

3. Расширение ошибки в кодовых криптосистемах (2018 г.). Кодовые криптосистемы с открытым ключом имеют вероятность ошибки декодирования, что позволяет, например, проводить GJS-атаку. В статье [49] авторы значительно усиливают эту реакционную атаку, показывая, что после нахождения всего одного паттерна вектора ошибок, который ведёт к отказу декодирования, время, необходимое для нахождения другого сообщения, которое также приведёт к отказу декодирования, становится очень мало́. Этот результат часто используется в совокупности с атаками по сторонним каналам, позволяющими различать успешное и ошибочное декодирование, так как такая утечка информации по сторонним каналам позволяет значительно ускорить поиск первого трудно расшифровываемого сообщения.

4. Совершенствование атак по побочным каналам на кодовые криптосистемы (2019 г.). В статье [107] авторы улучшают атаку на QcBits, предложенную в [106], и демонстрируют, что с помощью утечки по энергопотреблению возможно восстановить ключ без необходимости решать систему зашумлённых двоичных линейных уравнений. В дополнение делается вывод, что криптосистема BIKE по состоянию на время проведения второго раунда конкурса NIST может быть также уязвима к предложенной атаке.

5. Восстановление секретного ключа атакой по времени на криптосистемы HQC и BIKE (2021 г.). В [108] исследована возможность атаки по времени на схемы BIKE и HQC, актуальные на момент публикации. Несмотря на попытку создать реализацию с постоянным временем работы, в системе HQC для генерации случайного вектора фиксированного веса в повторном шифровании при применении преобразования Фуджисаки — Окамото использована процедура выборки с отклонением, время выполнения которой зависит от начального значения θ, в свою очередь зависящего от сообщения именно при инкапсуляции и декапсуляции ключа. В схеме BIKE при декапсуляции ключа также допущены утечки по времени при генерации кодового слова фиксированного веса, что позволяет различать успешность декодирования. Эта информация впоследствии использована в GJS-атаке, позволяющей восстановить секретный ключ. Авторы предполагают, что для выполнения данной атаки злоумышленник имеет возможность взаимодействовать с системой: выполнять зашифрование с инкапсулированным ключом, подавать полученные шифртексты для декапсуляции и наблюдать за выводом процедуры декапсуляции, а также получать информацию о времени выполнения декапсуляции. Впоследствии авторы схемы HQC заменили алгоритм генерации случайного вектора заданного веса алгоритмом 5 из статьи [109] 2021 г.

6. Атака по времени доступа к памяти на криптосистему HQC (2023 г.). В [110] авторы демонстрируют атаку с выбранным шифртекстом по времени доступа к памяти на официальную реализацию системы HQC. Эта работа во многом вдохновлена атакой по времени, описанной в [108].

На стадии профилирования атакующий использует технику flush-and-reload, которая полагается на использование программами общего кэша: в первой фазе атакующий удаляет из кэша участок памяти, затем дожидается исполнения целевой программы. Наконец, атакующий снова запрашивает тот участок памяти, который удалил на первом этапе. Быстрое получение доступа к памяти означает, что при исполнении этот участок был использован и заново внесён в кэш. На основе полученной таким образом информации атакующий строит оракул PC для проверки того, что определённый шифртекст действительно расшифровывается в определённое сообщение. Уязвимость, позволявшая реализовать данную атаку, заключалась в том, что при выполнении процедуры случайной генерации векторов e и r_1 фиксированного веса в кэш загружались только ненулевые координаты векторов. Заметим, что в актуальной реализации HQC время выполнения постоянно, а указанная процедура получает доступ ко всему вектору.

7. Атака по электромагнитному излучению на криптосистему HQC (2023 г.). В работе [111] авторы предложили атаку для восстановления общего ключа на основе алгоритма распространения доверия на несколько шагов алгоритма декапсуляции схемы HQC-KEM: алгоритм декодирования кодов Рида — Соломона и алгоритм кодирования Рида — Соломона, использующийся для повторного шифрования при применении преобразования Фуджисаки — Окамото. Предполагается, что злоумышленник имеет полный контроль над точной копией устройства и может проводить измерения электромагнитного излучения при выполнении операции умножения в поле Галуа. Авторы показывают, что маскирование и перемешивание являются недостаточно эффективными стратегиями противодействия подобной атаке, и оценивают стратегию полного перемешивания, которая могла бы помешать провести данную атаку. Однако в связи со сложностью применения подобной контрмеры именно для защиты алгоритма кодирования Рида — Соломона авторы предлагают заменить его.

9.3. Азиатская конференция по криптографии ASIACRYPT. Авторы работы [48] 2016 г. предложили новую атаку на схемы шифрования с открытым ключом, использующие квазициклические коды, и назвали её реакционной. Впоследствии эту атаку стали называть атакой Гуо — Йохансона — Станковского или GJS-атакой — по именам авторов. В ходе атаки злоумышленник пытается восстановить секретный ключ,

исходя из статистики ошибок декодирования. На первом шаге он посылает специальные сообщения получателю и наблюдает за реакцией последнего: удалось ли декодировать сообщение или произошла ошибка декодирования. Анализ распределения ошибок декодирования позволяет злоумышленнику построить так называемый спектр расстояний — набор расстояний между парами единиц в секретном ключе. На втором шаге атакующий пытается восстановить секретный ключ на основе спектра расстояний. Авторы также предложили модификацию атаки для схем инкапсуляции ключа и описали контрмеры к данной атаке.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019), а также при финансовой поддержке Национального технологического центра цифровой криптографии. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Shor P. W.** Algorithms for quantum computation: Discrete logarithms and factoring // Proc. 35th Annu. Symp. Foundations of Computer Science (Santa Fe, USA, Nov. 20–22, 1994). Los Alamitos, CA: IEEE Comput. Soc., 1994. P. 124–134. DOI: 10.1109/SFCS.1994.365700.
2. **Niederreiter H.** Knapsack-type cryptosystems and algebraic coding theory // Prob. Control Inf. Theory. 1986. V. 15, No. 2. P. 157–166.
3. **McEliece R. J.** A public-key cryptosystem based on algebraic coding theory // DSN Progress Rep. 1978. V. 42–44. P. 114–116.
4. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece. Specification. Chicago: Univ. Ill. Chic., 2022. 16 p. URL: classic.mceliece.org/spec.html (accessed: 6.03.2026).
5. **Высоцкая В. В., Чижов И. В.** Постквантовая схема инкапсуляции ключа «Кодиеум» // Докл. XXVI Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 19–22 марта 2024 г.). М.: РусКрипто, 2024. 16 p. URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_chizhov.pdf (дата обращения: 6.03.2026).
6. **Высоцкая В. В., Чижов И. В.** Схема постквантовой электронной подписи на основе протокола идентификации Штерна // Докл. XXIII Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 23–26 марта 2021 г.). М.: РусКрипто, 2021. 27 p. URL: ruscrypto.ru/resource/archive/rc2021/files/02_vysotskaya_chizhov.pdf (дата обращения: 6.03.2026).

7. **Berlekamp E., McEliece R., Van Tilborg H.** On the inherent intractability of certain coding problems (corresp.) // IEEE Trans. Inf. Theory. 1978. V. 24, No. 3. P. 384–386. DOI: 10.1109/TIT.1978.1055873.
8. **Гоппа В. Д.** Рациональное представление кодов и (L, g)-коды // Пробл. передачи информации. 1971. Т. 7, № 3. С. 41–49.
9. **Hofheinz D., Hövelmanns K., Kiltz E.** A modular analysis of the Fujisaki–Okamoto transformation // Theory of cryptography. Proc. 15th Int. Conf. (Baltimore, MD, USA, Nov. 12–15, 2017). Pt. I. Cham: Springer, 2017. P. 341–371. (Lect. Notes Comput. Sci.; V. 10677). DOI: 10.1007/978-3-319-70500-2_12.
10. **Bindel N., Hamburg M., Hövelmanns K.** [et al.]. Tighter proofs of CCA security in the quantum random oracle model // Theory of cryptography. Proc. 17th Int. Conf. (Nuremberg, Germany, Dec. 1–5, 2019). Pt. II. Cham: Springer, 2019. P. 61–90. (Lect. Notes Comput. Sci.; V. 11892). DOI: 10.1007/978-3-030-36033-7_3.
11. **Fiat A., Shamir A.** How to prove yourself: Practical solutions to identification and signature problems // Advances in cryptology — CRYPTO'86. Proc. Conf. Theory and Applications of Cryptographic Techniques (Santa Barbara, USA, Aug. 11–15, 1986). Heidelberg: Springer, 1987. P. 186–194. (Lect. Notes Comput. Sci.; V. 263). DOI: 10.1007/3-540-47721-7_12.
12. **Gao S., Mateer T.** Additive fast Fourier transforms over finite fields // IEEE Trans. Inf. Theory. 2010. V. 56, No. 12. P. 6265–6272.
13. **Sendrier N.** Finding the permutation between equivalent linear codes: The support splitting algorithm // IEEE Trans. Inf. Theory. 2000. V. 46, No. 4. P. 1193–1203. DOI: 10.1109/18.850662.
14. **Stern J.** A method for finding codewords of small weight // Coding theory and applications. Proc. 3rd Int. Colloq. (Toulon, France, Nov. 2–4, 1988). Heidelberg: Springer, 1988. P. 106–113. (Lect. Notes Comput. Sci.; V. 388). DOI: 10.1007/BFb0019850.
15. **Сидельников В. М., Шестаков С. О.** О системе шифрования, построенной на основе обобщённых кодов Рида — Соломона // Дискрет. математика. 1992. Т. 4, № 3. С. 57–63.
16. **Davydov V. V., Beliaev V. V., Kustov E. F.** [et al.]. Modern variations of McEliece and Niederreiter cryptosystems // J. Sci. Tech. Inf. Technol. Mech. Opt. 2022. V. 22, No. 2. P. 324–331. DOI: 10.17586/2226-1494-2022-22-2-324-331.
17. **Сидельников В. М.** Открытое шифрование на основе двоичных кодов Рида — Маллера // Дискрет. математика. 1994. Т. 6, № 2. С. 3–20.
18. **Minder L., Shokrollahi A.** Cryptanalysis of the Sidelnikov cryptosystem // Advances in cryptology — EUROCRYPT 2007. Proc. 26th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques (Barcelona, Spain, May 20–24, 2007). Heidelberg: Springer, 2007. P. 347–360. (Lect. Notes Comput. Sci.; V. 4515). DOI: 10.1007/978-3-540-72540-4_20.
19. **Overbeck R., Sendrier N.** Code-based cryptography // Post-quantum cryptography. Heidelberg: Springer, 2009. P. 95–145.

20. **González de la Torre M. A., Hernández Encinas L., Sánchez García J. I.** Structural analysis of code-based algorithms of the NIST post-quantum call // Logic J. IGPL. 2024. V. 33, No. 5. Article ID jzae071. 12 p. DOI: 10.1093/jigpal/jzae071.
21. **Alagic G., Bros M., Ciadoux P.** [et al.]. Status report on the fourth round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2025. DOI: 10.6028/NIST.IR.8545.
22. **Alagic G., Apon D. C., Cooper D.** [et al.]. Status report on the third round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2022. DOI: 10.6028/NIST.IR.8413-upd1.
23. **Alagic G., Alperin-Sheriff J., Apon D. C.** [et al.]. Status report on the second round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2020. DOI: 10.6028/NIST.IR.8309.
24. **Albrecht M. R., Bernstein D. J., Chou T.** [et al.]. Classic McEliece // Post-quantum cryptography. Round 3 submissions. Gaithersburg, MD: NIST, 2020. URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions (accessed: 6.03.2026).
25. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece // Post-quantum cryptography. Round 4 submissions. Gaithersburg, MD: NIST, 2022. URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-4-submissions (accessed: 6.03.2026).
26. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece. Implementation. Chicago: Univ. Ill. Chic., 2022. URL: classic.mceliece.org/impl.html (accessed: 6.03.2026).
27. CIRCL: Cloudflare interoperable reusable cryptographic library. San Francisco: Cloudflare, 2023. URL: github.com/cloudflare/circl (accessed: 6.03.2026).
28. Implement Classic McEliece. San Francisco: Cloudflare, 2022. URL: github.com/cloudflare/circl/pull/378 (accessed: 6.03.2026).
29. Open Quantum Safe liboqs: C library for prototyping and experimenting with quantum-resistant cryptography. 2025. URL: github.com/open-quantum-safe/liboqs/tree/main/src/kem/classic_mceliece (accessed: 6.03.2026).
30. **Wiggers T., Stebila D.** Clean, portable, tested implementations of post-quantum cryptography. 2023. URL: github.com/PQClean/PQClean (accessed: 6.03.2026).
31. **Hülsing A., Ning K.-C., Schwabe P., Weber F., Zimmermann P. R.** Post-quantum WireGuard // Proc. 42nd IEEE Symp. Security and Privacy (San Francisco, USA, May24–27, 2021). Los Alamitos, CA: IEEE Comput. Soc., 2021. P. 304–321. DOI: 10.1109/SP40001.2021.00030.
32. Software co-design acceleration of Classic McEliece key encapsulation mechanism. 2021. URL: github.com/beatsnbytes/classic_mceliece (accessed: 6.03.2026).

33. Discrete math final project for 2018 — Implementation of the McEliece cryptosystem. 2018. URL: github.com/arpanrau/McEliece-Implementation (accessed: 6.03.2026).

34. **Nießen T.** Purely educational PoC design and implementation of a PQC key exchange using Classic McEliece. 2019. URL: github.com/tniessen/node-mceliece-key-exchange-poc (accessed: 6.03.2026).

35. **Bernstein D. J.** The McEliece cryptosystem // Talks 1st Post-Quantum Cryptography Summer School in Universities (Chengdu, China, July 17, 2024). 76 p. URL: cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf (accessed: 6.03.2026).

36. ГОСТ 34.11—2018. Информационная технология. Криптографическая защита информации. Функция хэширования. Введ. 01.06.2019. М.: Стандартинформ, 2018. 25 с.

37. **Vysotskaya V. V., Chizhov I. V.** Design criteria of a new code-based KEM // J. Comput. Virol. Hacking Tech. 2024. V. 20, No. 3. P. 497–511. DOI: 10.1007/s11416-024-00527-z.

38. **Ge J., Liao H., Xue R.** Measure-rewind-extract: Tighter proofs of one-way to hiding and CCA security in the quantum random oracle model // Advances in cryptology — ASIACRYPT 2024. Proc. 30th Int. Conf. Theory and Application of Cryptology and Information Security (Kolkata, India, Dec. 9–13, 2024). Pt. IV. Singapore: Springer, 2024. P. 3–34. (Lect. Notes Comput. Sci.; V. 15487). DOI: 10.1007/978-981-96-0894-2_1.

39. **Stern J.** A new identification scheme based on syndrome decoding // Advances in cryptology — CRYPTO'93. Proc. 13th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 22–26, 1993). Heidelberg: Springer, 1994. P. 13–21. (Lect. Notes Comput. Sci.; V. 773). DOI: 10.1007/3-540-48329-2_2.

40. **Vysotskaya V. V., Chizhov I. V.** The security of the code-based signature scheme based on the Stern identification protocol // Прикл. дискрет. математика. 2022. № 57. С. 67–90. DOI: 10.17223/20710410/57/5.

41. **Царегородцев К. Д.** Троичная лемма о разветвлении и её приложение к анализу стойкости одной кодовой схемы подписи // Прикл. дискрет. математика. 2023. № 59. С. 58–71. DOI: 10.17223/20710410/59/3.

42. **Высоцкая В. В., Дас Д. К.** Анализ устойчивости постквантовой электронной подписи «Шиповник» к атакам, нацеленным на хэш-функции // Докл. XXVI Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 19–22 марта 2024 г.). М.: РусКрипто, 2024. 36 р. URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_das.pdf (дата обращения: 6.03.2026).

43. Открытая реализация алгоритма электронной цифровой подписи «Шиповник» для ТК26. М.: QApp, 2023. URL: github.com/QAPP-tech/shipovnik_tc26 (дата обращения: 6.03.2026).

44. **Prange E.** The use of information sets in decoding cyclic codes // IRE Trans. Inf. Theory. 1962. V. 8, No. 5. P. 5–9. DOI: 10.1109/TIT.1962.1057777.

45. Sendrier N. Decoding one out of many // Post-quantum cryptography. Proc. 4th Int. Workshop (Taipei, China, Nov. 29 – Dec. 2, 2011). Heidelberg: Springer, 2011. P. 51–67. (Lect. Notes Comput. Sci.; V. 7071). DOI: 10.1007/978-3-642-25405-5_4.

46. Löndahl C., Johansson T., Shooshtari M. K., Ahmadian-Attari M., Aref M. R. Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension // Des. Codes Cryptogr. 2016. V. 80, No. 2. P. 359–377. DOI: 10.1007/s10623-015-0099-x.

47. Guo Q., Johansson T., Löndahl C. A new algorithm for solving ring-lpn with a reducible polynomial // IEEE Trans. Inf. Theory. 2015. V. 61, No. 11. P. 6204–6212. DOI: 10.1109/TIT.2015.2475738.

48. Guo Q., Johansson T., Stankovski P. A key recovery attack on MDPC with CCA security using decoding errors // Advances in cryptology — ASIACRYPT 2016. Proc. 22nd Int. Conf. Theory and Application of Cryptology and Information Security (Hanoi, Vietnam, Dec. 4–8, 2016). Pt. I. Heidelberg: Springer, 2016. P. 789–815. (Lect. Notes Comput. Sci.; V. 10031). DOI: 10.1007/978-3-662-53887-6_29.

49. Nilsson A., Johansson T., Wagner P. S. Error amplification in code-based cryptography // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019. V. 2019, No. 1. P. 238–258. DOI: 10.46586/tches.v2019.i1.238-258.

50. Aguilar-Melchor C., Blazy O., Deneuville J.-C. [et al.]. Efficient encryption from random quasi-cyclic codes // IEEE Trans. Inf. Theory. 2018. V. 64, No. 5. P. 3927–3943. DOI: 10.1109/TIT.2018.2804444.

51. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC cryptosystem specification. Gaithersburg, MD: NIST, 2025. URL: pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (accessed: 6.03.2026).

52. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC. NIST submission packages. 2025. URL: pqc-hqc.org/doc/archive_submissions.zip (accessed: 6.03.2026).

53. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC. Optimized implementation. 2024. URL: web.archive.org/web/20250712014511/pqc-hqc.org/doc/hqc-optimized-implementation_2024-10-30.zip (accessed: 6.03.2026).

54. Aragon N., Barreto P., Bettaieb S. [et al.]. BIKE cryptosystem specification. Gaithersburg, MD: NIST, 2024. URL: bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf (accessed: 6.03.2026).

55. Misoczki R., Tillich J.-P., Sendrier N. [et al.]. MDPC-McEliece: New McEliece variants from moderate density parity-check codes // Proc. 2013 IEEE Int. Symp. Information Theory (Istanbul, Turkey, July 7–12, 2013). Piscataway: IEEE, 2013. P. 2069–2073. DOI: 10.1109/ISIT.2013.6620590.

56. Aragon N., Barreto P., Bettaieb S. [et al.]. BIKE. Reference implementation. 2024. URL: bikesuite.org/reference.html (accessed: 6.03.2026).

57. Additional implementation of BIKE. Seattle: AWS Labs, 2024. URL: github.com/awslabs/bike-kem (accessed: 6.03.2026).

58. **Mangard S., Oswald E., Popp T.** Power analysis attacks: Revealing the secrets of smart cards. New York: Springer, 2007. 338 p. DOI: 10.1007/978-0-387-38162-6.
59. **Standaert F. X.** Introduction to side-channel attacks // Secure integrated circuits and systems. New York: Springer, 2010. P. 27–42. DOI: 10.1007/978-0-387-71829-3_2.
60. **Peeters E., Standaert F. X., Quisquater J. J.** Power and electromagnetic analysis: Improved model, consequences and comparisons // Integration. 2007. V. 40, No. 1. P. 52–60. DOI: 10.1016/j.vlsi.2005.12.013.
61. **Standaert F. X., Mace F., Peeters E.** [et al.]. Updates on the security of FPGAs against power analysis attacks // Reconfigurable computing: Architectures and applications. Rev. Sel. Pap. 2nd Int. Workshop (Delft, The Netherlands, Mar. 1–3, 2006). Heidelberg: Springer, 2006. P. 335–346. (Lect. Notes Comput. Sci.; V. 3985). DOI: 10.1007/11802839_42.
62. **Жуков А. Е.** Криптоанализ по побочным каналам (side channel attacks) // Защита информации. Инсайд. 2010. № 5. С. 28–33.
63. **Chari S., Rao J. R., Rohatgi P.** Template attacks // Cryptographic hardware and embedded systems — CHES 2002. Rev. Pap. 4th Int. Workshop (Redwood Shores, CA, USA, Aug. 13–15, 2002). Heidelberg: Springer, 2003. P. 13–28. (Lect. Notes Comput. Sci.; V. 2523). DOI: 10.1007/3-540-36400-5_3.
64. **Anderson R., Kuhn M.** Tamper resistance — A cautionary note // Proc. 2nd USENIX Workshop Electronic Commerce (Oakland, CA, USA, Nov. 18–21, 1996). Pittsburgh, PA: Carnegie Mellon Univ., 1996. P. 1–11.
65. **Tuyls P., Schrijen G.-J., Škorić B.** [et al.]. Read-proof hardware from protective coatings // Cryptographic hardware and embedded systems — CHES 2006. Proc. 8th Int. Workshop (Yokohama, Japan, Oct. 10–13, 2006). Heidelberg: Springer, 2006. P. 369–383. (Lect. Notes Comput. Sci.; V. 4249). DOI: 10.1007/11894063_29.
66. **Shamir A.** Protecting smart cards from passive power analysis with detached power supplies // Cryptographic hardware and embedded systems — CHES 2000. Proc. 2nd Int. Workshop (Worcester, MA, USA, Aug. 17–18, 2000). Heidelberg: Springer, 2000. P. 71–77. (Lect. Notes Comput. Sci.; V. 1965). DOI: 10.1007/3-540-44499-8_5.
67. **Goubin L., Patarin J.** DES and differential power analysis. The "Duplication" method // Cryptographic hardware and embedded systems. Proc. 1st Int. Workshop (Worcester, MA, USA, Aug. 12–13, 1999). Heidelberg: Springer, 1999. P. 158–172. (Lect. Notes Comput. Sci.; V. 1717). DOI: 10.1007/3-540-48059-5_15.
68. **May D., Muller H. L., Smart N. P.** Random register renaming to foil DPA // Cryptographic hardware and embedded systems — CHES 2001. Proc. 3rd Int. Workshop (Paris, France, May 14–16, 2001). Heidelberg: Springer, 2001. P. 28–38. (Lect. Notes Comput. Sci.; V. 2162). DOI: 10.1007/3-540-44709-1_4.

69. Chari S., Jutla S. C., Rao R. J. [et al.]. Towards sound approaches to counteract power-analysis attacks // Advances in cryptology — CRYPTO'99. Proc. 19th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 15–19, 1999). Heidelberg: Springer, 1999. P. 398–412. (Lect. Notes Comput. Sci.; V. 1666). DOI: 10.1007/3-540-48405-1_26.

70. Ueno R., Homma N., Aoki T. Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation // Constructive side-channel analysis and secure design. Rev. Sel. Pap. 8th Int. Workshop (Paris, France, Apr. 13–14, 2017). Cham: Springer, 2017. P. 50–64. (Lect. Notes Comput. Sci.; V. 10348). DOI: 10.1007/978-3-319-64647-3_4.

71. Schwabe P., Stoffelen K. All the AES you need on Cortex-M3 and M4 // Selected areas in cryptography — SAC 2016. Rev. Sel. Pap. 23rd Int. Conf. (St. John's, NL, Canada, Aug. 10–12, 2016). Cham: Springer, 2016. P. 180–194. (Lect. Notes Comput. Sci.; V. 10532). DOI: 10.1007/978-3-319-69453-5_10.

72. Tiri K., Akmal M., Verbauwhede I. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards // Proc. 28th European Solid-State Circuits Conf. (Florence, Italy, Sept. 24–26, 2002). Piscataway: IEEE, 2002. P. 403–406.

73. Patterson N. The algebraic decoding of Goppa codes // IEEE Trans. Inf. Theory. 1975. V. 21, No. 2. P. 203–207. DOI: 10.1109/TIT.1975.1055350.

74. Strenzke F., Tews E., Molter G. [et al.]. Side channels in the McEliece PKC // Post-quantum cryptography. Proc. 2nd Int. Workshop (Cincinnati, OH, USA, Oct. 17–19, 2008). Heidelberg: Springer, 2008. P. 216–229. (Lect. Notes Comput. Sci.; V. 5299). DOI: 10.1007/978-3-540-88403-3_15.

75. Avanzi R., Hoerder S., Page D. [et al.]. Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems // J. Cryptogr. Eng. 2011. V. 1, No. 4. P. 271–281. DOI: 10.1007/s13389-011-0024-9.

76. Shoufan A., Strenzke F., Molter H. G. [et al.]. A timing attack against Patterson algorithm in the McEliece PKC // Information security and cryptology — ICISC 2009. Rev. Sel. Pap. 12th Int. Conf. (Seoul, Korea, Dec. 2–4, 2009). Heidelberg: Springer, 2010. P. 161–175. (Lect. Notes Comput. Sci.; V. 5984). DOI: 10.1007/978-3-642-14423-3_12.

77. Strenzke F. A timing attack against the secret permutation in the McEliece PKC // Post-quantum cryptography. Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010). Heidelberg: Springer, 2010. P. 95–107. (Lect. Notes Comput. Sci.; V. 6061). DOI: 10.1007/978-3-642-12929-2_8.

78. Strenzke F. Timing attacks against the syndrome inversion in code-based cryptosystems // Post-quantum cryptography. Proc. 5th Int. Workshop (Limoges, France, June 4–7, 2013). Heidelberg: Springer, 2013. P. 217–230. (Lect. Notes Comput. Sci.; V. 7932). DOI: 10.1007/978-3-642-38616-9_15.

79. **Heyse S., Moradi A., Paar C.** Practical power analysis attacks on software implementations of McEliece // Post-quantum cryptography. Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010). Heidelberg: Springer, 2010. P. 108–125. (Lect. Notes Comput. Sci.; V. 6061). DOI: 10.1007/978-3-642-12929-2_9.
80. **Molter H. G., Stöttinger M., Shoufan A.** [et al.]. A simple power analysis attack on a McEliece cryptoprocessor // J. Cryptogr. Eng. 2011. V. 1, No. 1. P. 29–36. DOI: 10.1007/s13389-011-0001-3.
81. **Shoufan A., Wink T., Molter H. G.** [et al.]. A novel processor architecture for McEliece cryptosystem and FPGA platforms // Proc. 20th IEEE Int. Conf. Application-Specific Systems, Architectures and Processors (Boston, MA, USA, July 7–9, 2009). Los Alamitos, CA: IEEE Comput. Soc., 2009. P. 98–105. DOI: 10.1109/ASAP.2009.29.
82. **Heyse S., Von Maurich I., Güneysu T.** Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices // Cryptographic hardware and embedded systems — CHES 2013. Proc. 15th Int. Workshop (Santa Barbara, USA, Aug. 20–23, 2013). Heidelberg: Springer, 2013. P. 273–292. (Lect. Notes Comput. Sci.; V. 8086). DOI: 10.1007/978-3-642-40349-1_16.
83. **Von Maurich I., Güneysu T.** Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices // Post-quantum cryptography. Proc. 6th Int. Workshop (Waterloo, ON, Canada, Oct. 1–3, 2014). Cham: Springer, 2014. P. 266–282. (Lect. Notes Comput. Sci.; V. 8772). DOI: 10.1007/978-3-319-11659-4_16.
84. **Von Maurich I., Güneysu T.** Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices // Proc. 2014 Design, Automation and Test in Europe Conf. (Dresden, Germany, Mar. 24–28, 2014). Piscataway: IEEE, 2014. P. 1–6. DOI: 10.7873/DATE.2014.051.
85. **Chen C., Eisenbarth T., Von Maurich I.** [et al.]. Differential power analysis of a McEliece cryptosystem // Applied cryptography and network security. Rev. Sel. Pap. 13th Int. Conf. (New York, USA, June 2–5, 2015). Cham: Springer, 2015. P. 538–556. (Lect. Notes Comput. Sci.; V. 9092). DOI: 10.1007/978-3-319-28166-7_26.
86. **Schamberger T., Renner J., Sigl G.** [et al.]. A power side-channel attack on the CCA2-secure HQC KEM // Smart card research and advanced applications. Rev. Sel. Pap. 19th Int. Conf. (Lübeck, Germany, Nov. 18–19, 2020). Cham: Springer, 2020. P. 119–134. (Lect. Notes Comput. Sci.; V. 12609). DOI: 10.1007/978-3-030-68487-7_8.
87. **Hlauschek C., Lahr N., Schröder R. L.** On the timing leakage of the deterministic re-encryption in HQC KEM. San Diego, 2021. 24 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2021/1485/20211115:124514). URL: eprint.iacr.org/archive/2021/1485/20211115:124514 (accessed: 6.03.2026).

88. **Wafo-Tapa G., Bettaieb S., Bidoux L.** [et al.]. A practicable timing attack against HQC and its countermeasure // Adv. Math. Commun. 2022. V. 16, No. 3. P. 621–642. DOI: 10.3934/amc.2020126.
89. **Paiva T. B., Terada R.** A timing attack on the HQC encryption scheme // Selected areas in cryptography — SAC 2019. Rev. Sel. Pap. 26th Int. Conf. (Waterloo, ON, Canada, Aug. 12–16, 2019). Cham: Springer, 2019. P. 551–573. (Lect. Notes Comput. Sci.; V. 11959). DOI: 10.1007/978-3-030-38471-5_22.
90. **Lahr N., Niederhagen R., Petri R.** [et al.]. Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation // Advances in cryptology — ASIACRYPT 2020. Proc. 26th Int. Conf. Theory and Application of Cryptology and Information Security (Daejeon, South Korea, Dec. 7–11, 2020). Pt. I. Cham: Springer, 2020. P. 881–910. (Lect. Notes Comput. Sci.; V. 12491). DOI: 10.1007/978-3-030-64837-4_29.
91. **Guo Q., Johansson A., Johansson T.** A key-recovery side-channel attack on Classic McEliece implementations // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 4. P. 800–827. DOI: 10.46586/tches.v2022.i4.800-827.
92. **Pircher S., Geier J., Danner J.** [et al.]. Key-recovery fault injection attack on the Classic McEliece KEM // Code-based cryptography. Rev. Sel. Pap. 10th Int. Workshop (Trondheim, Norway, May 29–30, 2022). Cham: Springer, 2022. P. 37–61. (Lect. Notes Comput. Sci.; V. 13839). DOI: 10.1007/978-3-031-29689-5_3.
93. **Grosso V., Cayrel P.-L., Colombier B.** [et al.]. Punctured syndrome decoding problem: Efficient side-channel attacks against Classic McEliece // Constructive side-channel analysis and secure design. Proc. 14th Int. Workshop (Munich, Germany, Apr. 3–4, 2023). Cham: Springer, 2023. P. 170–192. (Lect. Notes Comput. Sci.; V. 13979). DOI: 10.1007/978-3-031-29497-6_9.
94. **Brinkmann M., Chuengsatiansup C., May A.** [et al.]. Leaky McEliece: Secret key recovery from highly erroneous side-channel information // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2025. V. 2025, No. 2. P. 94–125. DOI: 10.46586/tches.v2025.i2.94-125.
95. **Bitzer S., Delvaux J., Kirshanova E.** [et al.]. How to lose some weight: A practical template syndrome decoding attack // Des. Codes Cryptogr. 2025. V. 93, No. 7. P. 2503–2519. DOI: 10.1007/s10623-025-01603-1.
96. **Drăgoi V.-F., Colombier B., Vallet N.** [et al.]. Full key-recovery cubic-time template attack on Classic McEliece decapsulation. San Diego, 2024. 25 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2024/1694). URL: eprint.iacr.org/2024/1694 (accessed: 6.03.2026).
97. **Ueno R., Xagawa K., Tanaka Y.** [et al.]. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 1. P. 296–322. DOI: 10.46586/tches.v2022.i1.296-322.

98. Bernstein D. J., Lange T., Peters C. Attacking and defending the McEliece cryptosystem // Post-quantum cryptography. Proc. 2nd Int. Workshop (Cincinnati, OH, USA Oct. 17–19, 2008). Heidelberg: Springer, 2008. P. 31–46. (Lect. Notes Comput. Sci.; V. 5299). DOI: 10.1007/978-3-540-88403-3_3.

99. Canteaut A., Chabaud F. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511 // IEEE Trans. Inf. Theory. 1998. V. 44, No. 1. P. 367–378. DOI: 10.1109/18.651067.

100. Canteaut A., Sendrier N. Cryptanalysis of the original McEliece cryptosystem // Advances in cryptology — ASIACRYPT'98. Proc. Int. Conf. Theory and Application of Cryptology and Information Security (Beijing, China, Oct. 18–22, 1998). Heidelberg: Springer, 1998. P. 187–199. (Lect. Notes Comput. Sci.; V. 1514). DOI: 10.1007/3-540-49649-1_16.

101. Eaton E., Lequesne M., Parent A. [et al.]. QC-MDPC: A timing attack and a CCA2 KEM // Post-quantum cryptography. Proc. 9th Int. Conf. (Fort Lauderdale, FL, USA, Apr. 9–11, 2018). Cham: Springer, 2018. P. 47–76. (Lect. Notes Comput. Sci.; V. 10786). DOI: 10.1007/978-3-319-79063-3_3.

102. Drucker N., Gueron S., Kostic D. QC-MDPC decoders with several shades of gray // Post-quantum cryptography. Proc. 11th Int. Conf. (Paris, France, Apr. 15–17, 2020). Cham: Springer, 2020. P. 35–50. (Lect. Notes Comput. Sci.; V. 12100). DOI: 10.1007/978-3-030-44223-1_3.

103. Schamberger T., Holzbaur L., Renner J. [et al.]. A power side-channel attack on the Reed–Muller Reed–Solomon version of the HQC cryptosystem // Post-quantum cryptography. Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28-30, 2022). Cham: Springer, 2022. P. 327–352. (Lect. Notes Comput. Sci.; V. 13512). DOI: 10.1007/978-3-031-17234-2_16.

104. Goy G., Loiseau A., Gaborit P. A new key recovery side-channel attack on HQC with chosen ciphertext // Post-quantum cryptography. Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28–30, 2022). Cham: Springer, 2022. P. 353–371. (Lect. Notes Comput. Sci.; V. 13512). DOI: 10.1007/978-3-031-17234-2_17.

105. Chou T. QcBits: Constant-time small-key code-based cryptography // Cryptographic hardware and embedded systems — CHES 2016. Proc. 18th Int. Conf. (Santa Barbara, USA, Aug. 17–19, 2016). Heidelberg: Springer, 2016. P. 280–300. (Lect. Notes Comput. Sci.; V. 9813). DOI: 10.1007/978-3-662-53140-2_14.

106. Rossi M., Hamburg M., Hutter M. [et al.]. A side-channel assisted cryptanalytic attack against QcBits // Cryptographic hardware and embedded systems — CHES 2017. Proc. 19th Int. Conf. (Taipei, China, Sept. 25–28, 2017). Cham: Springer, 2017. P. 3–23. (Lect. Notes Comput. Sci.; V. 10529). DOI: 10.1007/978-3-319-66787-4_1.

107. Sim B.-Y., Kwon J., Choi K. Y. [et al.]. Novel side-channel attacks on quasi-cyclic code-based cryptography // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019. V. 2019, No. 1. P. 180–212. DOI: 10.46586/tches.v2019.i4.180-212.

108. Guo Q., Hlauschek C., Johansson T. [et al.]. Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 3. P. 223–263. DOI: 10.46586/tches.v2022.i3.223-263.

109. Sendrier N. Secure sampling of constant-weight words — Application to BIKE. San Diego, 2021. 16 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2021/1631). URL: eprint.iacr.org/2021/1631 (accessed: 6.03.2026).

110. Huang S., Sim Q. R., Chuengsatiansup C. [et al.]. Cache-timing attack against HQC. San Diego, 2023. 34 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2023/102). URL: eprint.iacr.org/2023/102 (accessed: 6.03.2026).

111. Goy G., Maillard J., Gaborit P. [et al.]. Single trace HQC shared key recovery with SASCA // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2024. V. 2024, No. 2. P. 64–87. DOI: 10.46586/tches.v2024.i2.64-87.

Бахарев Александр Олегович
Воронов Денис Максимович
Коломеец Николай Александрович
Токарева Наталья Николаевна
Хильчук Ирина Сергеевна
Шапоренко Александр Сергеевич

Статья поступила
18 июня 2025 г.
После доработки —
11 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 5–68

UDC 519.7 DOI: 10.33048/daio.2025.32.837

SIDE-CHANNEL ATTACKS ON CODE-BASED POST-QUANTUM CRYPTOGRAPHIC SYSTEMS: A SURVEY. PART 1

A. O. Bakharev[1, 2, a], *D. M. Voronov*[1, 2, b], *N. A. Kolomeec*[1, 2, c], *N. N. Tokareva*[1, 2, 3, d], *I. S. Khilchuk*[1, 2, e], and *A. S. Shaporenko*[1, 2, f]

[1] National Technology Center for Digital Cryptography,
1 Ramensky Boulevard, 119192 Moscow, Russia
[2] Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
[3] Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: [a] a.bakharev@g.nsu.ru, [b] d.voronov2@g.nsu.ru,
[c] n.kolomeets@g.nsu.ru, [d] crypto1127@mail.ru,
[e] i.khilchuk@g.nsu.ru, [f] a.shaporenko@g.nsu.ru

Abstract. This work of two parts provides a structured analytical review devoted to side-channel attacks on post-quantum code-based cryptosystems. The first part of the review presents a description of the main cryptographic primitives and algorithms used in code-based cryptosystems, as well as description of the most significant modern code-based cryptosystems: Classic McEliece, Codiaeum, Shipovnik, BIKE, and HQC. This survey is carried out within the scientific and research project «Kulminatsiya» of the National Technology Center for Digital Cryptography. Tab. 5, illustr. 14, bibliogr. 111.

Keywords: post-quantum cryptography, side-channel attack, code-based cryptographic system.

References

1. **P. W. Shor,** Algorithms for quantum computation: Discrete logarithms and factoring, in *Proc. 35th Annu. Symp. Foundations of Computer Science* (Santa Fe, USA, Nov. 20–22, 1994) (IEEE Comput. Soc., Los Alamitos, CA, 1994), pp. 124–134, DOI: 10.1109/SFCS.1994.365700.
2. **H. Niederreiter,** Knapsack-type cryptosystems and algebraic coding theory, *Prob. Control Inf. Theory* **15** (2), 157–166 (1986).

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

3. **R. J. McEliece,** A public-key cryptosystem based on algebraic coding theory, *DSN Progress Rep.* **42–44**, 114–116 (1978).
4. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece. Specification (Univ. Ill. Chic., Chicago, 2022), URL: `classic.mceliece.org/spec.html` (accessed: 6.03.2026).
5. **V. V. Vysotskaya** and **I. V. Chizhov,** Post-quantum key encapsulation mechanism "Kodieum", in *Dokl. XXVI Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 19–22, 2024) (RusCrypto, Moscow, 2024) [Russian], URL: `ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_chizhov.pdf` (accessed: 6.03.2026).
6. **V. V. Vysotskaya** and **I. V. Chizhov,** Post-quantum signature scheme based on the Stern identification protocol, in *Dokl. XXIII Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 23–26, 2021) (RusCrypto, Moscow, 2021) [Russian], URL: `ruscrypto.ru/resource/archive/rc2021/files/02_vysotskaya_chizhov.pdf` (accessed: 6.03.2026).
7. **E. Berlekamp, R. McEliece,** and **H. Van Tilborg,** On the inherent intractability of certain coding problems (corresp.), *IEEE Trans. Inf. Theory* **24** (3), 384–386 (1978), DOI: `10.1109/TIT.1978.1055873`.
8. **V. D. Goppa,** A rational representation of codes and (L, g)-codes, *Probl. Peredachi Inf.* **7** (3), 41–49 (1971) [Russian] [*Probl. Inf. Transm.* **7** (3), 223–229 (1971)].
9. **D. Hofheinz, K. Hövelmanns,** and **E. Kiltz,** A modular analysis of the Fujisaki–Okamoto transformation, in *Theory of Cryptography*, Proc. 15th Int. Conf. (Baltimore, MD, USA, Nov. 12–15, 2017), Pt. I (Springer, Cham, 2017), pp. 341–371 (Lect. Notes Comput. Sci., Vol. 10677), DOI: `10.1007/978-3-319-70500-2_12`.
10. **N. Bindel, M. Hamburg, K. Hövelmanns,** [et al.], Tighter proofs of CCA security in the quantum random oracle model, in *Theory of Cryptography*, Proc. 17th Int. Conf. (Nuremberg, Germany, Dec. 1–5, 2019), Pt. II (Springer, Cham, 2019), pp. 61–90 (Lect. Notes Comput. Sci., Vol. 11892), DOI: `10.1007/978-3-030-36033-7_3`.
11. **A. Fiat** and **A. Shamir,** How to prove yourself: Practical solutions to identification and signature problems, in *Advances in Cryptology — CRYPTO'86*, Proc. Conf. Theory and Applications of Cryptographic Techniques (Santa Barbara, USA, Aug. 11–15, 1986) (Springer, Heidelberg, 1987), pp. 186–194 (Lect. Notes Comput. Sci., Vol. 263), DOI: `10.1007/3-540-47721-7_12`.
12. **S. Gao** and **T. Mateer,** Additive fast Fourier transforms over finite fields, *IEEE Trans. Inf. Theory* **56** (12), 6265–6272 (2010).
13. **N. Sendrier,** Finding the permutation between equivalent linear codes: The support splitting algorithm, *IEEE Trans. Inf. Theory* **46** (4), 1193–1203 (2000).
14. **J. Stern,** A method for finding codewords of small weight, in *Coding Theory and Applications*, Proc. 3rd Int. Colloq. (Toulon, France, Nov. 2–4, 1988) (Springer, Heidelberg, 1988), pp. 106–113 (Lect. Notes Comput. Sci., Vol. 388), DOI: `10.1007/BFb0019850`.

15. **V. M. Sidelnikov** and **S. O. Shestakov,** On insecurity of cryptosystems based on generalized Reed–Solomon codes, *Diskretn. Mat.* **4** (3), 57–63 (1992) [Russian] [*Discrete Math. Appl.* **2** (4), 439–444 (1992), DOI: 10.1515/dma.1992.2.4.439].
16. **V. V. Davydov, V. V. Beliaev, E. F. Kustov,** [et al.], Modern variations of McEliece and Niederreiter cryptosystems, *J. Sci. Tech. Inf. Technol. Mech. Opt.* **22** (2), 324–331 (2022), DOI: 10.17586/2226-1494-2022-22-2-324-331.
17. **V. M. Sidelnikov,** A public-key cryptosystem based on binary Reed–Muller codes, *Diskretn. Mat.* **6** (2), 3–20 (1994) [Russian] [*Discrete Math. Appl.* **4** (3), 191–207 (1994), DOI: 10.1515/dma.1994.4.3.191].
18. **L. Minder** and **A. Shokrollahi,** Cryptanalysis of the Sidelnikov cryptosystem, in *Advances in Cryptology — EUROCRYPT 2007*, Proc. 26th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques (Barcelona, Spain, May 20–24, 2007) (Springer, Heidelberg, 2007), pp. 347–360 (Lect. Notes Comput. Sci., Vol. 4515), DOI: 10.1007/978-3-540-72540-4_20.
19. **R. Overbeck** and **N. Sendrier,** Code-based cryptography, in *Post-Quantum Cryptography* (Springer, Heidelberg, 2009), pp. 95–145, DOI: 10.1007/978-3-540-88702-7_4.
20. **M. A. González de la Torre, L. Hernández Encinas,** and **J. I. Sánchez García,** Structural analysis of code-based algorithms of the NIST post-quantum call, *Logic J. IGPL* **33** (5), ID jzae071 (2024), DOI: 10.1093/jigpal/jzae071.
21. **G. Alagic, M. Bros, P. Ciadoux,** [et al.], Status report on the fourth round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2025), DOI: 10.6028/NIST.IR.8545.
22. **G. Alagic, D. C. Apon, D. Cooper,** [et al.], Status report on the third round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2022), DOI: 10.6028/NIST.IR.8413-upd1.
23. **G. Alagic, J. Alperin-Sheriff, D. C. Apon,** [et al.], Status report on the second round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2020), DOI: 10.6028/NIST.IR.8309.
24. **M. R. Albrecht, D. J. Bernstein, T. Chou,** [et al.], Classic McEliece, in *Post-Quantum Cryptography. Round 3 Submissions* (NIST, Gaithersburg, MD, 2020), URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions (accessed: 6.03.2026).
25. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece, in *Post-Quantum Cryptography. Round 4 Submissions* (NIST, Gaithersburg, MD, 2022), URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-4-submissions (accessed: 6.03.2026).
26. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece. Implementation (Univ. Ill. Chic., Chicago, 2022), URL: classic.mceliece.org/impl.html (accessed: 6.03.2026).

27. CIRCL: Cloudflare interoperable reusable cryptographic library (Cloudflare, San Francisco, 2023), URL: `github.com/cloudflare/circl` (accessed: 6.03.2026).

28. Implement Classic McEliece (Cloudflare, San Francisco, 2022), URL: `github.com/cloudflare/circl/pull/378` (accessed: 6.03.2026).

29. Open Quantum Safe liboqs: C library for prototyping and experimenting with quantum-resistant cryptography, 2025, URL: `github.com/open-quantum-safe/liboqs/tree/main/src/kem/classic_mceliece` (accessed: 6.03.2026).

30. **T. Wiggers** and **D. Stebila,** Clean, portable, tested implementations of post-quantum cryptography, 2023, URL: `github.com/PQClean/PQClean` (accessed: 6.03.2026).

31. **A. Hülsing, K.-C. Ning, P. Schwabe, F. Weber,** and **P. R. Zimmermann,** Post-quantum WireGuard, in *Proc. 42nd IEEE Symp. Security and Privacy* (San Francisco, USA, May24–27, 2021) (IEEE Comput. Soc., Los Alamitos, CA, 2021), pp. 304–321, DOI: `10.1109/SP40001.2021.00030`.

32. Software co-design acceleration of Classic McEliece key encapsulation mechanism, 2021, URL: `github.com/beatsnbytes/classic_mceliece` (accessed: 6.03.2026).

33. Discrete math final project for 2018 — Implementation of the McEliece cryptosystem, 2018, URL: `github.com/arpanrau/McEliece-Implementation` (accessed: 6.03.2026).

34. **T. Nießen,** Purely educational PoC design and implementation of a PQC key exchange using Classic McEliece, 2019, URL: `github.com/tniessen/node-mceliece-key-exchange-poc` (accessed: 6.03.2026).

35. **D. J. Bernstein,** The McEliece cryptosystem, in *Talks 1st Post-Quantum Cryptography Summer School in Universities* (Chengdu, China, July 17, 2024), URL: `cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf` (accessed: 6.03.2026).

36. Information technology. Cryptographic data security. Hash function, *GOST R 34.11—2018* (Standartinform, Moscow, 2018) [Russian].

37. **V. V. Vysotskaya** and **I. V. Chizhov,** Design criteria of a new code-based KEM, *J. Comput. Virol. Hacking Tech.* **20** (3), 497–511 (2024), DOI: `10.1007/s11416-024-00527-z`.

38. **J. Ge, H. Liao,** and **R. Xue,** Measure-rewind-extract: Tighter proofs of one-way to hiding and CCA security in the quantum random oracle model, in *Advances in Cryptology — ASIACRYPT 2024*, Proc. 30th Int. Conf. Theory and Application of Cryptology and Information Security (Kolkata, India, Dec. 9–13, 2024), Pt. IV (Springer, Singapore, 2024), pp. 3–34 (Lect. Notes Comput. Sci., Vol. 15487), DOI: `10.1007/978-981-96-0894-2_1`.

39. **J. Stern,** A new identification scheme based on syndrome decoding, in *Advances in Cryptology — CRYPTO'93*, Proc. 13th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 22–26, 1993) (Springer, Heidelberg, 1994), pp. 13–21 (Lect. Notes Comput. Sci., Vol. 773), DOI: `10.1007/3-540-48329-2_2`.

40. **V. V. Vysotskaya, Chizhov I. V.** The security of the code-based signature scheme based on the Stern identification protocol, *Prikl. Diskretn. Mat.*, No. 57, 67–90 (2022) [Russian], DOI: 10.17223/20710410/57/5.
41. **K. D. Tsaregorodtsev,** Ternary forking lemma and its application to the analysis of one code-based signature, *Prikl. Diskretn. Mat.*, No. 59, 58–71 (2023) [Russian], DOI: 10.17223/20710410/59/3.
42. **V. V. Vysotskaya** and **D. K. Das,** Analyzing the resistance of the post-quantum signature "Shipovnik" to attacks against hash functions, in *Dokl. XXVI Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 19–22, 2024) (RusCrypto, Moscow, 2024), URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_das.pdf (accessed: 6.03.2026).
43. A public implementation of the signature algorithm "Shipovnik" for TK26 (QApp, Moscow, 2023) [Russian], URL: github.com/QAPP-tech/shipovnik_tc26 (accessed: 6.03.2026).
44. **E. Prange,** The use of information sets in decoding cyclic codes, *IRE Trans. Inf. Theory* **8** (5), 5–9 (1962), DOI: 10.1109/TIT.1962.1057777.
45. **N. Sendrier,** Decoding one out of many, in *Post-Quantum Cryptography*, Proc. 4th Int. Workshop (Taipei, China, Nov. 29 – Dec. 2, 2011) (Springer, Heidelberg, 2011), pp. 51–67 (Lect. Notes Comput. Sci., Vol. 7071), DOI: 10.1007/978-3-642-25405-5_4.
46. **C. Löndahl, T. Johansson, M. K. Shooshtari, M. Ahmadian-Attari,** and **M. R. Aref,** Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension, *Des. Codes Cryptogr.* **80** (2), 359–377 (2016), DOI: 10.1007/s10623-015-0099-x.
47. **Q. Guo, T. Johansson,** and **C. Löndahl,** A new algorithm for solving ring-lpn with a reducible polynomial, *IEEE Trans. Inf. Theory* **61** (11), 6204–6212 (2015), DOI: 10.1109/TIT.2015.2475738.
48. **Q. Guo, T. Johansson,** and **P. Stankovski,** A key recovery attack on MDPC with CCA security using decoding errors, in *Advances in Cryptology — ASIACRYPT 2016*, Proc. 22nd Int. Conf. Theory and Application of Cryptology and Information Security (Hanoi, Vietnam, Dec. 4–8, 2016), Pt. I (Springer, Heidelberg, 2016), pp. 789–815 (Lect. Notes Comput. Sci., Vol. 10031), DOI: 10.1007/978-3-662-53887-6_29.
49. **A. Nilsson, T. Johansson,** and **P. S. Wagner,** Error amplification in code-based cryptography, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019** (1), 238–258 (2019), DOI: 10.46586/tches.v2019.i1.238-258.
50. **C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville,** [et al.], Efficient encryption from random quasi-cyclic codes, *IEEE Trans. Inf. Theory* **64** (5), 3927–3943 (2018), DOI: 10.1109/TIT.2018.2804444.
51. **P. Gaborit, C. Aguilar-Melchor, N. Aragon,** [et al.], HQC cryptosystem specification (NIST, Gaithersburg, MD, 2025), URL: pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (accessed: 6.03.2026).
52. **P. Gaborit, C. Aguilar-Melchor, N. Aragon,** [et al.], HQC. NIST submission packages, 2025, URL: pqc-hqc.org/doc/archive_submissions.zip (accessed: 6.03.2026).

53. P. Gaborit, C. Aguilar-Melchor, N. Aragon, [et al.], HQC. Optimized implementation, 2024, URL: `web.archive.org/web/20250712014511/pqc-hqc.org/doc/hqc-optimized-implementation_2024-10-30.zip` (accessed: 6.03.2026).

54. N. Aragon, P. Barreto, S. Bettaieb, [et al.], BIKE cryptosystem specification (NIST, Gaithersburg, MD, 2024), URL: `bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf` (accessed: 6.03.2026).

55. R. Misoczki, J.-P. Tillich, N. Sendrier, [et al.], MDPC-McEliece: New McEliece variants from moderate density parity-check codes, in *Proc. 2013 IEEE Int. Symp. Information Theory* (Istanbul, Turkey, July 7–12, 2013) (IEEE, Piscataway, 2013), pp. 2069–2073, DOI: `10.1109/ISIT.2013.6620590`.

56. N. Aragon, P. Barreto, S. Bettaieb, [et al.], BIKE. Reference implementation, 2024, URL: `bikesuite.org/reference.html` (accessed: 6.03.2026).

57. Additional implementation of BIKE (AWS Labs, Seattle, 2024), URL: `github.com/awslabs/bike-kem` (accessed: 6.03.2026).

58. S. Mangard, E. Oswald, and **T. Popp,** *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Springer, New York, 2007), DOI: `10.1007/978-0-387-38162-6`.

59. F. X. Standaert, Introduction to side-channel attacks, in *Secure Integrated Circuits and Systems* (Springer, New York, 2010), pp. 27–42, DOI: `10.1007/978-0-387-71829-3_2`.

60. E. Peeters, F. X. Standaert, and **J. J. Quisquater,** Power and electromagnetic analysis: Improved model, consequences and comparisons, *Integration* **40** (1), 52–60 (2007), DOI: `10.1016/j.vlsi.2005.12.013`.

61. F. X. Standaert, F. Mace, E. Peeters, [et al.], Updates on the security of FPGAs against power analysis attacks, in *Reconfigurable Computing: Architectures and Applications*, Rev. Sel. Pap. 2nd Int. Workshop (Delft, The Netherlands, Mar. 1–3, 2006) (Springer, Heidelberg, 2006), pp. 335–346 (Lect. Notes Comput. Sci., Vol. 3985), DOI: `10.1007/11802839_42`.

62. A. E. Zhukov, Side channel attacks, *Inf. Secur., Inside*, No. 5, 28–33 (2010) [Russian].

63. S. Chari, J. R. Rao, and **P. Rohatgi,** Template attacks, in *Cryptographic Hardware and Embedded Systems — CHES 2002*, Rev. Pap. 4th Int. Workshop (Redwood Shores, CA, USA, Aug. 13–15, 2002) (Springer, Heidelberg, 2003), pp. 13–28 (Lect. Notes Comput. Sci., Vol. 2523), DOI: `10.1007/3-540-36400-5_3`.

64. R. Anderson and **M. Kuhn,** Tamper resistance — A cautionary note, in *Proc. 2nd USENIX Workshop Electronic Commerce* (Oakland, CA, USA, Nov. 18–21, 1996) (Carnegie Mellon Univ., Pittsburgh, PA, 1996), pp. 1–11.

65. P. Tuyls, G.-J. Schrijen, B. Škorić, [et al.], Read-proof hardware from protective coatings, in *Cryptographic Hardware and Embedded Systems — CHES 2006*, Proc. 8th Int. Workshop (Yokohama, Japan, Oct. 10–13, 2006) (Springer, Heidelberg, 2006), pp. 369–383 (Lect. Notes Comput. Sci., Vol. 4249), DOI: `10.1007/11894063_29`.

66. A. Shamir, Protecting smart cards from passive power analysis with detached power supplies, in *Cryptographic Hardware and Embedded Systems — CHES 2000*, Proc. 2nd Int. Workshop (Worcester, MA, USA, Aug. 17–18, 2000) (Springer, Heidelberg, 2000), pp. 71–77 (Lect. Notes Comput. Sci., Vol. 1965), DOI: 10.1007/3-540-44499-8_5.

67. L. Goubin and **J. Patarin,** DES and differential power analysis. The "Duplication" method, in *Cryptographic Hardware and Embedded Systems*, Proc. 1st Int. Workshop (Worcester, MA, USA, Aug. 12–13, 1999) (Springer, Heidelberg, 1999), pp. 158–172 (Lect. Notes Comput. Sci., Vol. 1717), DOI: 10.1007/3-540-48059-5_15.

68. D. May, H. L. Muller, and **N. P. Smart,** Random register renaming to foil DPA, in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Proc. 3rd Int. Workshop (Paris, France, May 14–16, 2001) (Springer, Heidelberg, 2001), pp. 28–38 (Lect. Notes Comput. Sci., Vol. 2162), DOI: 10.1007/3-540-44709-1_4.

69. S. Chari, S. C. Jutla, R. J. Rao, [et al.], Towards sound approaches to counteract power-analysis attacks, in *Advances in Cryptology — CRYPTO'99*, Proc. 19th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 15–19, 1999) (Springer, Heidelberg, 1999), pp. 398–412 (Lect. Notes Comput. Sci., Vol. 1666), DOI: 10.1007/3-540-48405-1_26.

70. R. Ueno, N. Homma, and **T. Aoki,** Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation, in *Constructive Side-Channel Analysis and Secure Design*, Rev. Sel. Pap. 8th Int. Workshop (Paris, France, Apr. 13–14, 2017) (Springer, Cham, 2017), pp. 50–64 (Lect. Notes Comput. Sci., Vol. 10348), DOI: 10.1007/978-3-319-64647-3_4.

71. P. Schwabe and **K. Stoffelen,** All the AES you need on Cortex-M3 and M4, in *Selected Areas in Cryptography — SAC 2016*, Rev. Sel. Pap. 23rd Int. Conf. (St. John's, NL, Canada, Aug. 10–12, 2016) (Springer, Cham, 2016), pp. 180–194 (Lect. Notes Comput. Sci., Vol. 10532), DOI: 10.1007/978-3-319-69453-5_10.

72. K. Tiri, M. Akmal, and **I. Verbauwhede,** A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards, in *Proc. 28th European Solid-State Circuits Conf.* (Florence, Italy, Sept. 24–26, 2002) (IEEE, Piscataway, 2002), pp. 403–406.

73. N. Patterson, The algebraic decoding of Goppa codes, *IEEE Trans. Inf. Theory* **21** (2), 203–207 (1975), DOI: 10.1109/TIT.1975.1055350.

74. F. Strenzke, E. Tews, G. Molter, [et al.], Side channels in the McEliece PKC, in *Post-Quantum Cryptography*, Proc. 2nd Int. Workshop (Cincinnati, OH, USA, Oct. 17–19, 2008) (Springer, Heidelberg, 2008), pp. 216–229 (Lect. Notes Comput. Sci., Vol. 5299), DOI: 10.1007/978-3-540-88403-3_15.

75. R. Avanzi, S. Hoerder, D. Page, [et al.], Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems, *J. Cryptogr. Eng.* **1** (4), 271–281 (2011), DOI: 10.1007/s13389-011-0024-9.

76. A. Shoufan, F. Strenzke, H. G. Molter, [et al.], A timing attack against Patterson algorithm in the McEliece PKC, in *Information Security and Cryptology — ICISC 2009*, Rev. Sel. Pap. 12th Int. Conf. (Seoul, Korea, Dec. 2–4, 2009) (Springer, Heidelberg, 2010), pp. 161–175 (Lect. Notes Comput. Sci., Vol. 5984), DOI: 10.1007/978-3-642-14423-3_12.

77. F. Strenzke, A timing attack against the secret permutation in the McEliece PKC, in *Post-Quantum Cryptography*, Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010) (Springer, Heidelberg, 2010), pp. 95–107 (Lect. Notes Comput. Sci., Vol. 6061), DOI: 10.1007/978-3-642-12929-2_8.

78. F. Strenzke, Timing attacks against the syndrome inversion in code-based cryptosystems, in *Post-Quantum Cryptography*, Proc. 5th Int. Workshop (Limoges, France, June 4–7, 2013) (Springer, Heidelberg, 2013), pp. 217–230 (Lect. Notes Comput. Sci., Vol. 7932).

79. S. Heyse, A. Moradi, and **C. Paar,** Practical power analysis attacks on software implementations of McEliece, in *Post-Quantum Cryptography*, Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010) (Springer, Heidelberg, 2010), pp. 108–125 (Lect. Notes Comput. Sci., Vol. 6061), DOI: 10.1007/978-3-642-12929-2_9.

80. H. G. Molter, M. Stöttinger, A. Shoufan, [et al.], A simple power analysis attack on a McEliece cryptoprocessor, *J. Cryptogr. Eng.* **1** (1), 29–36 (2011), DOI: 10.1007/s13389-011-0001-3.

81. A. Shoufan, T. Wink, H. G. Molter, [et al.], A novel processor architecture for McEliece cryptosystem and FPGA platforms, in *Proc. 20th IEEE Int. Conf. Application-Specific Systems, Architectures and Processors* (Boston, MA, USA, July 7–9, 2009) (IEEE Comput. Soc., Los Alamitos, CA, 2009), pp. 98–105, DOI: 10.1109/ASAP.2009.29.

82. S. Heyse, I. Von Maurich, and **T. Güneysu,** Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices, in *Cryptographic Hardware and Embedded Systems — CHES 2013*, Proc. 15th Int. Workshop (Santa Barbara, USA, Aug. 20–23, 2013) (Springer, Heidelberg, 2013), pp. 273–292 (Lect. Notes Comput. Sci., Vol. 8086), DOI: 10.1007/978-3-642-40349-1_16.

83. I. Von Maurich, and **T. Güneysu,** Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices, in *Post-Quantum Cryptography*, Proc. 6th Int. Workshop (Waterloo, ON, Canada, Oct. 1–3, 2014) (Springer, Cham, 2014), pp. 266–282 (Lect. Notes Comput. Sci., Vol. 8772), DOI: 10.1007/978-3-319-11659-4_16.

84. I. Von Maurich, and **T. Güneysu,** Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices, in *2014 Design, Automation and Test in Europe Conf.* (Dresden, Germany, Mar. 24–28, 2014) (IEEE, Piscataway, 2014), pp. 1–6, DOI: 10.7873/DATE.2014.051.

85. C. Chen, T. Eisenbarth, and **I. Von Maurich,** [et al.], Differential power analysis of a McEliece cryptosystem, in *Applied Cryptography and Network Security*, Rev. Sel. Pap. 13th Int. Conf. (New York, USA, June 2–5, 2015) (Springer, Cham, 2015), pp. 538–556 (Lect. Notes Comput. Sci., Vol. 9092).

86. T. Schamberger, J. Renner, G. Sigl, [et al.], A power side-channel attack on the CCA2-secure HQC KEM, in *Smart Card Research and Advanced Applications*, Rev. Sel. Pap. 19th Int. Conf. (Lübeck, Germany, Nov. 18–19, 2020) (Springer, Cham, 2020), pp. 119–134 (Lect. Notes Comput. Sci., Vol. 12609), DOI: 10.1007/978-3-030-68487-7_8.

87. C. Hlauschek, N. Lahr, and **R. L. Schröder,** On the timing leakage of the deterministic re-encryption in HQC KEM (Univ. California, San Diego, 2021) (Cryptol. ePrint Archive, Pap. 2021/1485/20211115:124514), URL: eprint.iacr.org/archive/2021/1485/20211115:124514 (accessed: 6.03.2026).

88. G. Wafo-Tapa, S. Bettaieb, L. Bidoux, [et al.], A practicable timing attack against HQC and its countermeasure, *Adv. Math. Commun.* **16** (3), 621–642 (2022), DOI: 10.3934/amc.2020126.

89. T. B. Paiva and **R. Terada,** A timing attack on the HQC encryption scheme, in *Selected Areas in Cryptography — SAC 2019*, Rev. Sel. Pap. 26th Int. Conf. (Waterloo, ON, Canada, Aug. 12–16, 2019) (Springer, Cham, 2019), pp. 551–573 (Lect. Notes Comput. Sci., Vol. 11959), DOI: 10.1007/978-3-030-38471-5_22.

90. N. Lahr, R. Niederhagen, R. Petri, [et al.], Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation, in *Advances in Cryptology — ASIACRYPT 2020*, Proc. 26th Int. Conf. Theory and Application of Cryptology and Information Security (Daejeon, South Korea, Dec. 7–11, 2020), Pt. I (Springer, Cham, 2020), pp. 881–910 (Lect. Notes Comput. Sci., Vol. 12491), DOI: 10.1007/978-3-030-64837-4_29.

91. Q. Guo, A. Johansson, and **T. Johansson,** A key-recovery side-channel attack on Classic McEliece implementations, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (4), 800–827 (2022), DOI: 10.46586/tches.v2022.i4.800-827.

92. S. Pircher, J. Geier, J. Danner, [et al.], Key-recovery fault injection attack on the Classic McEliece KEM, in *Code-Based Cryptography*, Rev. Sel. Pap. 10th Int. Workshop (Trondheim, Norway, May 29–30, 2022) (Springer, Cham, 2022), pp. 37–61 (Lect. Notes Comput. Sci., Vol. 13839), DOI: 10.1007/978-3-031-29689-5_3.

93. V. Grosso, P.-L. Cayrel, B. Colombier, [et al.], Punctured syndrome decoding problem: Efficient side-channel attacks against Classic McEliece, in *Constructive Side-Channel Analysis and Secure Design*, Proc. 14th Int. Workshop (Munich, Germany, Apr. 3–4, 2023) (Springer, Cham, 2023), pp. 170–192 (Lect. Notes Comput. Sci., Vol. 13979), DOI: 10.1007/978-3-031-29497-6_9.

94. M. Brinkmann, C. Chuengsatiansup, A. May, [et al.], Leaky McEliece: Secret key recovery from highly erroneous side-channel information, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2025** (2), 94–125 (2025), DOI: 10.46586/tches.v2025.i2.94-125.

95. S. Bitzer, J. Delvaux, E. Kirshanova, [et al.], How to lose some weight: A practical template syndrome decoding attack, *Des. Codes Cryptogr.* **93** (7), 2503–2519 (2025), DOI: 10.1007/s10623-025-01603-1.

96. V.-F. Drăgoi, B. Colombier, N. Vallet, [et al.], Full key-recovery cubic-time template attack on Classic McEliece decapsulation (Univ. California, San Diego, 2024) (Cryptol. ePrint Archive, Pap. 2024/1694), URL: eprint.iacr.org/2024/1694 (accessed: 6.03.2026).

97. R. Ueno, K. Xagawa, Y. Tanaka, [et al.], Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (1), 296–322 (2022), DOI: 10.46586/tches.v2022.i1.296-322.

98. D. J. Bernstein, T. Lange, and **C. Peters,** Attacking and defending the McEliece cryptosystem, in *Post-Quantum Cryptography*, Proc. 2nd Int. Workshop (Cincinnati, OH, USA Oct. 17–19, 2008) (Springer, Heidelberg, 2008), pp. 31–46 (Lect. Notes Comput. Sci., Vol. 5299), DOI: 10.1007/978-3-540-88403-3_3.

99. A. Canteaut and **F. Chabaud,** A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511, *IEEE Trans. Inf. Theory* **44** (1), 367–378 (1998), DOI: 10.1109/18.651067.

100. A. Canteaut and **N. Sendrier,** Cryptanalysis of the original McEliece cryptosystem, in *Advances in Cryptology — ASIACRYPT'98*, Proc. Int. Conf. Theory and Application of Cryptology and Information Security (Beijing, China, Oct. 18–22, 1998) (Springer, Heidelberg, 1998), pp. 187–199 (Lect. Notes Comput. Sci., Vol. 1514), DOI: 10.1007/3-540-49649-1_16.

101. E. Eaton, M. Lequesne, A. Parent, [et al.], QC-MDPC: A timing attack and a CCA2 KEM, in *Post-Quantum Cryptography*, Proc. 9th Int. Conf. (Fort Lauderdale, FL, USA, Apr. 9–11, 2018) (Springer, Cham, 2018), pp. 47–76 (Lect. Notes Comput. Sci., Vol. 10786), DOI: 10.1007/978-3-319-79063-3_3.

102. N. Drucker, S. Gueron, and **D. Kostic,** QC-MDPC decoders with several shades of gray, in *Post-Quantum Cryptography*, Proc. 11th Int. Conf. (Paris, France, Apr. 15–17, 2020) (Springer, Cham, 2020), pp. 35–50 (Lect. Notes Comput. Sci., Vol. 12100), DOI: 10.1007/978-3-030-44223-1_3.

103. T. Schamberger, L. Holzbaur, J. Renner, [et al.], A power side-channel attack on the Reed–Muller Reed–Solomon version of the HQC cryptosystem, in *Post-Quantum Cryptography*, Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28-30, 2022) (Springer, Cham, 2022), pp. 327–352 (Lect. Notes Comput. Sci., Vol. 13512), DOI: 10.1007/978-3-031-17234-2_16.

104. G. Goy, A. Loiseau, and **P. Gaborit,** A new key recovery side-channel attack on HQC with chosen ciphertext, in *Post-Quantum Cryptography*, Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28–30, 2022) (Springer, Cham, 2022), pp. 353–371 (Lect. Notes Comput. Sci., Vol. 13512), DOI: 10.1007/978-3-031-17234-2_17.

105. T. Chou, QcBits: Constant-time small-key code-based cryptography, in *Cryptographic Hardware and Embedded Systems — CHES 2016*, Proc. 18th Int. Conf. (Santa Barbara, USA, Aug. 17–19, 2016) (Springer, Heidelberg, 2016), pp. 280–300 (Lect. Notes Comput. Sci., Vol. 9813), DOI: 10.1007/978-3-662-53140-2_14.

106. M. Rossi, M. Hamburg, M. Hutter, [et al.], A side-channel assisted cryptanalytic attack against QcBits, in *Cryptographic Hardware and Embedded Systems — CHES 2017*, Proc. 19th Int. Conf. (Taipei, China, Sept. 25–28, 2017) (Springer, Cham, 2017), pp. 3–23 (Lect. Notes Comput. Sci., Vol. 10529), DOI: 10.1007/978-3-319-66787-4_1.

107. B.-Y. Sim, J. Kwon, K. Y. Choi, [et al.], Novel side-channel attacks on quasi-cyclic code-based cryptography, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019** (1), 180–212 (2019), DOI: 10.46586/tches.v2019.i4.180-212.

108. Q. Guo, C. Hlauschek, T. Johansson, [et al.], Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (3), 223–263 (2022), DOI: 10.46586/tches.v2022.i3.223-263.

109. N. Sendrier, Secure sampling of constant-weight words — Application to BIKE (Univ. California, San Diego, 2021) (Cryptol. ePrint Archive, Pap. 2021/1631), URL: eprint.iacr.org/2021/1631 (accessed: 6.03.2026).

110. S. Huang, Q. R. Sim, C. Chuengsatiansup, [et al.], Cache-timing attack against HQC (Univ. California, San Diego, 2023) (Cryptol. ePrint Archive, Pap. 2023/102), URL: eprint.iacr.org/2023/102 (accessed: 6.03.2026).

111. G. Goy, J. Maillard, P. Gaborit, [et al.], Single trace HQC shared key recovery with SASCA, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2024** (2), 64–87 (2024), DOI: 10.46586/tches.v2024.i2.64-87.

Aleksandr O. Bakharev
Denis M. Voronov
Nikolay A. Kolomeec
Natalia N. Tokareva
Irina S. Khilchuk
Aleksandr S. Shaporenko

Received June 18, 2025
Revised August 11, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 69–101

УДК 519.176 DOI: 10.33048/daio.2025.32.815

ОБ ЭКСТРЕМАЛЬНЫХ ДВУСВЯЗНЫХ ГРАФАХ С ФИКСИРОВАННЫМ ДИАМЕТРОМ

Д. Л. Белоцерковский

Российский гос. университет нефти и газа (НИУ) им. И. М. Губкина,
Ленинский пр., 65, 119991 Москва, Россия

E-mail: belozer68@mail.ru

Аннотация. Рассматриваются две задачи теории экстремальных графов. Первая задача сформулирована и полностью доказана в виде теоремы о точной нижней границе числа рёбер для двусвязных графов с диаметром, не превосходящим некоторого фиксированного значения. В процессе доказательства разработан метод перераспределения вкладов, который в дальнейшем применяется для решения второй задачи: перечисления графов, на которых найденная граница достигается. Так как задача перечисления графов весьма трудна и ранее в общем виде не решалась, на базе разработанного метода перечислены все так называемые предельные графы. Задача нахождения всех предельных графов сформулирована и доказана в виде второй теоремы статьи. Далее объясняется, почему именно предельные графы наиболее интересны с точки зрения задачи перечисления, а также приведены приёмы генерации остальных экстремальных графов с помощью уже полученных предельных графов. Ил. 11, библиогр. 21.

Ключевые слова: граф, диаметр, точная нижняя граница, распределение вкладов, метод перераспределения вкладов (МПВ).

Введение

Общая постановка рассмотренной в статье задачи сформулирована в 1970–80-х гг. в рамках проблемы топологической оптимизации сетей ЭВМ [1–3]. Требовалось генерировать графы с заданными надёжностными свойствами с минимальным числом рёбер [3–5]. Из-за трудности решаемой задачи многие публикации по этой тематике касались фиксированных значений диаметра графа [6–11]. В этой работе разрабатывается методика для произвольного диаметра и усиливаются предыдущие результаты, полученные автором [12].

Рассматриваются конечные неориентированные простые графы [13, 14]. Обозначим через $\mathfrak{I}(n,d)$ множество двусвязных графов с n вершинами и диаметром, не превосходящим d, а через $f(n,d)$ — минимальное число рёбер в графах из класса $\mathfrak{I}(n,d)$. Графы из $\mathfrak{I}(n,d)$ с минимальным числом рёбер, равным $f(n,d)$, назовём *экстремальными*.

Значение $f(n,d)$ изучалось и ранее. В частности, Боллобаш [15, с. 188; 16, с. 194] показал, что $f(n,d)/n \to d/(d-1)$ при $n \to \infty$. Мурти в [5, 6] и Кассетта в [7–9] доказали, что при $d \leqslant 4$

$$f(n,d) = \left\lceil \frac{dn-2d-1}{d-1} \right\rceil. \tag{1}$$

В [16] Боллобаш отметил нехватку идей для решения возникающих проблем. Именно поэтому интерес к подобным задачам достаточно быстро угас. Можно отметить переиздание книги Боллобаша спустя 26 лет [17], а также последующую публикацию ещё одной книги [18] и её переиздание [19] с главой, посвящённой экстремальным задачам. В дальнейшем похожие задачи рассматривались уже для случайных графов [20].

Автором в [12] предложен метод перераспределения вкладов (далее МПВ), позволяющий определять точную нижнюю границу числа рёбер в графах с ограничением на диаметр. Развитие МПВ позволяет находить графы, на которых эта граница достигается. С помощью предложенного метода автор в [12] доказал равенство (1) при $d \geqslant 5$ для всех достаточно больших n.

Насколько известно автору, последняя публикация, посвящённая этим задачам, датируется 2012 годом [21]. В ней (1) доказано при любых $d \geqslant 2$ и $n \geqslant 4$, для чего предложен метод анализа циклов в рассматриваемых графах. При этом в [21] отсутствуют идеи для решения задачи перечисления экстремальных графов.

В настоящей статье не только показано, что равенство (1) имеет место для любых $d \geqslant 2$ и $n \geqslant 5$, но и решается задача перечисления графов с $f(n,d)$ рёбрами. В работе все переменные d,k,l,m,n,p,q,i,j принимают только целые неотрицательные значения и, в основном, используется терминология, данная в [12].

1. Формулировки теорем с необходимыми пояснениями

Далее докажем следующую теорему.

Теорема 1. *При $d \geqslant 2$ и $n \geqslant 5$ выполнено $f(n,d) = \left\lceil \frac{dn-2d-1}{d-1} \right\rceil$.*

Что касается графов, на которых достигается значение $f(n,d)$, то попытка аналитически перечислить все такие графы из теоремы 1 выглядит крайне непростой задачей из-за огромного их числа и никогда ранее

не решалась. В этой статье предложен МПВ, который позволяет, в целом, значительно продвинуться в решении этой задачи.

Обоснуем корректность оценки $f(n,d)$. Теорема 1 тривиальна в случае $n \leqslant 2d+1$. Очевидно, что при этом экстремальны только циклы с n вершинами, n рёбрами и $d = \lfloor n/2 \rfloor$. Проверим, что $f(n,d) = n$. Если n нечётно, то $d = (n-1)/2$ и

$$f(n,d) = \left\lceil \frac{(n-1)n - 2(n-1) - 2}{n-3} \right\rceil = \lceil n \rceil = n.$$

Если n чётно, то $d = n/2$ и

$$f(n,d) = \left\lceil \frac{n^2 - 2n - 2}{n-2} \right\rceil = \left\lceil n - \frac{2}{n-2} \right\rceil = n.$$

Если $n > 2d+1$, то экстремальные графы состоят из большего числа рёбер, чем цикл, поэтому $f(n,d) > n$. Сделаем оценку сверху числа $f(n,d)$.

Утверждение 1. *Если* $n \geqslant 2d+2$, *то* $f(n,d) \leqslant \left\lceil \frac{dn-2d-1}{d-1} \right\rceil$.

Доказательство. Предположим, что

$$\left\lceil \frac{dn-2d-1}{d-1} \right\rceil = \frac{dn-2d-1+d-1-j}{d-1},$$

где

$$j = \begin{cases} d-1, & \text{если } d-1 \mid dn-2d-1, \\ (dn-2d-1) \bmod (d-1) & \text{иначе.} \end{cases} \tag{2}$$

Рассмотрим граф G, состоящий из $k \geqslant 1$ различных цепей длины d, одной цепи длины $d+1$ и одной цепи длины $j+1$ (если $j = d-1$, то цепей длины d становится $k+1$). Под длиной цепи понимаем число рёбер в ней. Концевые вершины (далее концы) у всех цепей одни и те же — это вершины v_0 и v_0', а внутренние вершины степени 2 попарно различны (степень $\deg v$ вершины v равна числу инцидентных ей рёбер). Очевидно, что $G \in \mathfrak{I}(n,d)$. Подсчитаем число вершин и рёбер в этом графе. Число вершин равно $n = k(d-1) + d + 2 + j$, откуда $k = (n-d-2-j)/(d-1)$, а число рёбер —

$$kd + d + 1 + j + 1 = \frac{dn-d-j-2}{d-1} = \\ = \frac{dn-2d-1+(d-j-1)}{d-1} = \left\lceil \frac{dn-2d-1}{d-1} \right\rceil.$$

На рис. 1 представлен граф $G \in \mathfrak{I}(12,4)$, где $k=1$, $j=d-1$. Утверждение 1 доказано.

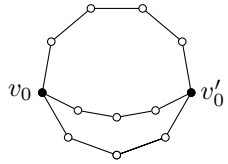


Рис. 1. Предельный граф из $\mathfrak{I}(12, 4)$

Из утверждения 1 также следует, что

$$f(n,d) \leqslant \left\lceil \frac{dn-2d-1}{d-1} \right\rceil = \left\lceil 2n-1-\frac{(d-2)n+d+2}{d-1} \right\rceil < 2n.$$

Тем самым доказано

Утверждение 2. *Если* $n \geqslant 2d+2$, *то* $n < f(n,d) < 2n$.

Далее определим, какие графы будем перечислять с помощью утверждения 2.

Положим $g(n,d) = (dn-2d-1)/(d-1)$, если число в правой части натурально, а при $(dn-2d-1)/(d-1) \notin \mathbb{N}$ оставим $g(n,d)$ неопределённым. Другими словами,

$$g(n,d) = \begin{cases} f(n,d), & \text{если } j = d-1 \text{ в (2)}, \\ \text{не существует}, & \text{если } 1 \leqslant j \leqslant d-2. \end{cases}$$

Рассмотрим некоторый экстремальный граф с n вершинами и $m = g(n,d)$ рёбрами. Экстремальные графы с $n+j$ вершинами, $1 \leqslant j \leqslant d-1$, назовём *серией*.

По теореме 1 число рёбер в графе из серии равно

$$f(n+j,\, d) = \left\lceil \frac{d(n+j)-2d-1}{d-1} \right\rceil = \left\lceil \frac{dn-2d-1}{d-1} + \frac{dj}{d-1} \right\rceil =$$
$$= g(n,d) + \left\lceil j + \frac{j}{d-1} \right\rceil = g(n,d) + j + 1.$$

Тем самым графы серии, имеющие на одну вершину больше, содержат и рёбер на одно больше. Графы серии при $j = 1$ назовём *начальными*, при $2 \leqslant j \leqslant d-2$ — *промежуточными*, а при $j = d-1$ — *предельными*. Таким образом, только предельные графы содержат $g(n+d-1, d) = g(n,d)+d$ рёбер. Заметим, что также по теореме 1 начальный граф серии содержит на одну вершину и на два ребра больше, чем предельный граф предыдущий серии.

Поясним, чем интересна задача поиска предельных графов, введя для различных графов серии *коэффициент рёберности*

$$\beta_j = \frac{f(n+j,\, d)}{n+j}, \quad 1 \leqslant j \leqslant d-1,$$

который показывает среднее число рёбер, инцидентных одной вершине графа.

Лемма 1. *Последовательность коэффициентов рёберности убывающая.*

Доказательство. Вычислим $\beta_{j-1} - \beta_j$, используя утверждение 2:

$$\beta_{j-1} - \beta_j = \frac{f(n+j-1,d)}{n+j-1} - \frac{f(n+j,\,d)}{n+j} = \\ = \frac{f(n+j-1,d)}{n+j-1} - \frac{f(n+j-1,d)+1}{n+j} = \\ = \frac{f(n+j-1,d)-(n+j-1)}{(n+j-1)(n+j)} > 0.$$

Лемма 1 доказана.

Следствие 1. *Коэффициент рёберности минимален для предельных графов серии, т. е.* $\beta_{d-1} = \min\limits_{1\leqslant j\leqslant d-1} \beta_j$.

Лемма 2. *Коэффициент рёберности начального графа серии выше, чем у предельного графа предыдущей серии, т. е.* $\beta_1 > g(n,d)/n$.

Доказательство. Снова воспользуемся утверждением 2:

$$\beta_1 - \frac{g(n,d)}{n} = \frac{f(n+1,d)}{n+1} - \frac{g(n,d)}{n} = \\ = \frac{g(n,d)+2}{n+1} - \frac{g(n,d)}{n} = \frac{2n-g(n,d)}{n(n+1)} > 0.$$

Лемма 2 доказана.

Из лемм 1 и 2 следует, что коэффициент рёберности предельного графа ниже, чем у остальных графов серии и начального графа следующей серии. Говоря неформально, в этом смысле предельные графы можно считать «самыми» экстремальными, в которых в среднем одной вершине инцидентно наименьшее число рёбер. Далее установим принцип, по которому эти графы генерируются.

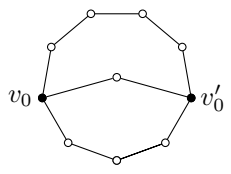


Рис. 2. Начальный граф из $\mathfrak{I}(10,4)$

Рассмотрим, например, экстремальные графы из класса $\mathfrak{I}(n,4)$ при $10 \leqslant n \leqslant 12$. Из теоремы 1 следует, что $f(9,4) = \lceil (4{\cdot}9{-}9)/3 \rceil = g(9,4) = 9$ и единственный предельный граф — это цикл из 9 вершин. При $n = 10$ графы указанной серии имеют $f(10,4) = g(9,4){+}2 = 11$ рёбер и являются начальными (один из них на рис. 2), а при $n = 12$ имеют $f(12,4) = g(9,4) + 4 = 13$ рёбер и являются предельными (рис. 1).

Выясним как связаны друг с другом графы серии. Назовём *генерацией* любой приём, позволяющий получить из графа серии другой граф той же серии, но с бóльшим или меньшим числом вершин. Назовём *l-нитью* цепь длины l в графе G, все внутренние вершины которой имеют степень 2. Ребро будем считать l-нитью для $l = 1$. Назовём *удлинением* такое увеличение длины l-нити, которое сохраняет полученный с помощью этой операции граф в классе $\mathfrak{I}(n,d)$, т. е. не увеличивает диаметра. Удлинение увеличивает число вершин и рёбер в графе на одинаковое значение, а также сохраняет двусвязность. Например, предельный граф из $\mathfrak{I}(12,4)$ на рис. 1 получен из начального графа $\mathfrak{I}(10,4)$ на рис. 2 удлинением его 2-нити на 2.

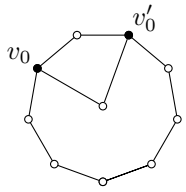


Рис. 3. Другой начальный граф из $\mathfrak{I}(10,4)$

Не всякий начальный граф может быть «удлинён». Так, например, граф из $\mathfrak{I}(10,4)$ на рис. 3 начальный, но удлинение любой его нити приводит к увеличению диаметра.

Обратную операцию уменьшения длины l-нити назовём *укорочением*, если длина получающейся в результате нити не менее 1. Например, 2-нить графа на рис. 2 может быть получена из 4-нити графа на рис. 1 укорочением последней на 2. Очевидно, что после укорочения граф остаётся в классе $\mathfrak{I}(n,d)$, поэтому укорочение и удлинение генерируют некоторые графы серии.

Лемма 3. *Любой экстремальный граф серии из класса $\mathfrak{I}(n+j,d)$, $2 \leqslant j \leqslant d-1$, может быть получен удлинением из некоторого начального графа класса $\mathfrak{I}(n+1,d)$.*

Доказательство. Рассмотрим произвольный граф $G_1 \in \mathfrak{I}(n+j,d)$. Уменьшим длину некоторой l-нити в G_1 на единицу и проведём эту операцию $j-1$ раз для, возможно, разных нитей. Тем самым с помощью

укорочения получим начальный граф $G_2 \in \mathfrak{I}(n+1, d)$, содержащий рёбер и вершин на $j-1$ меньше, поэтому можно сказать, что G_1 сгенерирован из G_2 с помощью удлинения. Лемма 3 доказана.

В качестве примера рассмотрим экстремальные графы из $\mathfrak{I}(n, d)$ для $n \geqslant 2d+1$. Если $n = 2d+1$, то единственный экстремальный граф — это цикл длины $2d+1$, содержащий столько же вершин. Так как $g(2d+1, d) = 2d+1$, это предельный граф. Рассмотрим некоторые графы серии при $2d+2 \leqslant n \leqslant 3d$. Для $n = 2d+2$ имеем число рёбер $m = f(2d+2, d) = 2d+3$ и некоторое множество экстремальных начальных графов, два из которых изображены на рис. 2 и 3 при $n = 10$, $d = 4$.

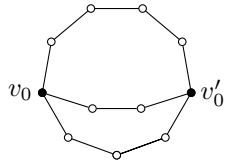


Рис. 4. Промежуточный граф из $\mathfrak{I}(11, 4)$

Удлиним 2-нить с концами v_0 и v_0' графа на рис. 2. Если удлиним на 1, то получим промежуточный граф из $\mathfrak{I}(11, 4)$ (рис. 4); если на 2, то — предельный граф из $\mathfrak{I}(12, 4)$ (см. рис. 1).

Экстремальные графы серии можно разделить на три непустых множества: начальные (см. рис. 2 и 3), промежуточные, полученные из начальных удлинением не более $d-3$ раз (см. рис. 4), и предельные, полученные из начальных удлинением $d-2$ раз и которые более нельзя «удлинить» (см. рис. 1). Перечисление всех предельных графов является одной из целей этой статьи в дальнейшем.

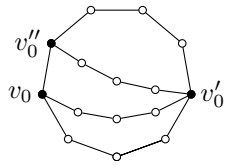


Рис. 5. Предельный граф из $\mathfrak{I}(15, 4)$

Предельные графы из $\mathfrak{I}(12, 4)$ с $g(12, 4) = (4 \cdot 12 - 2 \cdot 4 - 1)/(4-1) = 13$ и из $\mathfrak{I}(15, 4)$ с $g(15, 4) = (4 \cdot 15 - 2 \cdot 4 - 1)/(4-1) = 17$ рёбрами представлены на рис. 1 и 5.

Введём ещё одну операцию, позволяющую генерировать экстремальные графы. На рис. 6 представлена l-нить $v_0 v_1 v_2 v_0'$ некоторого графа для $l = 3$, при этом концы нити v_0 и v_0' могут иметь как степень 2, так

и бóльшую степень. Добавим к графу новую l-нить $v_0v_3v_4v_0'$ той же длины (обозначена штриховой линией), концы которой будут совпадать с v_0 и v_0'. Такую операцию будем называть *дублированием* l-нити. Нетрудно видеть, что дублирование не увеличивает диаметра и сохраняет двусвязность графа.

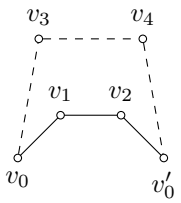


Рис. 6. Операция генерации экстремальных графов

Следующая теорема служит перечислением предельных графов при $d \geqslant 3$, $n \geqslant 7$. При $d \leqslant 4$ эта задача решена в [5–9], поэтому далее рассматриваем $d \geqslant 5$.

Теорема 2. *Множество предельных графов из* $\mathfrak{I}(n,d)$ *для* $d \geqslant 3$ *и* $n \geqslant 7$ *состоит из цикла длины* $2d+1$ *и графов, полученных из него дублированием либо одной и той же* d*-нити* (*см. рис.* 1), *либо двух разных* d*-нитей, имеющих общий конец* v_0' (*см. рис.* 5).

На рис. 5 и 6 дублированы 3- и 4-нити с концами v_0, v_0' и v_0', v_0''. Заметим также, что из предельного графа с помощью дублирования можно получать начальные графы. Например, начальный граф с 10 вершинами на рис. 3 получен дублированием 2-нити из предельного — простого цикла длины 9.

Доказательствам теорем 1 и 2 отведена оставшаяся часть статьи. Идея доказательства теоремы 1 состоит в поиске графа $G' \in \mathfrak{I}(n,d)$ с числом рёбер $m = (dn-2d-2)/(d-1) \in \mathbb{N}$. Заметим, что $m < f(n,d)$, поскольку

$$\frac{dn-2d-2}{d-1} = \frac{dn-2d-1}{d-1} - \frac{1}{d-1} < \left\lceil \frac{dn-2d-1}{d-1} \right\rceil = f(n,d),$$

поэтому если такой граф G' будет найден, то для всех рассматриваемых n получим $f(n,d) = \lceil (dn-2d-2)/(d-1) \rceil$.

Будем считать граф G' предельным, а графы из $\mathfrak{I}(n+1,d)$ начальными, полученными из G', например, дублированием 2-нити. Число рёбер такого начального графа равно

$$\frac{dn-2d-2}{d-1} + 2 = \frac{d(n+1)-2d-2}{d-1} + \frac{d-2}{d-1} = \left\lceil \frac{d(n+1)-2d-2}{d-1} \right\rceil.$$

По лемме 3 остальные графы получаются с помощью удлинения, при этом в графе с $n+i$ вершинами $f(n+i,d) = \lceil (d(n+i)-2d-2)/(d-1) \rceil$

рёбер, $1 \leqslant i \leqslant d-1$. Перебрав все возможные варианты, покажем, что графа G' не существует, откуда $f(n,d) \geqslant \lceil (dn-2d-1)/(d-1) \rceil$. Последнее неравенство вместе с утверждением 1 приводят к равенству $f(n,d) = \lceil (dn-2d-1)/(d-1) \rceil$, тем самым доказывая теорему 1.

Для доказательства теоремы 2 будем искать предельные графы $G'' \in \mathfrak{I}(n,d)$ с числом рёбер $m = g(n,d) = (dn-2d-1)/(d-1)$ и найдём только те графы, которые указаны в формулировке теоремы.

Некоторые тонкости доказательств содержатся в разд. 7 с необходимыми ссылками в тексте.

2. Случай длинных l-нитей и общая схема МПВ

Покажем, что в предельном графе G' нет длинных нитей.

Лемма 4. *Длина любой l-нити в предельном графе $G' \in \mathfrak{I}(n,d)$, имеющем $m \leqslant (dn-2d-2)/(d-1)$ рёбер, не превосходит $d-1$, т. е. $l \leqslant d-1$.*

Доказательство. Предположим, напротив, что в G' имеется l-нить, $l \geqslant d$, и рассмотрим некоторую d-нить — подграф этой l-нити (обе нити могут совпадать при $l = d$). Если дублировать выбранную d-нить, то в результате получим граф из $\mathfrak{I}(n+d-1,d)$.

Воспользуемся теоремой 2 из [12], где доказано, что при $d \geqslant 5$ графа G' не существует для любого числа вершин, начиная с некоторого n, зависящего от d и его чётности. Выберем k такое, чтобы $n+k(d-1)$ было не меньше, чем указанные значения в [12], и k раз дублируем отмеченную d-нить в G'. Получаем граф из $\mathfrak{I}(n+k(d-1),d)$ с числом рёбер, равным

$$m + kd \leqslant \left\lceil \frac{dn-2d-2}{d-1} + kd \right\rceil = \left\lceil \frac{d(n+k(d-1))-2d-2}{d-1} \right\rceil.$$

Однако, как доказано в [12], такого графа не существует, следовательно, $l \leqslant d-1$ при $d \geqslant 5$. В [5–9] получен аналогичный результат при $d \leqslant 4$. Лемма 4 доказана.

Дальнейшие рассуждения будем применять к графу G, подразумевая под ним любой из графов G' и G'', если их различие не оговорено особо. Стало быть, пусть G — искомый граф с $m \leqslant (dn-2d-1)/(d-1)$ рёбрами.

Начнём с того, что припишем каждому ребру графа G вклад, равный 2. Ребро uv отдаёт вклады двум инцидентным вершинам u и v: значение $\alpha_{uv}^u \geqslant 0$ — вершине u и значение $\alpha_{uv}^v \geqslant 0$ — вершине v, при этом $\alpha_{uv}^u + \alpha_{uv}^v = 2$. Такой перенос вкладов с рёбер на вершины будем называть методом перераспределения вкладов (МПВ). Далее для простоты будем писать $\alpha_{uv}^u = \alpha_u$, $\alpha_{uv}^v = \alpha_v$, если ребро uv очевидно из контекста.

Обозначив через $N(v)$ множество соседей вершины $v \in V(G)$, назовём её *условной степенью* величину

$$\deg' v = \sum_{u \in N(v)} \alpha_{uv}^{v}.$$

Если здесь $\alpha_{uv}^{v} = 1$ для каждого ребра uv, то $\deg' v = \deg v$. В любом случае имеет место равенство $\sum\limits_{v \in V(G)} \deg' v = \sum\limits_{v \in V(G)} \deg v = 2m$.

3. Подграфы G_1 и G_2 и классификация вершин

Пусть G_1 и G_2 — подграфы графа G, а $V_0 \subseteq V(G)$ — разделяющее множество вершин такое, что

$$G_1 \cup G_2 = G, \quad V(G_1) \cap V(G_2) = V_0, \quad E(G_1) \cap E(G_2) = \varnothing.$$

Предположим, что граф G_i имеет n_i вершин и m_i рёбер, $i = 1, 2$, а $n_0 = |V_0|$. Для оценки m_1 снизу будем распределять вклады в графе G_1, а $m_2 \geqslant n_2 - 1$ в силу того, что G_2 связен. Если в V_0 имеются смежные вершины, то рёбра между ними будем считать принадлежащими G_2.

Обозначим через V_p множество вершин графа G_1, находящихся на расстоянии $p \in \mathbb{N}$ от V_0, максимальное из которых обозначим через $q = p_{\max}$. Очевидно, что $V(G_1) = \bigcup\limits_{p=0}^{q} V_p$. Следует заметить, что некоторые V_p могут быть пустыми. Если потребуется, графы G_1, G_2, множество V_0 и максимальное расстояние $p_{\max}$ будем выбирать по усмотрению.

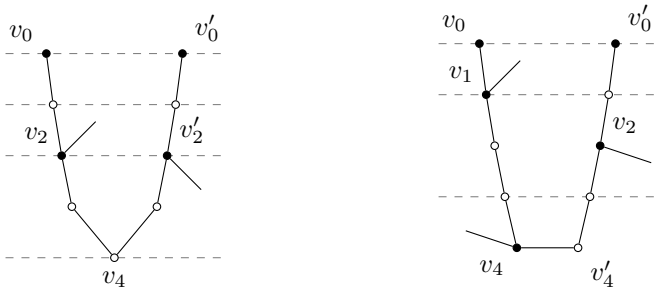
Граф G_1 «разберём» на различные цепи $v_0 \dots v_p v_{p+1} \dots v_{\overline{p}}$, где $\overline{p} \leqslant q$, в которых будет осуществляться перераспределение вкладов. Часто цепи будут рассматриваться по возрастанию индекса p. Будем говорить, что цепь *начинается* в v_0 и *заканчивается* в $v_{\overline{p}}$, имеет *первое* ребро v_0v_1 и *последнее* ребро $v_{\overline{p}-1}v_{\overline{p}}$. Вершины v_0 и $v_{\overline{p}}$ назовём *началом* и *концом* цепи соответственно, а остальные вершины цепи *внутренние*. Вклад последнего ребра $v_{\overline{p}-1}v_{\overline{p}}$ в вершину $v_{\overline{p}}$ назовём *последним вкладом* цепи. Для вершины v_p вклады от рёбер, инцидентных соседям из V_{p-1}, назовём *предыдущими*, а от рёбер, инцидентных соседям из V_{p+1}, *последующими*.

Далее классифицируем вершины графа G_1, которые поделим на два типа: сильные и несильные. Вершину $v_p \in V_p$, $p \geqslant 1$, имеющую хотя бы двух соседей из V_{p-1}, будем называть *сильной*, а если $|N(v_p) \cap V_{p-1}| = 1$, то вершина v_p *несильная*. Также будем считать сильными все вершины из V_0. Если $\deg v_p \geqslant 3$, то вершину v_p назовём *узлом*, иначе $\deg v_p = 2$ и вершину v_p будем называть *неузел*. Таким образом узлы и неузлы могут быть сильными и несильными.

Если у несильного узла из V_p нет соседей в V_p, то такую вершину назовём 1-*узлом*. Тем самым у 1-узла ровно один сосед в V_{p-1} и не менее двух

в V_{p+1}. Если два несильных узла из V_p смежны, то эти вершины назовём 2-*узлами*. Остальные узлы будут сильными, их назовём 3-*узлами*.

Классифицируем концы цепи. Если $\overline{p} \leqslant q$, то при $\deg v_{\overline{p}} = 2$ конец $v_{\overline{p}}$ назовём *особым*, а при $\deg v_{\overline{p}} \geqslant 3$ — *ординарным*. Из сказанного следует, что особый конец может иметь двух соседей в $V_{\overline{p}-1}$ и, следовательно, быть сильным. Если особый конец имеет одного соседа в $V_{\overline{p}-1}$, то назовём его *слабым*. Заметим, что ординарный конец является либо 2-, либо 3-узлом.



а *б*

Рис. 7. Примеры различных узлов подграфа G_1

На рис. 7 представлены возможные различные варианты вершин и цепей для $q = 4$. Белые вершины — неузлы, штриховой линией показаны «уровни» — подмножества V_p, $0 \leqslant p \leqslant 4$. На рис. 7а изображены вершины: сильные v_0, v'_0, сильная особая v_4, 1-узел v'_2, 3-узел v_2. Цепь $v'_0 \dots v_4$ состоит из двух подцепей — нитей $v'_0 \dots v'_2$ и $v'_2 \dots v_4$. На рис. 7б представлены 3-узел v_1, 2-узел v_2, ординарная вершина v_4 и особая слабая v'_4. В рассуждениях могут рассматриваться различные цепи: например, $v_0 \dots v_4 \dots v'_0$ на рис. 7а или $v_0 \dots v_4 v'_4 \dots v'_0$ на рис. 7б.

4. Структура доказательства теоремы 1, цепи, рёбра, вклады в G_1 и G_2

Если после распределения вкладов для вершины v графа $G_1 \setminus V_0$ получим $\deg' v \geqslant 2d/(d-1)$, то вершину v назовём *удачной*, а само неравенство — *условием удачности*. Пусть

$$C_1 = \sum_{v \in G_1 \setminus V_0} \deg' v - \frac{2d(n_1 - |V_0|)}{d-1}.$$

Если в графе $G_1 \setminus V_0$ все вершины удачны, то $C_1 \geqslant 0$. Вершина v *неудачна*, если $\deg' v < 2d/(d-1)$.

При анализе цепи $v_0 v_1 \dots v_{\overline{p}}$ в G_1 важную роль играют внутренние узлы цепи, для которых имеет место

Лемма 5. *Если все внутренние вершины цепи $v_0v_1\dots v_{\overline{p}}$ являются 1-узлами и все вершины этой цепи, кроме v_0, удачные, то условие удачности выполнено так же для всех вершин, кроме v_0, если внутренние вершины представляют собой произвольные узлы.*

Применяя лемму 5 и предполагая, что все узловые внутренние вершины цепи $v_0v_1\dots v_{\overline{p}}$ суть 1-узлы, будем выстраивать и корректировать схему МПВ так, чтобы все вершины рассматриваемой цепи, кроме v_0, были удачными (здесь под v_0 понимается произвольная вершина из V_0).

Рассмотрим нити наибольшей длины $l_{\max} = L$ в графе G. Из всех подобных нитей подграфа G_2 выберем нить T с концами v_0 и v_0' такую, что вершина v_0 имеет минимальную степень среди всех концевых вершин множества длиннейших нитей. Следовательно, $\deg v_0 \leqslant \deg v_0'$. Тогда $V_0 = \{v_0, v_0'\}$, $m_2 \geqslant n_2 - 1$. В подграфе G_2, кроме $n_2 - 1$ рёбер нити T, может иметься ребро v_0v_0', поэтому $m_2 = n_2 - 1 + C_2$, где $C_2 \in \{0, 1\}$, и

$$\sum_{v\in G_2} \deg' v = 2n_2 - 2 + 2C_2.$$

Далее предполагается, что в сумме $2n_2 - 2$ есть некоторый вклад, который обозначим через C и перераспределим в рамках МПВ, добавив к вкладам каждого ребра из $E(G_1)$, инцидентного v_0 и вершинам из V_1, чтобы дополнить условные степени некоторых неудачных вершин.

Значения C, C_1, C_2 будем называть *избытком*. Если вершина v неудачна, то величину $2d/(d-1) - \deg' v$ назовём *недостатком*. Если для доказательства удачности не используется избыток C, то такой случай назовём *стандартным*, если используется — *трудным*. Все эти случаи рассмотрены при доказательстве леммы 6.

Лемма 6. *Если в графе G длиннейшая нить имеет длину $L \leqslant d - 1$, то все вершины графа $G_1 \setminus V_0$ удачны.*

Доказательства лемм 5 и 6 представлены в разд. 7. Оценим значение C.

Утверждение 3. *Для графа G имеет место оценка $C \geqslant \frac{4}{d-1}$.*

Доказательство. Из леммы 6 следует, что в графе G вершины подграфа G_1 удачны. Тогда, используя соотношения $n = n_1 + n_2 - 2$, $n_2 = L + 1 \leqslant d$, получаем

$$\frac{2dn - 4d - 2}{d-1} \geqslant 2m = \sum_{v\in G_2} \deg' v + \sum_{v\in G_1} \deg' v \geqslant$$

$$\geqslant 2n_2 - 2 - C + 2C_2 + \frac{2d(n_1 - 2)}{d-1},$$

откуда

$$C \geqslant 2C_2 + \frac{4}{d-1} \geqslant \frac{4}{d-1}.$$

Утверждение 3 доказано.

5. Окончание доказательства теоремы 1

Утверждение 4. *Графа G' не существует.*

Доказательство. Пусть граф G' существует, а $G_2 = T$ — его нить наибольшей длины с концами v_0 и v_0'. Тогда $|V_0| = 2$ и в силу леммы 6 получаем

$$2m = \sum_{v \in G_1} \deg' v + \sum_{v \in G_2} \deg' v = \frac{2dn - 4d - 4}{d-1},$$

следовательно,

$$\sum_{v \in G_2} \deg' v = 2n_2 - 2 - C + 2C_2, \quad \sum_{v \in G_1} \deg' v = \frac{2d(n_1 - 2)}{d-1} + C_1.$$

Далее, используя оценку из утверждения 3 и соотношения $n = n_1 + n_2 - 2$, $n_2 = L + 1 \leqslant d$ (последнее неравенство следует из леммы 4), запишем равенство

$$2n_2 - 2 - C + 2C_2 + \frac{2d(n_1 - 2)}{d-1} + C_1 = \frac{2d(n_1 + n_2 - 2) - 4d - 4}{d-1}.$$

В результате получим $2C_2 + C_1 \leqslant -2/(d-1)$, что противоречит очевидному условию $2C_2 + C_1 \geqslant 0$. Утверждение 4 доказано.

Этим завершается доказательство теоремы 1.

6. Окончание доказательства теоремы 2

Найдём граф $G'' \in \mathfrak{I}(n, d)$ с $g(n, d) = (dn - 2d - 1)/(d-1)$ рёбрами для $d \geqslant 3$. Для G'' выполнены условия лемм 5 и 6, а дальнейшее рассмотрение будет зависеть от значения L.

Случай 1. Если $L \leqslant d - 2$, то из леммы 6 и условия $C \geqslant 4/(d-1)$ следует равенство

$$2m = 2L - C + \frac{2d(n - L - 1)}{d-1} = \frac{2dn - 4d - 2}{d-1},$$

поэтому $L = d - 1$ и таких графов нет.

В дальнейшем назовём вершину $v \in V(G_1)$ *строго удачной* (СУ), если $\deg' v = 2d/(d-1)$, и *удачной с избытком* или *очень удачной*, если $\deg' v > 2d/(d-1)$. *Удачная* цепь содержит только удачные вершины, *строго удачная* — только строго удачные, а *цепь с избытком* — удачные и очень удачные.

Случай 2. Если $L = d-1$, то по лемме 6 все вершины графа $G_1 \setminus V_0$ удачны. В стандартном случае имеем

$$2m = 2(d-1) - 0 + \frac{2d(n-d)}{d-1} = \frac{2dn-4d-2}{d-1},$$

что приводит к противоречию $2 = -2$. В трудном случае используем утверждение 3:

$$2m = 2(d-1) - \frac{4}{d-1} + \frac{2d(n-d)}{d-1} = \frac{2dn-4d-2}{d-1},$$

откуда $0 = 0$, так что все вершины графа $G_1 \setminus V_0$ СУ и нет цепей с избытком.

В доказательстве леммы 6 показано (см. разд. 7), что избыток $C = 4/(d-1)$ используется, только если $\deg v_0 = 3$, поэтому для поиска графа G'' рассмотрим в качестве подграфа G_2 нить T с концом v_0, положив $V_0 = \{\overline{v}_0, \widetilde{v}_0, v_0'\}$ (рис. 8).

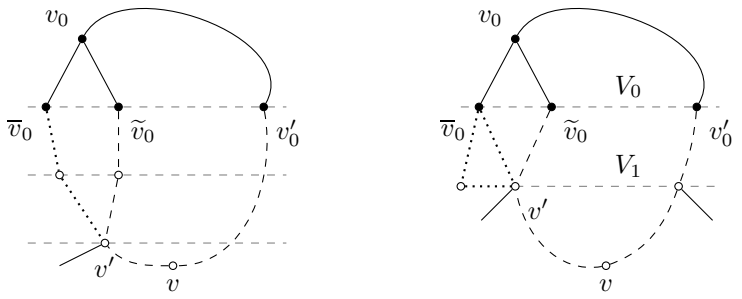


Рис. 8. Варианты цепи $\widetilde{v}_0 \dots v \dots v_0'$ длины $d+1$

Пусть для любой вершины v графа G_1 длина цепи $\overline{v}_0 \dots v \dots v_0'$ или $\widetilde{v}_0 \dots v \dots v_0'$ с рёбрами из G_1 не превосходит $2d+1-d = d+1$. Сформулируем общую схему МПВ для всех вершин графа G_1, которую с небольшими изменениями будем применять и далее (такая же схема применялась в [12]).

Правило 1 (П1). Каждое ребро v_0v или $v_0'v$, где $v \in V_1$, даёт вклад 2 вершине v и не даёт никакого вклада вершинам v_0 и v_0'.

Пусть вершина $u \in V_p$, $1 \leqslant p \leqslant q-1$, получает суммарный вклад α от соседей из V_{p-1} и смежна $k \geqslant 1$ соседям из V_{p+1}. Тогда выполнено

Правило 2 (П2). Каждое ребро uv для $u \in V_p$ и $v \in V_{p+1}$ даёт вершине u вклад $\alpha_u = \frac{1}{k}(2d/(d-1) - \alpha)$, если $\alpha < 2d/(d-1)$, и $\alpha_u = 0$, если $\alpha \geqslant 2d/(d-1)$, а вершине v — вклад $\alpha_v = 2 - \alpha_u$.

Правило 3 (П3). Каждое ребро uv, где $\{u, v\} \subseteq V_p$, $1 \leqslant p \leqslant q$, даёт вклад 1 вершинам u и v.

Заметим, что $q \leqslant \lceil d/2 \rceil$. Тогда, применяя П1–П3, для цепи $\overline{v}_0 \dots v \dots v'_0$ длины не более $l \leqslant d-1$ имеем $q \leqslant \lfloor (d-1)/2 \rfloor$, так что для неё не выполнено условие СУ и эта цепь с избытком, поэтому таких цепей нет. Если $l = d > L$, то существует ребро вне цепи $\overline{v}_0 \dots v \dots v'_0$, инцидентное внутренней вершине этой цепи, которое по П2 передаёт цепи ненулевой вклад, так что условие СУ также нарушается.

Рассмотрим цепь $\widetilde{v}_0 \dots v \dots v'_0$ длины $d+1$. По правилам МПВ для выполнения СУ необходимо, чтобы суммарный вклад всех d внутренних вершин цепи был равен $2d+2+2/(d-1)$. Все рёбра цепи дают суммарный вклад $2d+2$, поэтому суммарный недостаток равен $2/(d-1)$. По П2 он может быть компенсирован двумя способами.

Все вершины цепи $\widetilde{v}_0 \dots v \dots v'_0$ неузловые, за исключением следующих: 1) 1-узел $v' \in V_2$, смежный вершине из V_3 вне указанной цепи (рис. 8а); 2) два 1-узла из V_1, один из которых, например v', лежит в V_1 (рис. 8б) (в обоих случаях цепь $\widetilde{v}_0 \dots v'_0$ показана штриховой линией). Однако в первом случае (рис. 8а) имеется цикл длины $2d+1$, содержащий вершины v' и $\overline{v}_0$, так что цепь $\overline{v}_0 \dots v'$ длины 2 нарушает СУ для v'. Во втором случае (рис. 8б) СУ нарушается либо ребром $\overline{v}_0 v'$, либо цепью $\overline{v}_0 \dots v'$ длины 2, поэтому искомых графов при $L = d-1$ нет.

Случай 3. Если $L = d > d-1$, то лемма 6 не применима. Пусть V_0 — множество концов d-нитей и $|V_0| = n_0$. Здесь два возможных случая: $n_0 = 2$ и $n_0 \geqslant 3$. В качестве G_1 рассматриваются наборы нитей длины не более $d-1$. Предположим, что, как и при $L = d-1$, любая вершина v графа G_1 принадлежит цепи $v_0 \dots v \dots v'_0$ длины не более $d+1$ (рис. 9).

При $n_0 = 2$ покажем, что все вершины графа $G_1 \setminus V_0$ удачны. Здесь цепь $v_0 \dots v \dots v'_0$ имеет длину $d+1$ и суммарный вклад $2d+2$ для всех вершин цепи. По П1–П3 вклад от рёбер вне цепи не менее $2/(d-1)$. Следовательно, цепь удачна, поэтому

$$m = d + \frac{d(n-d-1)}{d-1} = \frac{dn-2d}{d-1} > g(n,d)$$

и граф не экстремален.

Пусть $n_0 \geqslant 3$ и v_0, v'_0 — пара концов некоторой d-нити в G''. Допустим, что $v''_0 \notin \{v_0, v'_0\}$ — конец другой d-нити (рис. 9). Тогда для выполнения ограничения на диаметр второй конец $\overline{v}_0$ второй d-нити совпадает с v_0 или v'_0. Пусть $\overline{v}_0 = v'_0$, а v''_0 смежна не только с v_0, но и со всеми остальными вершинами из множества $V_0 \setminus v'_0$. Тем самым в V_0 имеется не менее $(n_0-1)(n_0-2)/2$ рёбер. По П1–П3, как и при $n_0 = 2$, все вершины графа $G_1 \setminus V_0$ удачны, а вершины графа $G_2 \setminus V_0$ принадлежат d-нитям, где

на $d-1$ вершин приходится d рёбер, так что они тоже удачны. Тогда

$$m = \frac{(n_0-1)(n_0-2)}{2} + \frac{d(n-n_0)}{d-1} = \frac{dn-2d-1}{d-1},$$

откуда $n_0 = 3$, так что в V_0 имеется ровно одно ребро $v_0''v_0$ и все вершины графа $G_1 \setminus V_0$ СУ. Если $G_1 \setminus V_0 = \varnothing$, то получаем граф на рис. 5. Покажем, что это единственный вариант при $L = d$ и других графов нет.

Пусть $G_1 \setminus V_0 \neq \varnothing$. Тогда для любой вершины v графа $G_1 \setminus V_0$ цепь $v_0 \dots v \dots v_0'$ длины $d+1$ СУ и по П1–П3 имеет вклад от рёбер вне этой цепи, равный $2/(d-1)$. Для доказательства того, что $G_1 \setminus V_0 = \varnothing$, достаточно найти хотя бы одну цепь с избытком в G_1.

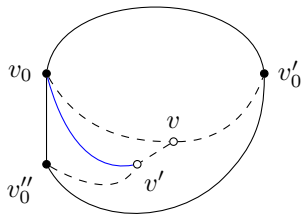


Рис. 9. Случай $G_1 \setminus V_0 \neq \varnothing$

Пусть имеется ребро vv' вне цепи $v_0 \dots v \dots v_0'$. Рассмотрим подграф — дерево с корнем v и висячими вершинами v_0, v_0', v_0'' (штриховая линия на рис. 9). Тогда подцепь $v_0'' \dots v' \dots v \dots v_0'$ и d-нить $v_0'v_0''$ образуют цикл длины $2d+1$, а подцепь $v_0' \dots v \dots v' \dots v_0''$, ребро $v_0''v_0$ и d-нить $v_0 \dots v_0'$ — цикл длины $2d+2$, что нарушает ограничение на диаметр d. Следовательно, в указанном дереве имеется ещё хотя бы одна ветвь длины не более $d-1$, например выделенная ветвь $v_0 \dots v'$. С добавленной ветвью в дереве имеется не более $x \leqslant 3d$ рёбер и $x-3$ вершин из G_1. Условие СУ выполняется, только если $x = 3d$, но цепь $v_0 \dots v$ не может быть d-нитью, так что ей инцидентно ещё хотя бы одно ребро графа G_1, дающее вклад, который нарушает СУ. Полученное противоречие доказывает равенство $G_1 \setminus V_0 = \varnothing$.

Случай 4. Если $L \geqslant d+1$, то в качестве G_2 выберем нить $T' \subseteq T$ длины $l = d+1$ с концами v_0, v_0'. Тогда

$$m_1 = \frac{dn-2d-1}{d-1} - (d+1) = \frac{d(n-(d+2))}{d-1},$$

так что $n-(d+2)$ вершинам графа $G_1 \setminus V_0$ инцидентно $d(n-(d+2))/(d-1)$ рёбер. Эти вершины образуют цепи, которые попарно не пересекаются

и являются d-нитями, так как иначе в G_1 имелись бы цепи длины меньше d, что нарушает условие СУ. Соответствующий граф представлен на рис. 1.

Для окончания доказательства теоремы 2 следует отметить, что в случаях 2–4 предполагается, что все вершины подграфов G_1 и G_2 принадлежат циклу длины не более $2d+1$. Если это не так, то для выполнения ограничения на диаметр d в G_1 должны существовать цепи длины не более d, имеющие инцидентные рёбра вне цепи, которые в соответствии с П1–П3 нарушают условия СУ. Теорема 2 доказана.

7. Дополнения к доказательствам теорем

Доказательство леммы 5. Сначала для любой вершины $v \in V_p$ рассмотрим вклад ребра $v_p v_{p+1}$, где $v_{p+1} \in V_{p+1}$. На рис. 10 показаны все виды узлов v_p степени 3.

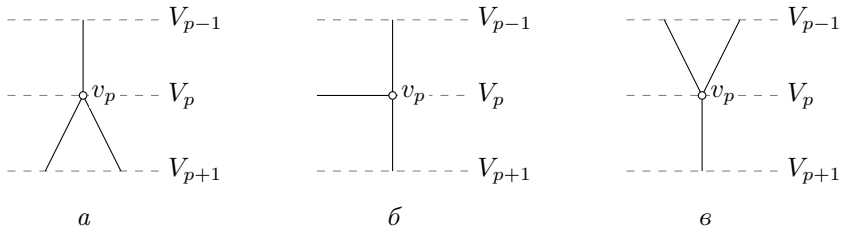


Рис. 10. Узлы степени 3: а) 1-узел; б) 2-узел; в) 3-узел

Пусть вершина v_p получает вклад α от соседа из V_{p-1}. Применим П1–П3. Если $\deg v_p = 2$, то вклад ребра $v_p v_{p+1}$ вершине v_p в силу её удачности равен $2d/(d-1)-\alpha$, а вклад ребра $v_p v_{p+1}$ вершине v_{p+1} равен $\overline{\alpha} = 2-(2d/(d-1)-\alpha) = \alpha - 2/(d-1)$. Пусть v_p узел степени $k \geqslant 3$. Определим, при каком типе узла вклад $\overline{\alpha}$ минимален.

Положим $\overline{\alpha} = \overline{\alpha}_i$, если v_p — i-узел степени k, $i = 1,2,3$. Если v_p — 1-узел (рис. 10а), то v_p смежна с $k-1$ вершинами из V_{p+1} и

$$\overline{\alpha}_1 = 2 - \frac{1}{k-1}\Big(\frac{2d}{d-1} - \alpha\Big).$$

Если v_p — 2-узел (рис. 10б), то v_p смежна хотя бы с одной вершиной из V_p и не более $k-2$ вершинами из V_{p-1}. Получаем

$$\overline{\alpha}_2 \geqslant 2 - \frac{1}{k-2}\Big(\frac{2d}{d-1} - \alpha - 1\Big).$$

Если v_p — 3-узел (рис. 10в), то v_p смежна хотя бы с двумя вершинами из V_{p-1}, сумма вклада от которых не менее чем 2α, и имеет не более

$k-2$ смежных вершин из V_{p-1}. Тогда

$$\overline{\alpha}_3 \geqslant 2 - \frac{1}{k-2}\Big(\frac{2d}{d-1} - 2\alpha\Big).$$

Очевидно, что $\overline{\alpha}_i$, $i = 1, 2, 3$, минимально при $k = 3$.

Утверждение 5. *Если* $\alpha \geqslant \frac{2}{3} + \frac{2}{3(d-1)}$, *то* $\overline{\alpha}_1 = \min\{\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3\}$.

Доказательство. Покажем, что $\overline{\alpha}_2 - \overline{\alpha}_1 \geqslant 0$ и $\overline{\alpha}_3 - \overline{\alpha}_1 \geqslant 0$. Имеем

$$\overline{\alpha}_2 - \overline{\alpha}_1 \geqslant 2 - \frac{\frac{2d}{d-1} - \alpha - 1}{k-2} - 2 + \frac{\frac{2d}{d-1} - \alpha}{k-1} \geqslant$$
$$\geqslant \frac{\alpha + k - 1 - \frac{2d}{d-1}}{(k-1)(k-2)} \geqslant \frac{\frac{2d}{3(d-1)} + 2 - \frac{2d}{d-1}}{(k-1)(k-2)} \geqslant$$
$$\geqslant \frac{2d-6}{3(d-1)(k-1)(k-2)} > 0,$$

а также

$$\overline{\alpha}_2 - \overline{\alpha}_1 \geqslant 2 - \frac{\frac{2d}{d-1} - 2\alpha}{k-2} - 2 + \frac{\frac{2d}{d-1} - \alpha}{k-1} \geqslant \frac{\alpha k - \frac{2d}{d-1}}{(k-1)(k-2)} \geqslant 0.$$

Утверждение 5 доказано.

Таким образом, для 1-узла v_p вклад ребра $v_p v_{p+1}$ вершине v_{p+1} минимальный. Это значит, что если в цепи все узлы являются 1-узлами, то конец цепи $v_0 \dots v_p v_{p+1} \dots v_{\overline{p}}$ получает минимальный последний вклад. Следовательно, условная степень $\deg' v_{\overline{p}}$, состоящая из суммы таких последних вкладов, в этом случае минимальна, поэтому для всех других типов узлов значение $\deg' v_{\overline{p}}$ будет не меньше. Лемма 5 доказана.

Доказательство леммы 6. Из правил П1–П3 получаем, что все внутренние вершины графа $G_1 \setminus V_0$ будут удачными, и нам остаётся проанализировать последние вклады для всех концов цепей и их условные степени. Например, для сильной особой v_4 на рис. 7а, слабой особой v_4', ординарной v_4 на рис. 7б.

Необходимо показать, что вершина v_q цепи $v_0 \dots v_p v_{p+1} \dots v_q$ удачна. Из удачности v_q следует удачность $v_{\overline{p}}$ при $\overline{p} \leqslant q$, что доказывает лемму 6. Так как $d(v_q, v) = d$ для любой вершины v нити T наибольшей длины в графе G, то расстояние от v_q до середины T максимально, поэтому $q = d - \lfloor L/2 \rfloor$, и дальнейшее рассмотрение зависит от чётности L.

Определим сумму наименьших последних вкладов для концов цепей. Пусть в цепи $v_0 \dots v_p v_{p+1} \dots v_q$ имеется $k \geqslant 0$ узлов. Положим $z' = v_0$, $z = v_q$, а саму цепь обозначим через $z'z$. Пусть $\alpha_{zz'}$ — последний вклад цепи $z'z$ вершине z. По лемме 5 все узлы цепи $z'z$ считаем 1-узлами, которые обозначим через $\overline{v}_1, \overline{v}_2, \dots, \overline{v}_k$. Пусть x_i, $i \in \overline{1, k+1}$, — длины нитей

в цепи $z'z$ с концами в вершинах v_0, v_q и 1-узлах $\overline{v}_i$, $i \in \overline{1,k}$. Нумерацию 1-узлов произведём в направлении от z к z', так что длина нити $z \dots \overline{v}_1$ равна x_1, а $\overline{v}_k \dots z' - x_{k+1}$. Далее будем полагать, что $\deg \overline{v}_1 = a \geqslant 3$ и $\deg \overline{v}_i = 3$ при $i \in \overline{2,k}$. Оценим вклад $\alpha_{zz'}$.

Утверждение 6. *Имеют место неравенства*

$$\alpha_{zz'} \geqslant \frac{2d-2x_1}{d-1} - \sum_{i=2}^{k+1} \frac{2x_i}{2^{i-2}(a-1)(d-1)} \geqslant \frac{2d}{d-1} - \sum_{i=1}^{k+1} \frac{2x_i}{2^{i-1}(d-1)}. \quad (3)$$

Доказательство. Воспользуемся индукцией по числу k 1-узлов в цепи $z'z$. Если $k=1$, то zz' состоит из нитей $z' \dots \overline{v}_1$ и $\overline{v}_1 \dots z$ длины x_2 и x_1 соответственно. Из П2 следует, что от первой нити вершине $\overline{v}_1$ поступает вклад $\alpha = 2 - 2(x_2-1)/(d-1)$, а от второй —

$$\frac{1}{a-1}\left(\frac{2d}{d-1} - \alpha\right) = \frac{2x_2}{(d-1)(a-1)}.$$

Отсюда получаем первое из неравенств (3):

$$\alpha_{zz'} \geqslant 2 - \frac{2(x_1-1)}{d-1} - \frac{2x_2}{(d-1)(a-1)} = \frac{2d-2x_1}{d-1} - \frac{2x_2}{(d-1)(a-1)}.$$

Пусть при k утверждение верно, рассмотрим случай $k+1$, в котором на цепи $z'z$ имеются 1-узлы $\overline{v}_{k+1}, \dots, \overline{v}_2$ степени 3. По предположению индукции вклад нити $\overline{v}_2 \dots \overline{v}_1$ вершине $\overline{v}_1$ равен

$$\alpha = \frac{2d-2x_2}{d-1} - \sum_{i=3}^{k+1} \frac{2x_i}{2^{i-2}(d-1)}.$$

Тогда из П2 следует оценка

$$\alpha_{zz'} \geqslant 2 - \frac{2(x_1-1)}{d-1} - \frac{\frac{2d}{d-1} - \alpha}{a-1} = \frac{2d-2x_1}{d-1} - \sum_{i=2}^{k+1} \frac{2x_i}{2^{i-2}(a-1)(d-1)}.$$

Второе из неравенств (3) очевидно и при $a=3$ даёт равенство. Утверждение 6 доказано.

Замечание 1. При $k=0$ цепь $z'z$ длины x_1 состоит только из неузлов и в силу П1 и П2 имеем

$$\alpha_{zz'} \geqslant 2 - \frac{2(x_1-1)}{d-1} = \frac{2d-2x_1}{d-1}.$$

Пусть вершина z сильная. Тогда в графе G имеется ещё одна цепь $v'_0 \dots v'_p v'_{p+1} \dots v_q$, $v_q = z$, содержащая $s \geqslant 0$ узлов, которые по лемме 5 также можно считать 1-узлами. Положим $z'' = v'_0$, саму цепь обозначим через $z''z$, а последний вклад цепи $z''z$ вершине z — через $\alpha_{zz''}$. Пусть y_j, $j \in \overline{1,s+1}$, — длины нитей в цепи $z''z$ и первый, считая от z, 1-узел

цепи $z''z$ имеет степень $b \geqslant 3$. Аналогично неравенствам (3) получаем нижнюю оценку для $\alpha_{zz''}$.

Утверждение 7. *Имеют место неравенства*

$$\alpha_{zz''} \geqslant \frac{2d-2y_1}{d-1} - \sum_{j=2}^{s+1} \frac{2y_j}{2^{j-2}(b-1)(d-1)} \geqslant \frac{2d}{d-1} - \sum_{j=1}^{s+1} \frac{2y_j}{2^{j-1}(d-1)}. \quad (4)$$

Далее рассмотрим систему из неравенств, ограничивающих сумму вкладов конца цепи z и длину нитей цепи, и уравнения, накладывающего условие на длину всей цепи:

$$\begin{gathered} \deg' z \geqslant \alpha_{zz'} + \alpha_{zz''}, \\ \sum_i x_i = \sum_j y_j = d - \lfloor L/2 \rfloor, \\ x_i,\, y_j \leqslant L, \quad i \in \overline{1, k+1},\, j \in \overline{1, s+1}. \end{gathered} \quad (5)$$

Решим систему (5) в зависимости от чётности L для доказательства удачности вершины z.

7.1. Чётная наибольшая длина нити в графе G. Пусть L чётно. Тогда $\sum x_i = \sum y_j = d - L/2$, а в виду неравенств (3) и (4) получаем

$$\deg' z \geqslant \frac{4d}{d-1} - \sum_{i=1}^{k+1} \frac{2x_i}{2^{i-1}(d-1)} - \sum_{j=1}^{s+1} \frac{2y_j}{2^{j-1}(d-1)} = \frac{4d}{d-1} -$$
$$- \frac{x_1+y_1}{d-1} - \frac{2d-L}{d-1} + \sum_{i\geqslant 3} \frac{x_i(1-1/2^{i-2})}{d-1} + \sum_{j\geqslant 3} \frac{y_j(1-1/2^{j-2})}{d-1}. \quad (6)$$

Заметим, что суммы по i и j в (6) неотрицательны.

Стандартные случаи. Если вершина z сильная особая, то имеем $x_1 + y_1 \leqslant L$ и из (6) получаем

$$\deg' z \geqslant \frac{4d}{d-1} - \frac{L}{d-1} - \frac{2d-L}{d-1} = \frac{2d}{d-1},$$

так что z удачна.

Если вершина z сильная ординарная, то рассмотрим два возможных случая: (а) $L \leqslant (2d-2)/3$, (б) $L \geqslant (2d-2)/3 + 2 = (2d+4)/3$.

(а) Сложим (3) и (4), применяя неравенство $x_1 + y_1 \leqslant 2L$:

$$\alpha_{zz'} + \alpha_{zz''} \geqslant \frac{4d}{d-1} - \frac{2L}{d-1} - \frac{2d-L}{d-1} = \frac{2d-L}{d-1} \geqslant \frac{4d+2}{3(d-1)}.$$

Для ординарной вершины z также выполнено правило П3, поэтому если z имеет ровно двух соседей из V_{q-1} и одного из V_q, то

$$\deg' z \geqslant \frac{4d+2}{3(d-1)} + 1 \geqslant \frac{2d}{d-1} + \frac{1}{3},$$

иначе z имеет не менее трёх соседей из V_{q-1} и

$$\deg' z \geqslant 3\alpha_{zz'} \geqslant \frac{2d+1}{d-1}.$$

Далее будем использовать избытки таких вершин для выполнения удачности смежных вершин.

(б) Преобразуем (3) и (4), применив равенства $\sum\limits_i x_i = \sum\limits_j y_j = d - L/2$:

$$\alpha_{zz'} + \alpha_{zz''} \geqslant \frac{1}{d-1}\Big(4d - 2\Big(\sum_i x_i + \sum_j y_j\Big) + \\ + \sum_{i\geqslant 2} x_i(2 - 1/2^{i-2}) + \sum_{j\geqslant 2} y_j(2 - 1/2^{j-2})\Big) \geqslant \\ \geqslant \frac{4d}{d-1} - \frac{4\big(d - \frac{L}{2}\big)}{d-1} = \frac{2L}{d-1} \geqslant \frac{4d+8}{3(d-1)}.$$

Поскольку $\deg' z \geqslant \alpha_{zz'} + \alpha_{zz''} + 1$, получаем $\deg' z \geqslant 2d/(d-1)$, если вершина z имеет ровно двух соседей из V_{q-1}, иначе z имеет не менее трёх соседей из V_{q-1} и

$$\deg' z \geqslant \frac{3L}{d-1} \geqslant \frac{2d+4}{d-1},$$

так что условие удачности имеет избыток, как и в случае (а).

Далее проведём анализ различных вариантов слабой вершины z.

Если вершина z слабая особая и не лежит на нити T, то рассмотрим два возможных варианта: (1с) z смежна со слабой особой $\overline{z}$, (2с) z смежна с ординарной $\overline{z}$.

(1с) Так как $z \notin V(T)$, то $x_1 + y_1 \leqslant L - 2$. В силу неравенств (3), (4) и наличия ребра $z\overline{z}$ имеем

$$\deg' z + \deg' \overline{z} \geqslant \alpha_{zz'} + \alpha_{\overline{z}z''} + 2 \geqslant \frac{4d}{d-1} - \frac{L-2}{d-1} - \frac{2d-L}{d-1} + 2 = \frac{4d}{d-1},$$

так что вершины z и $\overline{z}$ удачны.

(2с) Вновь рассмотрим два случая по величине L: (а) $L \leqslant (2d-2)/3$, (б) $L \geqslant (2d-2)/3 + 2 = (2d+4)/3$.

(а) Введём дополнительное правило, которое использует избыток ординарных вершин и корректирует правило П3.

Правило 4 (П4). Если $L \leqslant (2d-2)/3$, то ребро $z\overline{z}$, где $z \in V_q \setminus V(T)$ слабая особая, а $\overline{z} \in V_q$ ординарная, даёт вклад $\frac{4}{3} - \frac{1}{2(d-1)}$ вершине z и вклад $\frac{2}{3} - \frac{1}{2(d-1)}$ вершине $\overline{z}$. Для остальных смежных вершин из V_p, $1 \leqslant p \leqslant q$, выполнено правило П3.

Используя неравенства (3), получаем

$$\alpha_{zz'} \geqslant \frac{2d}{d-1} - \sum_{i\geqslant 1} \frac{2x_i}{2^{i-1}(d-1)} = \\ = \frac{2d}{d-1} - \frac{x_1}{d-1} - \sum_{i\geqslant 1} \frac{x_i}{d-1} + \sum_{i\geqslant 2} \frac{x_i(1-1/2^{i-2})}{d-1}.$$

Здесь последняя сумма неотрицательна, $\sum\limits_i x_i = d - L/2$, а из смежности вершин z и $\overline{z}$ следует, что $x_1 \leqslant L-2$, поэтому

$$\alpha_{zz'} \geqslant \frac{2d}{d-1} - \frac{L-2}{d-1} - \frac{d-L/2}{d-1} = \frac{d-L/2+2}{d-1}.$$

Применяя П4, с учётом неравенства $L \leqslant (2d-2)/3$ для z получаем

$$\deg' z \geqslant \frac{d-L/2+2}{d-1} + \frac{4}{3} - \frac{1}{2(d-1)} \geqslant \frac{2d}{d-1},$$

а для ординарной вершины $\overline{z}$ —

$$\deg' \overline{z} \geqslant \frac{d-L/2}{d-1} + 2\left(\frac{2}{3} + \frac{1}{2(d-1)}\right) \geqslant \frac{2d}{2(d-1)}.$$

(б) Для этого случая в дополнение к П1–П3 понадобится

Правило 4′ (П4′). Если $L \geqslant (2d+4)/3$, то для слабой особой вершины $z \in V_q$ и ординарной $\overline{z} \in V_q$ ребро $z\overline{z}$ даёт вклад $4/3$ вершине z и $2/3$ вершине $\overline{z}$. Для остальных смежных вершин из V_p, $1 \leqslant p \leqslant q$ выполнено правило П3.

Имеем

$$\deg' z \geqslant \frac{2d}{d-1} - \frac{2(d-L/2)}{d-1} + \frac{4}{3} = \frac{L}{d-1} + \frac{4}{3} \geqslant \frac{2d}{d-1},$$
$$\deg' \overline{z} \geqslant \frac{L}{d-1} + 2 \cdot \frac{2}{3} \geqslant \frac{2d}{d-1},$$

что доказывает удачность вершин z и $\overline{z}$.

Заметим, что здесь не используется, что $z \notin V(T)$, и поэтому если $L \geqslant (2d+4)/3$, а вершина $z \in V(T)$ слабая особая и смежна с ординарной $\overline{z}$, то z и $\overline{z}$ также удачны.

При анализе (2c) показана удачность ординарной вершины $\overline{z}$, смежной со слабой особой z. Если ординарная $\overline{z}$ смежна другим вершинам, то её удачность также следует из П2–П3.

Определим оставшиеся трудные случаи: (1т) $z, \overline{z} \in V(T)$ — слабые особые смежные вершины, (2т) слабая особая вершина $z \in V(T)$ смежна ординарной $\overline{z}$ при $L \leqslant (2d-2)/3$. Рассмотрим важные подслучаи случаев (1т) и (2т), не использующие избыток C.

Утверждение 8. *Если* $\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j \leqslant 2$, *то в случаях* (1т) *и* (2т) *вершины* z *и* $\overline{z}$ *удачны.*

Доказательство. (1т) Поскольку $x_1 + y_1 = L - 1$ и $L \leqslant d-1$, имеем

$$\begin{aligned}\deg' z + \deg' \overline{z} &\geqslant \alpha_{zz'} + \alpha_{\overline{z}z''} + 2 \geqslant \\ &\geqslant \frac{4d}{d-1} - \frac{2(x_1+y_1)}{d-1} + 2 - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)} \geqslant \\ &\geqslant \frac{6d-2L}{d-1} - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)} \geqslant \\ &\geqslant \frac{4d+2}{d-1} - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)},\end{aligned}$$

и если $\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j \leqslant 2$, то $\deg' z + \deg' \overline{z} \geqslant 4d/(d-1)$.

(2т) Определим правило П4″, дословно повторяющее П4′, за исключением ограничения на L: здесь $L \leqslant (2d-2)/3$.

Поскольку $x_1 \leqslant L-1$, $y_1 \leqslant L$ и $\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j \leqslant 2$, имеем

$$\begin{aligned}\deg' z + \deg' \overline{z} &\geqslant \frac{4d}{d-1} + \frac{2}{3} - \\ &\quad - \frac{2(x_1+y_1)}{d-1} + 2 - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)} \geqslant \\ &\geqslant \frac{20d-12L-2}{3(d-1)} - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)} \geqslant \\ &\geqslant \frac{4d+2}{d-1} - \sum_{i\geqslant 2} \frac{x_i}{2^{i-2}(d-1)} - \sum_{j\geqslant 2} \frac{y_j}{2^{j-2}(d-1)},\end{aligned}$$

откуда $\deg' z + \deg' \overline{z} \geqslant 4d/(d-1)$. Утверждение 8 доказано.

Положим $\deg v_0 = r$, при этом либо $r = 3$, либо $r \geqslant 4$. Покажем, что случай $r \geqslant 4$ стандартный.

(1т) Пусть $r \geqslant 4$. Напомним, что в этом случае вершины z и $\overline{z}$ принадлежат нити T, концы которой обозначим через $\overline{v}_1$ и $\overline{v}_1'$. Так как вершины $\overline{v}_1$ и $\overline{v}_1'$ — 1-узлы, при этом $\deg \overline{v}_1 = a \geqslant r \geqslant 4$, $\deg \overline{v}_1' = b \geqslant r \geqslant 4$, для оценки $\deg' z + \deg' \overline{z}$ применим (3)–(5) и неравенство $\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j \geqslant 3$:

$$\deg' z + \deg' \overline{z} \geqslant \frac{4d - 2(x_1+y_1)}{d-1} - \\ - \sum_{i\geqslant 2} \frac{2x_i}{2^{i-2}\cdot 3(d-1)} - \sum_{j\geqslant 2} \frac{2y_j}{2^{j-2}\cdot 3(d-1)} \geqslant \\ \geqslant \frac{4d-(L-1)}{d-1} - \frac{2d-L}{d-1} + 2 + \\ + \sum_{i\geqslant 2} x_i\Big(1 - \frac{2}{2^{i-2}\cdot 3(d-1)}\Big) + \sum_{j\geqslant 2} y_j\Big(1 - \frac{2}{2^{j-2}\cdot 3(d-1)}\Big) \geqslant \\ \geqslant \frac{4d-1}{d-1} + \frac{\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j}{3(d-1)} \geqslant \frac{4d-1}{d-1} + \frac{3}{3(d-1)} \geqslant \frac{4d}{d-1}.$$

(2т) Вновь $r \geqslant 4$. В этом случае ординарная вершина $\overline{z}$ является концом нити T, поэтому $\deg \overline{z} \geqslant 4$. Применим правило П4′ и оценим $\deg' z$, используя ограничение $L \leqslant (2d-2)/3$:

$$\deg' z \geqslant \frac{2d-2x_1}{d-1} - \sum_{i\geqslant 2} \frac{2x_i}{2^{i-2}\cdot 3(d-1)} + \frac{4}{3} \geqslant \\ \geqslant \frac{2d-(L-1)}{d-1} - \frac{d-L/2}{d-1} + \frac{4}{3} \geqslant \frac{2d+4}{3(d-1)} + \frac{4}{3} = \frac{2d}{d-1}.$$

При этом для $\overline{z}$ имеем

$$\deg' \overline{z} \geqslant \frac{2d-L}{d-1} - \frac{d-L/2}{d-1} + \frac{3\cdot 2}{3} \geqslant \frac{2d}{d-1}.$$

Тем самым вершины z и $\overline{z}$ удачны.

Трудные случаи и использование избытка. Пусть $r = 3$ и вершина v_0 смежна с двумя вершинами из V_1. Тогда, возможно, имеются две различные цепи вида $v_0 \dots v_q$, относящиеся к трудным случаям. Конец такой цепи $z = v_q$ может быть неудачен, поэтому его вкладу добавим избыток $C/2 = 2/(d-1)$. Из утверждения 8 следует, что здесь $\sum\limits_{i\geqslant 2} x_i + \sum\limits_{j\geqslant 2} y_j \geqslant 3$.

(1т) С учётом сказанного получаем

$$\deg' z + \deg' \overline{z} \geqslant \alpha_{zz'} + \alpha_{\overline{z}z''} + 2 \geqslant \frac{6d-2}{d-1} - \frac{2(x_1+y_1)}{d-1} - \frac{x_2+y_2}{d-1} -$$

$$-\sum_{i\geqslant 3}\frac{x_i}{2^{i-2}(d-1)}-\sum_{j\geqslant 3}\frac{y_j}{2^{j-2}(d-1)}\geqslant\frac{6d-2-(L-1)-2(d-L/2)}{d-1}+$$
$$+\sum_{i\geqslant 3}\frac{x_i(1-1/2^{i-2})}{d-1}+\sum_{j\geqslant 3}\frac{y_j(1-1/2^{j-2})}{d-1}\geqslant\frac{4d-1}{d-1}+\frac{x_3+y_3}{2(d-1)},$$

откуда удачность вершины z следует без применения избытка в случае $x_3+y_3\geqslant 2$. Если $x_3+y_3\leqslant 1$ и

$$\sum_{i\geqslant 4}x_i(1-1/2^{i-2})+\sum_{j\geqslant 4}y_j(1-1/2^{j-2})=0,$$

то недостаток для $\deg' z+\deg'\overline{z}$ составляет не более $1/(d-1)$, следовательно, избыток, добавляемый для удачности вершин z и $\overline{z}$, не может быть меньше $1/(d-1)$.

Пусть $x_3=1$, $y_3=0$ и цепь zz' состоит из трёх нитей: $z\overline{v}_1$ длины x_1, $\overline{v}_1\overline{v}_2$ длины x_2, $\overline{v}_2z'$ длины $x_3=1$, а цепь $\overline{z}z''$ из двух нитей: $\overline{z}\,\overline{u}_1$ длины $y_1,\overline{u}_1z''$ длины y_2, где $\overline{v}_1,\overline{v}_2,\overline{u}_1$ — 1-узлы. В цепях с недостатком некорректно считать, что $\deg\overline{v}_2=3$. Действительно, если $\overline{v}_2\in V_1$, то вершина $\overline{v}_2$ смежна с $\deg\overline{v}_2-1$ вершинами из V_2 и подцепи, начинающиеся в $\overline{v}_2$, могут заканчиваться концами, у которых также имеется недостаток. Стало быть, избыток $C/2=2/(d-1)$ для каждой подцепи следует разделить на $\deg\overline{v}_2-1$ частей, и остаётся открытым вопрос о его достаточности для удачности конца цепи. Те же рассуждения необходимо привести для вершин $\overline{v}_1$ и $\overline{u}_1$, поэтому для конца z, например, избыток будет составлять

$$\frac{C}{2(\deg\overline{v}_1-1)(\deg\overline{v}_2-1)}.$$

Тогда, полагая $\deg\overline{v}_1=a$, $\deg\overline{v}_2=b$, $\deg\overline{u}_1=c$, применяя (3), (4) и умножая избыток на $d-1$, приходим к тому, что можно ликвидировать недостаток величины $1/(d-1)\cdot(d-1)=1$:

$$1\cdot\Big(1-\frac{2}{(a-1)(b-1)}\Big)+x_2\Big(1-\frac{2}{a-1}\Big)+$$
$$+y_2\Big(1-\frac{2}{c-1}\Big)+\frac{2}{(a-1)(b-1)}\geqslant 1.$$

Поскольку все слагаемые в левой части неравенства неотрицательны, получаем $1\geqslant 1$, так что вершины z и $\overline{z}$ удачны.

Пусть $x_3=y_3=0$. Снова запишем условие удачности с учётом того, что $\overline{v}_2$ нет:

$$x_2\Big(1-\frac{2}{a-1}\Big)+y_2\Big(1-\frac{2}{c-1}\Big)+\frac{2}{a-1}\geqslant 1.$$

Так как $x_2 \geqslant 1$, имеем $1 \geqslant 1$, поэтому вершины z и $\overline{z}$ удачны.

(2т) Применяя П1–П4, получаем удачность ординарной $\overline{z}$:

$$\deg' \overline{z} \geqslant \frac{d - L/2}{d-1} + 2\Big(\frac{2}{3} + \frac{1}{2(d-1)}\Big) \geqslant \frac{2d}{d-1},$$

при этом для слабой особой вершины z имеем

$$\deg' z \geqslant \frac{d - L/2 + 1}{d-1} + \frac{4}{3} - \frac{1}{2(d-1)} + \sum_{i \geqslant 3} \frac{x_i(1 - 1/2^{i-2})}{d-1} \geqslant$$
$$\geqslant \frac{2d}{d-1} - \frac{1}{2(d-1)} + \sum_{i \geqslant 3} \frac{x_i(1 - 1/2^{i-2})}{d-1},$$

так что z удачна, если $x_3 \geqslant 1$. Если $x_3 = 0$, то недостаток составляет не более $\frac{1}{2(d-1)}$. Пусть вершина $\overline{v}_1$ делит цепь на две нити $z\overline{v}_1$ и $\overline{v}_1 z'$ и попрежнему $\deg \overline{v}_1 = a$. С учётом этого запишем условие удачности, как это сделано в случае (1т), добавив избыток $C/2$:

$$\frac{C}{2(a-1)} + x_2\Big(1 - \frac{2}{a-1}\Big) \geqslant \frac{1}{2},$$

т. е. $1 \geqslant 1/2$, следовательно, вершина z удачна.

Замечание 2. При чётном L во всех случаях оказываются справедливыми неравенства $\alpha > \alpha_{zz'} \geqslant \frac{2}{3} + \frac{2}{3(d-1)}$, так что имеет место утверждение 5.

7.2. Нечётная наибольшая длина нити в графе G. Пусть L нечётно. Тогда в нити T есть две смежные вершины, расположенные от $z' = v_0$ и $z'' = v_0'$ на расстоянии $(L-1)/2$ (на рис. 11 они обведены дополнительно), а равенства в (5) принимают вид $\sum\limits_i x_i = \sum\limits_j y_j = d - (L-1)/2$.

Рассмотрим два варианта в зависимости от $\deg v_0 = r$: (1н) $r = 3$, $z \notin T$ или $r \geqslant 4$, (2н) $r = 3$, $z \in T$.

(1н) Разделим этот случай на два подслучая: (а) z сильная, (б) z слабая. Будем применять П1–П3 и неравенства (3) и (4).

(а) Если вершина z сильная особая, то

$$\deg' z \geqslant \alpha_{zz'} + \alpha_{zz''} \geqslant \frac{4d}{d-1} - \frac{L-1}{d-1} - \frac{2d - L + 1}{d-1} = \frac{2d}{d-1}.$$

В случае $r \geqslant 4$ ближайшие к вершине z 1-узлы $\overline{v}_1$ и $\overline{v}_2$ имеют степени $\deg' \overline{v}_1 \geqslant 4$ и $\deg' \overline{v}_2 \geqslant 4$, поэтому

$$\deg' z \geqslant \alpha_{zz'} + \alpha_{zz''} \geqslant$$

$$\geqslant \frac{4d}{d-1} - \frac{L}{d-1} - \frac{2d-L+1}{d-1} + \frac{x_2+y_2+2(x_3+y_3)}{3(d-1)}.$$

Если при этом $x_3+y_3 \geqslant 1$ и $x_2+y_2 \geqslant 1$ или $x_3+y_3=0$ и $x_2+y_2 \geqslant 3$, то вершина z удачна. Если $x_2+y_2 \leqslant 2$, то

$$\deg' z \geqslant \frac{4d-2L-(x_2+y_2)}{d-1} \geqslant \frac{4d-2d+2-2}{d-1},$$

так что вершина z вновь удачна.

Если вершина z сильная ординарная, при этом не является концом нити T и $L \leqslant (2d-1)/3$, то

$$\alpha_{zz'} + \alpha_{zz''} \geqslant \frac{4d-2(L-1)-(2d-L+1)}{d-1} = \frac{2d-L+1}{d-1} \geqslant \frac{4d+4}{3(d-1)},$$

так что

$$\deg' z \geqslant \alpha_{zz'} + \alpha_{zz''} + 1 \geqslant \frac{4d+4}{3(d-1)} + 1 \geqslant \frac{2d}{d-1}$$

либо

$$\deg' z \geqslant 3\alpha_{zz'} \geqslant \frac{3(2d-L+1)}{2(d-1)} \geqslant \frac{2d+2}{d-1}.$$

Если же $L \geqslant (2d+5)/3$, то

$$\alpha_{zz'} + \alpha_{zz''} \geqslant \frac{4d-4(d-(L-1)/2)}{d-1} \geqslant \frac{2L-2}{d-1} \geqslant \frac{4d+4}{3(d-1)},$$

откуда

$$\deg' z \geqslant \alpha_{zz'} + \alpha_{zz''} + 1 \geqslant \frac{2d}{d-1}$$

либо

$$\deg' z \geqslant 3\alpha_{zz'} \geqslant \frac{2d+2}{d-1}.$$

Если сильная ординарная вершина z — конец нити T, то $\deg z \geqslant 4$, так что z удачна.

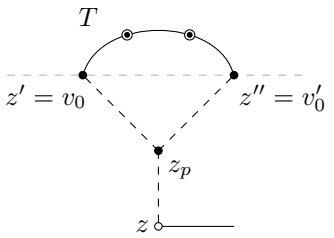


Рис. 11. Случай $r=3$, $z \notin T$ или $r \geqslant 4$ для слабой z

(б) Поскольку z слабая, только одна цепь zz' имеет длину $d-(L-1)/2$. Для выполнения условия $d(z,v) \leqslant d$ для $v \in V(T)$ необходимо, чтобы

в цепи zz' был 3-узел z_p такой, что $p = d(z_p, z') = d(z_p, z'')$ (рис. 11). Таким образом, цепи $z'z$ и $z''z$ разбиваются на две подцепи каждая: $z'z_p$, z_pz и $z''z_p$, z_pz соответственно.

Введём дополнительные обозначения. Пусть γ — длина цепи zz_p, значения α_p и α'_p — вклады вершине z_p от $v_{p-1}z_p$ и $v'_{p-1}z_p$, а α''_p — вклад вершине z_p от z_pv_{p+1}. Длины нитей, из которых состоят цепи z_pz' и z_pz'', обозначим через x'_i и y'_j. Ближайшая задача состоит из двух частей: 1) получить верхнюю оценку α''_p и нижнюю — для $2 - \alpha''_p$, 2) вычислить $\alpha_{zz'}$ и оценить $\deg' z$. Запишем для $\deg' z_p$ систему, аналогичную (5):

$$\begin{gathered} \deg' z_p \geqslant \alpha_p + \alpha'_p + \alpha''_p, \\ \sum_i x_i = \sum_j y_j = p = d - \frac{L-1}{2} - \gamma, \\ x'_i,\, y'_j \leqslant L, \quad i \in \overline{1, k+1},\, j \in \overline{1, s+1}. \end{gathered} \tag{7}$$

Применим (3), (4), П1–П3 и ограничения из (7). Если $L \leqslant (2d-3)/3$, то

$$\begin{aligned} \alpha_p + \alpha'_p &\geqslant \frac{4d - x'_1 - y'_1}{d-1} - \frac{\sum\limits_i x'_i + \sum\limits_j y'_j}{d-1} + \\ &\quad + \sum_{i \geqslant 2} \frac{x'_i(1 - 1/2^{i-2})}{d-1} + \sum_{j \geqslant 2} \frac{y'_j(1 - 1/2^{j-2})}{d-1} \geqslant \\ &\geqslant \frac{4d - 2L - (2d - (L-1) - 2\gamma)}{d-1} \geqslant \frac{2d - L - 1 + 2\gamma}{d-1} \geqslant \frac{4d + 6\gamma}{3(d-1)}, \end{aligned}$$

так что

$$\alpha''_p \leqslant \frac{2d}{d-1} - \alpha_p - \alpha'_p \leqslant \frac{2d - 6\gamma}{3(d-1)},$$

откуда получаем

$$\alpha_{zz'} \geqslant 2 - \alpha''_p - \frac{2(\gamma - 1)}{d-1} = \frac{4d}{3(d-1)}.$$

Следовательно, при $d \geqslant 5$

$$\deg' z \geqslant \frac{4d}{3(d-1)} + 1 \geqslant \frac{7d-3}{3(d-1)} \geqslant \frac{2d}{d-1}.$$

Если $L \geqslant (2d+3)/3$, то

$$\alpha_p + \alpha'_p \geqslant \frac{4d}{d-1} - 2 \cdot \frac{\sum\limits_i x'_i + \sum\limits_j y'_j}{d-1} +$$

$$+\sum_{i\geqslant 2}\frac{x'_i(2-1/2^{i-2})}{d-1}+\sum_{j\geqslant 2}\frac{y'_j(2-1/2^{j-2})}{d-1}\geqslant$$
$$\geqslant\frac{4d-4(d-(L-1)/2-\gamma)}{d-1}=\frac{2L-2+4\gamma}{d-1}\geqslant\frac{4d+12\gamma}{3(d-1)},$$

поэтому

$$\alpha''_p\leqslant\frac{2d-12\gamma}{3(d-1)},\quad \alpha_{zz'}\geqslant\frac{4d+10\gamma}{3(d-1)},$$

так что $\deg' z\geqslant 2d/(d-1)$. В любом случае вершина z оказывается удачной.

(2н) Чтобы избежать анализа трудных случаев для нечётного L, рассмотрим в качестве графа G_2 нить T вместе с вершиной v_0 и смежными с ней вершинами $\overline{v}_0,\widetilde{v}_0\notin V(T)$. В этом случае $V_0=\{\overline{v}_0,\widetilde{v}_0,v'_0\}$, $m_2=L+2$. Если $C_2=1$, то вершины $\widetilde{v}_0$ и v'_0 смежны. В G_2 есть вершина, удалённая от V_0 на расстояние $(L+1)/2$, поэтому для G_1 равенства во второй строке (5) принимают вид

$$\sum_i x_i=\sum_j y_j=d-\frac{L+1}{2}.$$

Остальные неравенства, включая оценки (3) и (4), остаются теми же, что и для чётного L. Это означает, что нижняя оценка $\deg' z$ увеличивается на $\frac{1}{2(d-1)}$, а для $\deg' z+\deg'\overline{z}$, где вершины z и $\overline{z}$ слабые, — на $\frac{1}{d-1}$.

Такое изменение оценок исключает необходимость рассмотрения трудных случаев, так как величина недостатка в случае чётного L не превосходила $\frac{1}{d-1}$. Таким образом, все случаи МПВ для указанного графа G_2 стандартны, а все вершины из $G_1\setminus V_0$ удачны. В G_2 вершинам $\overline{v}_0$ и $\widetilde{v}_0$ можно положить условную степень 2, но так как избыток C не использован, припишем вклады по $\frac{2}{d-1}$ к $\deg'\overline{v}_0$ и $\deg'\widetilde{v}_0$. В результате получим удачные вершины $\overline{v}_0$ и $\widetilde{v}_0$. Лемма 6 доказана.

Заключение

В настоящей работе разработан МПВ, позволяющий найти минимальное число рёбер в графах из $\mathfrak{I}(n,d)$ для $d\geqslant 5$ и перечислить предельные графы, на которых это число рёбер достигается, что усиливает предыдущие результаты по этой теме. Поставленные в статье задачи решаются с помощью систем линейных уравнений и неравенств с несколькими параметрами, отражающими различные свойства рассматриваемых графов. Указанный метод может быть перспективным для решения других задач экстремальной теории графов.

Автор выражает глубокую благодарность проф. Д. Ю. Ханукаевой кафедры высшей математики РГУ нефти и газа (НИУ) им. И. М. Губкина, ценные замечания которой помогли улучшить текст статьи.

Финансирование работы

Исследование выполнено за счёт бюджета Российского гос. университета нефти и газа (НИУ) им. И. М. Губкина. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Жожикашвили В. А., Вишневский В. М.** Сети массового обслуживания. Теория и применение к сетям ЭВМ. М.: Радио и связь, 1988. 192 с.
2. **Зайченко Ю. П.** Задачи проектирования структуры распределённых вычислительных сетей // Автоматика. 1981. № 4. С. 27–40.
3. **Вишневский В. М., Савинецкий А. Б., Федотов Е. В.** Анализ и реализация одного метода повышения производительности сети пакетной коммутации // Автоматика и вычисл. техника. 1987. № 2. С. 24–30.
4. **Фараджев И. А.** Генерирование неизоморфных графов с заданным распределением степеней вершин // Алгоритмические исследования в комбинаторике. М.: Наука, 1978. С. 11–19.
5. **Murty U. S. R.** On some extremal graphs // Acta Math. Acad. Sci. Hung. 1968. V. 19. P. 69–74.
6. **Murty U. S. R.** Extremal nonseparable graphs of diameter 2 // Proof techniques in graph theory. Proc. 2nd Ann Arbor Graph Theory Conf. (Ann Arbor, USA, Feb., 1968). New York: Acad. Press, 1969. P. 111–117.
7. **Caccetta L.** Extremal graphs of diameter 3 // J. Austral. Math. Soc. 1979. V. A28, No. 1. P. 63–84.
8. **Caccetta L.** On extremal graphs with given diameter and connectivity // Ann. New York Acad. Sci. Topics Graph Theory. 1979. V. 328. P. 76–94.
9. **Caccetta L.** Extremal graphs of diameter 4 // J. Comb. Theory, Ser. B. 1976. V. 21. P. 104–115.
10. **Белоцерковский Д. Л.** Характеризация некоторых экстремальных графов с диаметром не превосходящим трёх // Дискрет. математика. 1997. Т. 9, № 1. С. 134–146.
11. **Белоцерковский Д. Л.** Об одной задаче перечисления образующих графов с ограничением на диаметр // Пробл. управления. 2010. № 1. С. 2–6.
12. **Belotserkovsky D. L.** The smallest number of edges in a 2-connected graph with specified diameter // Discrete Math. 2007. V. 307, No. 19–20. P. 2376–2384.
13. **Харари Ф.** Теория графов. М.: Мир, 1973. 300 с.
14. **Харари Ф., Палмер Э.** Перечисление графов. М.: Мир, 1977. 327 с.

15. Bollobas B. Strongly two-connected graphs // Proc. 7th Southeastern Conf. Combinatorics, Graph Theory, and Computing (Baton Rouge, USA, Feb. 9–12, 1976). Winnipeg, Man.: Utilitas Math., 1976. P. 161–170. (Congr. Numer.; V. 17).
16. Bollobas B. Extremal graph theory. London: Acad. Press, 1978. 488 p.
17. Bollobas B. Extremal graph theory. Mineola: Dover Publ., 2004. 512 p.
18. Bollobas B. Modern graph theory. New York: Springer, 1998. 394 p. (Graduate Texts Math.; V. 184).
19. Bollobas B. Modern graph theory. Heidelberg: Springer, 2013. 408 p.
20. Lovasz L. Large networks and graph limits. Hoboken, NJ: AMS, 2012. 475 p.
21. Jarry A., Laugier A. On the minimum number of edges of two-connected graphs with given diameter // Discrete Math. 2012. V. 312. P. 757–762.

Белоцерковский Дмитрий Леонидович

Статья поступила
18 октября 2024 г.
После доработки —
2 июня 2025 г.
Принята к публикации
22 июня 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 69–101

UDC 519.176 DOI: 10.33048/daio.2025.32.815

ON EXTREME BICONNECTED GRAPHS WITH SPECIFIED DIAMETER

D. L. Belotserkovsky

Gubkin Russian State University of Oil and Gas,
65 Leninskiy Avenue, 119991 Moscow, Russia
E-mail: belozer68@mail.ru

Abstract. The article considers two problems in the extremal graph theory. The first problem is formulated and completely proved as a theorem on the exact lower bound for the number of edges in a biconnected graph of diameter not exceeding a given value. A discharging method is developed and used to solve the second problem: Enumerating the graphs for which the lower bound is attained. Since the problem of enumerating graphs is very difficult and has not been solved in general form before, all the so-called limit graphs are enumerated on the basis of the discharging method. The problem of finding all limit graphs is formulated and proved as the second theorem in the article. Further, we explain why limit graphs are the most promising for the enumeration problem, and give some techniques for constructing extremal graphs using the limit graphs already obtained. Illustr. 11, bibliogr. 21.

Keywords: graph, diameter, exact lower bound, charging method, discharging method (DM).

References

1. **V. A. Zhozhikashvili** and **V. M. Vishnevskiy,** *Queueing Networks: Theory and Application to Computer Networks* (Radio Svyaz, Moscow, 1988) [Russian].
2. **Yu. P. Zaychenko,** Problems of design for distributed computing networks, *Avtomatika*, No. 4, 27–40 (1981) [Russian].
3. **V. M. Vishnevskiy, A. B. Savinetskiy,** and **E. V. Fedotov,** Analysis and implementation of one method to improve the performance of a packet switching network, *Avtom. Vychisl. Tekh.*, No. 2, 24–30 (1987) [Russian].
4. **I. A. Faradzhev,** Generating non-isomorphic graphs with a given distribution of vertex degrees, in *Algorithmic Research in Combinatorics* (Nauka, Moscow, 1978), pp. 11–19 [Russian].

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

5. **U. S. R. Murty,** On some extremal graphs, *Acta Math. Acad. Sci. Hung.* **19**, 69–74 (1968).
6. **U. S. R. Murty,** Extremal nonseparable graphs of diameter 2, in *Proof Techniques in Graph Theory*, Proc. 2nd Ann Arbor Graph Theory Conf. (Ann Arbor, USA, Feb., 1968) (Acad. Press, New York, 1969), pp. 111–117.
7. **L. Caccetta,** Extremal graphs of diameter 3, *J. Austral. Math. Soc.* **A28** (1), 63–84 (1979).
8. **L. Caccetta,** On extremal graphs with given diameter and connectivity, *Ann. New York Acad. Sci. Topics Graph Theory* **328**, 76–94 (1979).
9. **L. Caccetta,** Extremal graphs of diameter 4, *J. Comb. Theory, Ser. B*, **21**, 104–115 (1976).
10. **D. L. Belotserkovsky,** Characterization of some extremal graphs with diameter not exceeding three, *Diskretn. Mat.* **9** (1), 134–146 (1997) [Russian] [*Discrete Math. Appl.* **7** (2), 163–176 (1997)].
11. **D. L. Belotserkovsky,** On a problem of enumeration of generating graphs with a diameter constraint, *Probl. Upr.*, No. 1, 2–6 (2010) [Russian].
12. **D. L. Belotserkovsky,** The smallest number of edges in a 2-connected graph with specified diameter, *Discrete Math.* **307** (19–20), 2376–2384 (2007).
13. **F. Harary,** *Graph Theory* (Addison-Wesley, London, 1969; Mir, Moscow, 1973 [Russian]).
14. **F. Harary** and **E. M. Palmer,** *Graphical Enumeration* (Acad. Press, New York, 1973; Mir, Moscow, 1977 [Russian]).
15. **B. Bollobas,** Strongly two-connected graphs, in *Proc. 7th Southeastern Conf. Combinatorics, Graph Theory, and Computing* (Baton Rouge, USA, Feb. 9–12, 1976) (Utilitas Math., Winnipeg, Man., 1976), pp. 161–170 (Congr. Numer., Vol. 17).
16. **B. Bollobas,** *Extremal Graph Theory* (Acad. Press, London, 1978).
17. **B. Bollobas,** *Extremal Graph Theory* (Dover Publ., Mineola, 2004).
18. **B. Bollobas,** *Modern Graph Theory* (Springer, New York, 1998) (Graduate Texts Math., Vol. 184).
19. **B. Bollobas,** *Modern Graph Theory* (Springer, Heidelberg, 2013).
20. **L. Lovasz,** *Large Networks and Graph Limits* (AMS, Hoboken, NJ, 2012).
21. **A. Jarry** and **A. Laugier,** On the minimum number of edges of two-connected graphs with given diameter, *Discrete Math.* **312**, 757–762 (2012).

Dmitry L. Belotserkovsky

Received October 18, 2024
Revised June 2, 2025
Accepted June 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 102–117

УДК 333.1:519.86 DOI: 10.33048/daio.2025.32.835

НЕЧЁТКОЕ ЯДРО И ВАЛЬРАСОВСКИЕ РАСПРЕДЕЛЕНИЯ ОДНОЙ МОДЕЛИ ПРОСТРАНСТВЕННОЙ ЭКОНОМИКИ

В. А. Васильев

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

E-mail: vasilev@math.nsc.ru

Аннотация. Анализируется эквивалентность неблокируемых и вальрасовских планов в пространственных моделях регионального взаимодействия, разработанных акад. А. Г. Гранбергом и его школой. Изучается непрерывный вариант гипотезы Эджворта: совпадение нечётких ядер с множествами равновесных распределений. Следует отметить, что помимо самостоятельной ценности, условия совпадения указанных множеств представляют значительный интерес при рассмотрении вопросов существования вальрасовских планов в пространственных моделях с неограниченными технологическими множествами. Ключевую роль в получении теоремы эквивалентности играют условия строгой автаркичности регионов (некоторый аналог известного условия Слейтера) и их неограниченности по функционалу. Важным общесистемным требованием является отсутствие так называемого «рога изобилия». Библиогр. 10.

Ключевые слова: пространственная экономика, вальрасовское распределение, нечёткое ядро, автаркия, «рог изобилия».

Введение

В заметке продолжается сравнительный анализ равновесных и устойчивых по отношению к блокированию нечёткими коалициями планов в пространственных моделях регионального взаимодействия, разработанных акад. А. Г. Гранбергом и его школой [1]. Изучается непрерывный вариант гипотезы Эджворта, касающийся совпадения нечётких ядер с множествами равновесных распределений в условиях совершенной конкуренции. Ключевую роль в обеспечении такого совпадения играют условия

строгой автаркичности регионов и их неограниченности по функционалу. В содержательном плане строгая автаркичность означает потенциальную возможность реализации региональных планов строгого превышения экспорта над импортом. Иначе говоря, автаркичность есть некоторая форма гипотетической автономности на региональном уровне (не путать с сознательной политикой, направленной на создание замкнутого хозяйства, обособленного от экономики других регионов). Что касается региональной неограниченности по функционалу, это условие является аналогом монотонности индивидуальных предпочтений и потенциальной неограниченности потребления в классических моделях чистого обмена (см., например, [2]). Отметим сразу, что для получения основного результата об эквивалентности равновесных и неблокируемых распределений изучаемой модели достаточно и более скромного требования, чем региональная неограниченность по функционалу. Преимущество последнего заключается в значительном упрощении проверки в сравнении с болеее слабым предположением о ненасыщаемости регионов (см. [3]). Кроме перечисленных важным общесистемным требованием в обосновании основного результата является отсутствие так называемого «рога изобилия». Как и в классических моделях равновесного анализа, указанное требование означает, что при нулевом экономическом потенциале рассматриваемой пространственной системы возможна лишь её нулевая хозяйственная активность.

Отметим, что исследование проводится по схеме, разработанной автором в одной из ранних работ [4] и применявшейся в ряде дальнейших публикаций (см., например, [3]). Помимо основного материала в настоящей заметке уточняются некоторые предшествующие результаты и восстанавливаются полные доказательств там, где ранее приводились лишь их краткие схемы, данные только в нежурнальных публикациях. Завершая введение, отметим, что помимо самостоятельной ценности, признаки совпадения нечётких ядер и множеств равновесных распределений представляют значительный интерес при рассмотрении вопросов существования вальрасовских планов в пространственных моделях с неограниченными технологическими множествами (см. [3]).

Кроме введения работа содержит ещё три раздела. В разд. 1 приводится описание изучаемой модели пространственной экономики. В разд. 2 вводятся ключевые понятия: определение нечёткого ядра и вальрасовского равновесия рассматриваемой модели регионального взаимодействия. Разд. 3 содержит доказательство основного результата работы, указывающего условия, достаточные для совпадения нечёткого ядра и множества вальрасовских распределений исследуемой модели пространственной экономики.

1. Описание модели M

Следуя [5, 6], дадим краткое описание рассматриваемой модели пространственной экономики M, предназначенной для анализа экономического взаимодействия r регионов, обменивающихся n транспортабельными видами продукции из единого для всех регионов списка $\{1,\dots,n\}$. Эта модель имеет вид

$$M = \langle R, \{A^s, G^s, H^s, b^s, d^s\}_{s\in R}\rangle,$$

где $R = \{1,\dots,r\}$ — множество (номеров) регионов; A^s — прямоугольная матрица размера $n_s \times l_s$, характеризующая производственный сектор региона $s \in R$; G^s и H^s — прямоугольные матрицы размера $n_s \times n$, описывающие способы вывоза и ввоза в регионе $s \in R$; b^s — вектор-столбец размерности n_s, характеризующий имеющийся ресурсно-технологический потенциал региона $s \in R$; d^s — вектор-столбец размерности n_s, описывающий затраты ресурсов и продукции, связанные с достижением целей развития региона $s \in R$.

Детальную интерпретацию и ряд важных приложений этой модели можно найти в [1, 5, 6]. Здесь отметим лишь, что в отличие от более общей постановки из [1], далее предполагается, как и в [6], что изучаемый межрегиональный обмен реализуется на единой, общей для всех рыночной площадке, поэтому в приводимом ниже описании ресурсно-технологических возможностей Z_s регионов $s \in R$ их внешние связи представлены лишь переменными вывоза и ввоза, без разделения по регионам-контрагентам. Напомним, что возможности региона s модели M по ввозу (вывозу) k-го продукта характеризуются k-м столбцом матрицы H^s (G^s), имеющим лишь два ненулевых элемента: 1 на k-й позиции и $-c^v_s$ на позиции, отвечающей внешнеторговым перевозкам, при этом c^v_s — транспортные издержки по ввозу единицы продукта k в регион s (k-й столбец матрицы G^s также имеет два ненулевых элемента -1 и $-c^u_s$ на тех же позициях, при этом c^u_s — транспортные издержки по вывозу единицы продукта k из региона s). В целом, каждая из матриц G^s и H^s — за исключением строк транспортных издержек — получена из единичной матрицы вычеркиванием столбцов, отвечающих отраслям с нетранспортабельной продукцией (с последующим умножением на -1 для G^s). Подробности, касающиеся других параметров модели M, см. в [1, 6].

Напомним [6], что множество Z_s ресурсно-технологических возможностей региона $s \in R$ имеет вид

$$Z_s = \big\{(x^s, u^s, v^s, \lambda_s) \in \mathbb{R}^{l_s}_+ \times \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{R}_+ \mid A^s x^s + G^s u^s + H^s v^s \geqslant b^s + \lambda_s d^s\big\}.$$

Неотрицательные вектор-столбцы $x^s = \big(x^s_i\big)^{l_s}_{i=1}$, $u^s = \big(u^s_j\big)^n_{j=1}$, $v^s = \big(v^s_j\big)^n_{j=1}$ для региона $s \in R$ модели M определяют объёмы производства, вывоза

и ввоза соответственно, а число $\lambda_s \in \mathbb{R}_+$ — степень достижения целей регионального развития. Как обычно, символом $\mathbb{R}$ обозначается множество вещественных чисел, а неравенство для векторов понимается в обычном покомпонентном смысле: $x \geqslant y$, если $x_i \geqslant y_i$ при $i = 1, \ldots, m$ для любых $x = (x_1, \ldots, x_m)$, $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$).

Качество ресурсно-технологических возможностей (планов) $z^s \in Z_s$ оценивается с помощью функций t_s, сопоставляющих каждому вектору $z^s = (x^s, u^s, v^s, \lambda_s)$ его последнюю компоненту λ_s:

$$t_s(z^s) = t_s(x^s, u^s, v^s, \lambda_s) = \lambda_s, \quad (x^s, u^s, v^s, \lambda_s) \in Z_s, \quad s \in R.$$

Другими словами, отображения $t_s \colon Z_s \to \mathbb{R}$ — целевые функции участников $s \in R$, характеризующие степень достижения целей их регионального развития.

Положим $Z = Z_M = \prod\limits_{s \in R} Z_s$ и через $Z(R) = Z_M(R)$ обозначим совокупность *сбалансированных планов* модели M:

$$Z_M(R) = \Big\{(x^s, u^s, v^s, \lambda_s)_{s \in R} \in Z_M \mid \sum_{s \in R} u^s \geqslant \sum_{s \in R} v^s\Big\}.$$

Если модель M однозначно определяется из контекста, то символ M опускается и используются сокращения Z, $Z(R)$ и т. п. Согласно интерпретации из [1, 6] условие сбалансированности $\sum\limits_{s \in R} u^s \geqslant \sum\limits_{s \in R} v^s$ означает, что по каждому из n транспортабельных продуктов k, участвующих в обмене, должно выполняться стандартное условие баланса: суммарный импорт $\sum\limits_{s \in R} v^s_k$ не превышает суммарного экспорта $\sum\limits_{s \in R} u^s_k$. Напомним, что в модели M предполагается, что списки $N_s = \{1, \ldots, n\}$, $s \in R$, транспортабельных продуктов регионов совпадают между собой и содержат ровно n элементов.

Аналогично множеству R всех регионов рассматриваются и сбалансированные планы непустых частей R, называемых *коалициями*. А именно, для каждой коалиции $T \subseteq R$ вводим обозначения $Z_T = Z_{M,T} = \prod\limits_{s \in T} Z_s$ и под $Z(T) = Z_M(T)$ понимаем совокупность сбалансированных планов этой коалиции

$$Z_M(T) = \Big\{(x^s, u^s, v^s, \lambda_s)_{s \in T} \in Z_{M,T} \mid \sum_{s \in T} u^s \geqslant \sum_{s \in T} v^s\Big\}.$$

Особое место в формулировке ряда утверждений, касающихся коалиционной стабильности равновесных планов, занимают одноэлементные коалиции. Отвечающие им множества сбалансированных планов имеют вид

$$Z_M(s) = \{(x^s, u^s, v^s, \lambda_s) \in Z_s \mid u^s \geqslant v^s\}, \quad s \in R.$$

Здесь и далее используются стандартные сокращения, при которых фигурные скобки в записи одноэлементных множеств опускаются: $Z_M(s) = Z_M(\{s\})$, $s \in R$.

Определение 1. Регион $s \in R$ называется *автаркическим*, если его технологические возможности могут обеспечить превышение экспорта над импортом

$$Z_M(s) = \{(x^s, u^s, v^s, \lambda_s) \in Z_s \mid u^s \geqslant v^s\} \neq \varnothing. \tag{M1}$$

Элементы множества $Z_M(s)$ называются *автаркическими планами региона* s.

Как видно из определения 1, выполнение условия (M1) означает лишь потенциальную реализуемость автономных — сбалансированных по экспорту и импорту — планов развития региона $s \in R$. Подчеркнём, что как в содержательном, так и в чисто формальном плане словосочетание «автаркический регион» есть термин, характеризующий вышеупомянутое свойство региональной экономики. В отличие от существительного «автаркия» этот термин не имеет никакого отношения к реальной экономической политике региона.

Важную роль в дальнейших рассмотрениях играет и такой объект, связанный с моделью M, как множество $Z_{M_0}(R)$ сбалансированных планов её однородной составляющей

$$M_0 = \langle R, \{A^s, G^s, H^s, 0, d^s\}_{s \in R}\rangle.$$

Согласно определению модель M_0 отличается от M только тем, что начальный ресурсно-технологический потенциал M_0 равен нулю: $b^s = 0$ для каждого региона $s \in R$.

Определение 2. Будем говорить, что в модели M *отсутствует «рог изобилия»*, если выполняется условие

$$Z_{M_0}(R) = \{0\}. \tag{M2}$$

Напомним, что условие (M2) трактуется как невозможность ненулевого выпуска при нулевом ресурсно-технологическом потенциале пространственной экономической системы M.

2. Вальрасовское равновесие и нечёткое ядро

Переходя к описанию вальрасовского равновесия в модели M, определим сначала понятие *бюджетных множеств* $B_s(p)$ регионов $s \in R$ при ценах $p = (p_1, \ldots, p_n) \in \mathbb{R}^n$ (см. также [6]). Последние задаются формулой, в которой $x \cdot y$ — скалярное произведение векторов x и y:

$$B_s(p) = \{z^s = (x^s, u^s, v^s, \lambda_s) \in Z_s \mid p \cdot u^s \geqslant p \cdot v^s\}, \quad s \in R.$$

Определение 3 (вальрасовское равновесие [6])**.** Сбалансированный план $\overline{z} = (\overline{x}^s, \overline{u}^s, \overline{v}^s, \overline{\lambda}_s)_{s\in R} \in Z_M(R)$ и ненулевой вектор цен $\overline{p} \in \mathbb{R}^n_+$ образуют вальрасовское равновесие модели M, если для каждого региона $s \in R$ выполняются условия

(W1) $\overline{p} \cdot \overline{u}^s \geqslant \overline{p} \cdot \overline{v}^s$;

(W2) $\overline{\lambda}_s \geqslant \lambda_s$ для всех планов $(x^s, u^s, v^s, \lambda_s) \in B_s(\overline{p})$.

При этом план $\overline{z}$ называется *вальрасовским* планом модели M, а цены $\overline{p}$ — *равновесными ценами* M. Совокупность вальрасовских планов модели M обозначим через $W(M)$.

Замечание 1. Вальрасовские планы называют также равновесными планами (или вальрасовскими распределениями), а равновесные цены — вальрасовскими (см., например, [1, 6]).

Введём ещё одно рассматриваемое в работе понятие оптимальности — определение нечёткого ядра применительно к рассматриваемой модели пространственной экономики. Рассмотрим гиперкуб

$$I^R = \{\tau = (\tau_1, \ldots, \tau_r) \in \mathbb{R}^R_+ \mid \tau_s \leqslant 1,\ s \in R\}$$

и, следуя [3], положим $\sigma_F = I^R \backslash \{0\}$. Согласно [2] (см. также [7]) элементы множества σ_F называются *нечёткими коалициями*. Каждая компонента τ_s вектора $\tau = (\tau_1, \ldots, \tau_r) \in \sigma_F$ трактуется как степень участия игрока s в большой коалиции R. Для нечёткой коалиции $\tau = (\tau_1, \ldots, \tau_r) \in \sigma_F$ через $R(\tau)$ обозначается *носитель* τ

$$R(\tau) = \{s \in R \mid \tau_s > 0\}.$$

Определение 4 (нечёткое ядро [6])**.** Коалиция $\tau = (\tau_1, \ldots, \tau_r) \in \sigma_F$ блокирует сбалансированный план $\overline{z} = (\overline{z}^s)_{s\in R} \in Z(R)$, если существуют региональные планы

$$z^s = (x^s, u^s, v^s, \lambda_s) \in Z_s, \quad s \in R(\tau),$$

для которых выполняются условия

(CF1) $t_s(z^s) > t_s(\overline{z}^s)$ для каждого $s \in R(\tau)$;

(CF2) $\sum\limits_{s\in R(\tau)} \tau_s u^s \geqslant \sum\limits_{s\in R(\tau)} \tau_s v^s$.

План $z \in Z(R)$, не блокируемый никакой коалицией $\tau \in \sigma_F$, называется *неблокируемым* сбалансированным планом. Обозначим совокупность всех неблокируемых сбалансированных планов модели M через $C_F(M)$ и назовём $C_F(M)$ её *нечётким ядром*.

В завершение раздела отметим, что естественным необходимым условием наличия как равновесных, так и неблокируемых сбалансированных планов является непустота множества $Z_M(R)$. Ясно, что автаркичность всех регионов гарантирует выполнение этого условия. Действительно,

предполагая существование автаркических планов $z_0^s = \left(x_0^s, u_0^s, v_0^s, \lambda_s^0\right)$, непосредственно из определения множества сбалансированных планов модели M получаем: план $\left(z_0^s\right)_{s\in R}$ принадлежит $Z_M(R)$ в силу очевидного неравенства $\sum\limits_{s\in R} u_0^s \geqslant \sum\limits_{s\in R} v_0^s$. Отметим, что автаркичность региона $s \in R$ обеспечивает, например, неотрицательная разрешимость системы $A^s x^s \geqslant b^s$ (или системы $A^s x^s \geqslant b^s + \lambda_s d^s$). В этих случаях автаркические планы имеют вид $\left(x_0^s, 0, 0, 0\right)$ или $\left(x_1^s, 0, 0, \lambda_s^1\right)$ соответственно. Здесь x_0^s — решение системы $A^s x^s \geqslant b^s$, $x^s \geqslant 0$, а $\left(x_1^s, \lambda_s^1\right)$ — решение системы $A^s x^s \geqslant b^s + \lambda_s d^s$, $x^s \geqslant 0$, $\lambda_s \geqslant 0$.

Представляют интерес и другие условия, но их формулировка упирается в отсутствие экономически осмысленной конкретики, касающейся матриц A_s, G_s, H_s и векторов b^s и d^s. Конечно, если для достижения целей развития региона $s \in R$ требуются ненулевые затраты всех ресурсов и продукции (т. е. имеет место строгая отрицательность всех компонент вектора d^s), то тривиальным автаркическим планом этого региона является любой элемент $z_0^s = \left(x_0^s, u_0^s, v_0^s, \lambda_s^0\right) \in Z_s$, удовлетворяющий условиям: $x_0^s = 0$, $u_0^s = 0$, $v_0^s = 0$ и $b_k^s + \lambda_s^0 d_k^s \leqslant 0$ для каждого $k = 1, \ldots, n$. Поскольку необходимость использования всех ресурсов и продукции для реализации целевых установок каждого региона представляется достаточно жёстким условием, вопрос отыскания более слабых и экономически содержательных требований, обеспечивающих существование сбалансированных планов, остаётся открытым.

3. Условия совпадения множеств $C_F(M)$ и $W(M)$

Для формулировки теоремы о совпадении нечёткого ядра и множества вальрасовских распределений модели M потребуется введение понятия строгой автаркичности региона $s \in R$ — техническое усиление введённого ранее условия автаркичности (M1) из разд. 1.

Определение 5. Регион $s \in R$ называется *строго автаркическим*, если его технологические возможности могут обеспечить строгое превышение экспорта над импортом:

$$\widehat{Z}_M(s) = \{(x^s, u^s, v^s, \lambda_s) \in Z_s \mid u^s \gg v^s\} \neq \varnothing, \qquad \text{(M1*)}$$

где $x \gg y$ означает выполнение строгих неравенств $x_i > y_i$, $i = 1, \ldots, m$, для $x, y \in \mathbb{R}^m$.

Кроме того, среди используемых далее характеристик, касающихся регионов как таковых, помимо (M1*) укажем ещё одну.

Определение 6. Регион $s \in R$ называется *неограниченным по функционалу*, если его целевая функция t_s не ограничена сверху на технологическом множестве Z_s

$$\sup_{z^s \in Z_s} t_s(z^s) = \infty. \tag{M3}$$

Условие (M3) является некоторым аналогом монотонности индивидуальных предпочтений и потенциальной неограниченности потребления в классических моделях чистого обмена (см., например, [2]).

Для технологических планов z^s регионов $s \in R$ введём обозначения

$$P_s(z^s) = \{\widetilde{z}^s \in Z_s \mid t_s(\widetilde{z}^s) > t_s(z^s)\}, \quad z^s \in Z_s,\ s \in R.$$

Справедливо простое, но полезное в дальнейших рассмотрениях утверждение.

Лемма 1. *Если в модели M отсутствует «рог изобилия», т. е. имеет место* (M2)*, и все её регионы неограниченны по функционалу, то для любого плана $z = (z^s)_{s \in R} \in Z_M(R)$ множество $P_s(z^s)$, $s \in R$, непустое.*

Доказательство. Ясно, что множество $Z_M(R)$ замкнуто. Кроме того, при отсутствии «рога изобилия» из многогранности множества $Z_M(R)$ на основании критерия Голдмана [8, следствие 1B] вытекает его ограниченность. Значит, множество $Z_M(R)$ сбалансированных планов рассматриваемой модели является компактом, поэтому проекции $\mathrm{Pr}_{Z_s} Z_M(R)$, $s \in R$, будучи непрерывными образами компакта $Z_M(R)$, также компактны. Итак, полагая $\widetilde{Z}_s = \mathrm{Pr}_{Z_s} Z_M(R)$, получаем, что максимумы непрерывных функций t_s на соответствующих множествах $\widetilde{Z}_s$ конечны. Отсюда в силу бесконечности супремумов функций $t_s(z^s)$ на их областях определения выполняются очевидные неравенства $\sup\limits_{z^s \in Z_s} t_s(z^s) > \max\limits_{z^s \in \widetilde{Z}_s} t_s(z^s)$, но эти неравенства и означают требуемое: для каждого $s \in R$ существует региональный план $\widetilde{z}_s \in Z_s$ такой, что $t_s(\widetilde{z}^s) > t_s(z^s)$. Итак, для каждого $s \in R$ множество $P_s(z^s)$ непустое. Лемма 1 доказана.

Пусть сбалансированный план $z = (z^s)_{s \in R}$ модели M удовлетворяет условию $P_s(z^s) \neq \varnothing$ для каждого региона $s \in R$. В таком случае в дальнейшем будем использовать обозначения

$$A_s(z^s) = \{T_s(\widetilde{z}^s) \mid \widetilde{z}^s \in P_s(z^s)\}, \quad s \in R,$$

где T_s — линейные операторы, определяемые соотношениями

$$T_s(\widetilde{z}^s) = \{\widetilde{u}^s - \widetilde{v}^s \mid \widetilde{z}^s = (\widetilde{x}^s, \widetilde{u}^s, \widetilde{v}^s, \widetilde{\lambda}_s) \in \mathbb{R}^{l_s} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}\}, \quad s \in R.$$

Важную роль в дальнейших рассмотрениях играет следующая техническая

Лемма 2. *Если в модели M отсутствует «рог изобилия» и все её регионы неограниченны по функционалу, то для любого плана $z = (z^s)_{s\in R}$ из $Z_M(R)$ множество*

$$A(z) = \Big\{ \sum_{s\in R} \tau_s w^s \mid (\tau_1, \dots, \tau_r) \in \sigma_F,\ (w^1, \dots, w^r) \in \prod_{s\in R} A_s(z^s) \Big\}.$$

выпукло [1)].

Доказательство. Для произвольного плана $z = (z^s)_{s\in R} \in Z_M(R)$ установим выпуклость множества $A(z)$. Начнём с того, что в силу леммы 1 оно определено корректно, поскольку $P_s(z^s) \neq \varnothing$, $s \in R$. Далее, рассмотрим произвольные элементы a^1 и a^2 из $A(z)$ и какое-либо число $\nu \in (0,1)$. Покажем, что вектор $a(\nu) = \nu a^1 + (1-\nu)a^2$ принадлежит множеству $A(z)$. Напомним, что по определению $A(z)$ существуют наборы

$$z^s_{(1)} = \big(x^s_{(1)}, u^s_{(1)}, v^s_{(1)}, \lambda^{(1)}_s\big) \in P_s(z^s), \quad s \in R,$$
$$z^s_{(2)} = \big(x^s_{(2)}, u^s_{(2)}, v^s_{(2)}, \lambda^{(2)}_s\big) \in P_s(z^s), \quad s \in R,$$

и нечёткие коалиции $\tau^{(1)} = \big(\tau^{(1)}_1, \dots, \tau^{(1)}_r\big)$, $\tau^{(2)} = \big(\tau^{(2)}_1, \dots, \tau^{(2)}_r\big)$ такие, что

$$a^1 = \sum_{s\in R} \tau^{(1)}_s \big(u^s_{(1)} - v^s_{(1)}\big), \quad a^2 = \sum_{s\in R} \tau^{(2)}_s \big(u^s_{(2)} - v^s_{(2)}\big).$$

Следовательно, полагая $w^s_{(1)} = u^s_{(1)} - v^s_{(1)}$, $w^s_{(2)} = u^s_{(2)} - v^s_{(2)}$, имеем

$$a(\nu) = \nu \sum_{s\in R} \tau^{(1)}_s w^s_{(1)} + (1-\nu) \sum_{s\in R} \tau^{(2)}_s w^s_{(2)}. \tag{1}$$

Далее, введём в рассмотрение нечёткую коалицию $\tau(\nu)$, определяемую формулой $\tau(\nu) = \nu\tau^{(1)} + (1-\nu)\tau^{(2)}$. Ясно, что компоненты $\tau(\nu)_s$ вектора $\tau(\nu)$ имеют вид

$$\tau(\nu)_s = \nu\tau^{(1)}_s + (1-\nu)\tau^{(2)}_s, \quad s \in R. \tag{2}$$

Определим наборы $z^s(\nu)$ соотношениями

$$z^s(\nu) = \begin{cases} \Big(\frac{\nu\tau^{(1)}_s}{\tau(\nu)_s}\Big) z^s_{(1)} + \Big(\frac{(1-\nu)\tau^{(2)}_s}{\tau(\nu)_s}\Big) z^s_{(2)}, & \text{если } s \in N(\tau^{(1)}) \cup N(\tau^{(2)}), \\ z^s_{(1)}, & \text{если } s \in \widetilde{R}, \end{cases} \tag{3}$$

где $\widetilde{R} = R \setminus (N(\tau^{(1)}) \cup N(\tau^{(2)}))$.

Понятно, что из формулы (1) и из включений $z^s_{(1)}, z^s_{(2)} \in P_s(z^s)$ в силу линейности функций t_s вытекают неравенства $t_s(z^s(\nu)) > t_s(z^s)$, $s \in R$.

[1)] Напомним, что согласно определению модели M векторы u^s, v^s, $s \in R$, принадлежат одному и тому же пространству $\mathbb{R}^n$, поэтому фигурирующие здесь векторы $w^s = \widetilde{u}^s - \widetilde{v}^s$, $s \in R$, также из пространства $\mathbb{R}^n$.

Отсюда с учётом выпуклости множеств Z_s для всех векторов $z^s(\nu)$, определяемых формулами (2) и (3), получаем включения

$$z^s(\nu) \in P_s(z^s), \quad s \in R. \tag{4}$$

Используя формулы (2), (3), нетрудно убедиться, что для каждого региона s из $N(\tau^{(1)}) \cup N(\tau^{(2)})$ выполняются соотношения

$$u^s(\nu) = \left(\frac{\nu\tau_s^{(1)}}{\tau(\nu)_s}\right)u^s_{(1)} + \left(\frac{(1-\nu)\tau_s^{(2)}}{\tau(\nu)_s}\right)u^s_{(2)},$$
$$v^s(\nu) = \left(\frac{\nu\tau_s^{(1)}}{\tau(\nu)_s}\right)v^s_{(1)} + \left(\frac{(1-\nu)\tau_s^{(2)}}{\tau(\nu)_s}\right)v^s_{(2)}.$$

Отсюда, учитывая, что при $s \in \widetilde{R}$ все компоненты $\tau(\nu)_s$ нулевые, получаем равенство

$$\sum_{s\in R}\tau(\nu)_s(u^s(\nu)-v^s(\nu)) = \nu\sum_{s\in N(\tau^{(1)})}\tau_s^{(1)}w^s_{(1)} + (1-\nu)\sum_{s\in N(\tau^{(2)})}\tau_s^{(2)}w^s_{(2)}, \tag{5}$$

где, как и ранее, $w^s_{(1)} = u^s_{(1)} - v^s_{(1)}$, $w^s_{(2)} = u^s_{(2)} - v^s_{(2)}$. Однако, в силу (1) правая часть равенства (5) и есть в точности вектор $a(\nu)$. Значит, ввиду включения $\sum\limits_{s\in R}\tau(\nu)_s(u^s(\nu)-v^s(\nu)) \in A(z)$, вытекающего из соотношений (3) и (4), получаем требуемое: вектор $a(\nu) = \nu a^1 + (1-\nu)a^2$ принадлежит множеству $A(z)$. Лемма 2 доказана.

Переходя к анализу условий эквивалентности вальрасовских распределений и распределений, не блокируемых нечёткими коалициями, напомним, что первые всегда принадлежат нечёткому ядру $C_F(M)$. А именно, справедливо следующее утверждение из [3] (ниже, для автономности изложения, приводится и его краткое доказательство).

Утверждение 1. *Для любой модели пространственной экономики M справедливо вложение*

$$W(M) \subseteq C_F(M). \tag{6}$$

Доказательство. Докажем (6), приводя к противоречию противоположное предположение. Пусть $\overline{z} = (\overline{z}^s)_{s\in R}$ — произвольное распределение из $W(M)$, а $\overline{p}$ — отвечающие ему равновесные цены. Допуская, что $\overline{z}$ блокируется какой-либо нечёткой коалицией $\tau = (\tau_1, \ldots, \tau_r)$, непосредственно из определения блокирования выводим, что существуют региональные планы $z^s = (x^s, u^s, v^s, \lambda_s)$, $s \in R(\tau)$, такие, что

$$\sum_{s\in R(\tau)}\tau_s u^s \geqslant \sum_{s\in R(\tau)}\tau_s v^s, \tag{7}$$

и, кроме того,

$$t_s(z^s) > t_s(\overline{z}^s), \quad s \in R(\tau). \tag{8}$$

В силу того, что распределение $\overline{z}$ вальрасовское, из (8) вытекают неравенства $\overline{p} \cdot u^s < \overline{p} \cdot v^s$, $s \in R(\tau)$. Умножая эти соотношения на соответствующие компоненты вектора τ и складывая получающиеся неравенства, имеем

$$\overline{p} \cdot \sum_{s \in R(\tau)} \tau_s u^s < \overline{p} \cdot \sum_{s \in R(\tau)} \tau_s v^s.$$

Однако последнее неравенство противоречит неотрицательности вектора цен $\overline{p}$ и неравенству (7), полученному из предположения о том, что нечёткая коалиция τ блокирует распределение $\overline{z}$ с помощью региональных планов z^s, $s \in R(\tau)$. Утверждение 1 доказано.

Воспользуемся приведёнными леммами 1, 2 и утверждением 1 для доказательства главного результата работы — приводимой ниже теоремы эквивалентности равновесных распределений и распределений, не блокируемых нечёткими коалициями, в рассматриваемой модели пространственной экономики.

Теорема 1. *Если в модели M отсутствует «рог изобилия» и каждый её регион строго автаркический и неограниченный по функционалу, то множество вальрасовских планов этой модели совпадает с её нечётким ядром:*

$$W(M) = C_F(M).$$

Доказательство. В силу утверждения 1 для доказательства теоремы достаточно установить справедливость вложения $C_F(M) \subseteq W(M)$. Поскольку в случае пустого нечёткого ядра такое вложение выполняется очевидным образом, в дальнейшем считаем, что $C_F(M) \neq \varnothing$. Итак, пусть $\overline{z} = (\overline{z}^s)_{s \in R}$ — произвольное распределение из $C_F(M)$. Нетрудно видеть, что непосредственно из определения блокирования вытекает соотношение

$$A(\overline{z}) \cap \mathbb{R}^n_+ = \varnothing. \tag{9}$$

Далее, в силу леммы 2 в условиях нашей теоремы множество $A(\overline{z})$ выпуклое. Следовательно, в силу очевидной выпуклости положительного ортанта на основании соотношения (9) выпуклые множества $A(\overline{z})$ и $\mathbb{R}^n_+$ удовлетворяют условию известной теоремы отделимости Минковского (см., например, [2, 9, 10]), согласно которой для непустых непересекающихся множеств $A(\overline{z})$ и $\mathbb{R}^n_+$ существует ненулевой вектор $\overline{p} \in \mathbb{R}^n$ такой, что выполняется неравенство

$$\sup_{a \in A(\overline{z})} \overline{p} \cdot a \leqslant \inf_{b \in \mathbb{R}^n_+} \overline{p} \cdot b. \tag{10}$$

Покажем, что пара $(\overline{p}, \overline{z})$ образует вальрасовское равновесие модели M. Начнём с проверки неотрицательности вектора $\overline{p}$. Легко видеть, что неотрицательность этого вектора есть прямое следствие неравенства (10). Действительно, допуская, что хотя бы одна из компонент $\overline{p}$ меньше нуля, получаем, что правая часть упомянутого неравенства была бы равна $-\infty$, что противоречило бы конечности его левой части. Значит, вектор $\overline{p}$ принадлежит конусу $\mathbb{R}^n_+$. Поскольку $\overline{p} \geqslant 0$, оба сомножителя в каждом из скалярных произведений в правой части неравенства (10) неотрицательны. Стало быть, эта правая часть равна нулю. Таким образом, неравенство (10) имеет следующую конкретизацию: $\sup\limits_{a \in A(\overline{z})} \overline{p} \cdot a \leqslant 0$. Ввиду очевидных вложений $A_s(\overline{z}^s) \subseteq A(\overline{z})$, $s \in R$, получаем

$$\overline{p} \cdot w^s \leqslant 0, \quad w^s \in A_s(\overline{z}^s), \quad s \in R, \tag{11}$$

откуда с учётом определения множеств $A_s(\overline{z}^s)$ следуют импликации

$$t_s(z^s) > t_s(\overline{z}^s) \Rightarrow \overline{p} \cdot u^s \leqslant \overline{p} \cdot v^s, \quad z^s \in Z_s,\ s \in R. \tag{12}$$

Покажем, что для объёмов экспорта и импорта в рассматриваемом распределении $\overline{z}$ из ядра $C_F(M)$ в ценах $\overline{p}$ для каждого из регионов выполняются точные равенства $\overline{p} \cdot \overline{u}^s = \overline{p} \cdot \overline{v}^s$, $s \in R$. Как будет видно из дальнейшего, для этого достаточно установить справедливость неравенств $\overline{p} \cdot \overline{u}^s \leqslant \overline{p} \cdot \overline{v}^s$ для всех $s \in R$. Итак, пусть s — произвольный регион модели M. Зафиксируем какой-либо набор $z^s_* \in P_s(\overline{z}^s)$ и положим $z^s(\mu) = \mu z^s_* + (1-\mu)\overline{z}^s$, где $\mu \in (0,1)$ произвольное. Ввиду линейности целевых функций t_s выполняются неравенства $t_s(z^s(\mu)) > t_s(\overline{z})$ для всех $\mu \in (0,1)$. В силу (12) из этих неравенств вытекают неравенства

$$\overline{p} \cdot u^s(\mu) \leqslant \overline{p} \cdot v^s(\mu), \quad \mu \in (0,1), \tag{13}$$

где $u^s(\mu)$ и $v^s(\mu)$ — объёмы экспорта и импорта в региональном плане $z^s(\mu)$. Далее, из построения $z^s(\mu)$ вытекают равенства

$$u^s(\mu) = \overline{u}^s + \mu(u^s_* - \overline{u}^s), \quad v^s(\mu) = \overline{v}^s + \mu(v^s_* - \overline{v}^s)$$

для каждого $\mu \in (0,1)$. Переходя в (13) к пределу при $\mu \to 0$, получаем требуемое неравенство $\overline{p} \cdot \overline{u}^s \leqslant \overline{p} \cdot \overline{v}^s$. Тем самым в силу произвольности выбора s справедливы неравенства

$$\overline{p} \cdot \overline{u}^s \leqslant \overline{p} \cdot \overline{v}^s, \quad s \in R, \tag{14}$$

суммируя которые по $s \in R$, получаем

$$\overline{p} \cdot \sum_{s \in R} \overline{u}^s \leqslant \overline{p} \cdot \sum_{s \in R} \overline{v}^s.$$

Ясно, что это неравенство вместе с требованием $\sum\limits_{s\in R}\overline{u}^s \geqslant \sum\limits_{s\in R}\overline{v}^s$ и условием $\overline{p}\geqslant 0$ влечёт

$$\overline{p}\cdot\sum_{s\in R}\overline{u}^s=\overline{p}\cdot\sum_{s\in R}\overline{v}^s.$$

Однако с учётом (14) из последнего равенства следуют искомые соотношения: для каждого региона $s\in R$ выполняется равенство $\overline{p}\cdot\overline{u}^s=\overline{p}\cdot\overline{v}^s$.

Для завершения доказательства равновесности распределения $\overline{z}$ остаётся установить справедливость импликаций

$$t_s(z^s)>t_s(\overline{z}^s)\Rightarrow\overline{p}\cdot u^s<\overline{p}\cdot v^s,\quad z^s\in Z_s. \tag{15}$$

Пусть набор $z^s=(x^s,u^s,v^s,\lambda_s)$ принадлежит множеству $P_s(\overline{z}^s)$. Согласно (11) имеем неравенство $\overline{p}\cdot u^s\leqslant\overline{p}\cdot v^s$. Допустим, что здесь реализуется равенство $\overline{p}\cdot u^s=\overline{p}\cdot v^s$. В силу строгой автаркичности регионов модели M существует набор $\widehat{z}^s=(\widehat{x}^s,\widehat{u}^s,\widehat{v}^s,\widehat{\lambda}_s)\in Z_s$ такой, что $\widehat{u}^s\gg\widehat{v}^s$. Для каждого числа $\mu\in(0,1)$ определим набор $\widehat{z}^s(\mu)=z^s+\mu(\widehat{z}^s-z^s)$. Ввиду непрерывности целевой функции t_s при достаточно малом значении $\mu=\widehat{\mu}$ набор $\widehat{z}^s(\widehat{\mu})$ строго предпочитается $\overline{z}^s$, т. е. выполняется включение $\widehat{z}^s(\widehat{\mu})\in P_s(\overline{z}^s)$. Значит, согласно соотношениям (11) имеем

$$\overline{p}\cdot(u^s+\widehat{\mu}(\widehat{u}^s-u^s))\leqslant\overline{p}\cdot(v^s+\widehat{\mu}(\widehat{v}^s-v^s)),$$

но вследствие условия $\overline{p}\in\mathbb{R}^n_+\setminus\{0\}$ из предположений $\widehat{u}^s\gg\widehat{v}^s$, $\widehat{\mu}>0$ и равенства $\overline{p}\cdot u^s=\overline{p}\cdot v^s$ получаем неравенство

$$\overline{p}\cdot((u^s-v^s)+\widehat{\mu}(\widehat{u}^s-\widehat{v}^s))=\widehat{\mu}\overline{p}\cdot(\widehat{u}^s-\widehat{v}^s)>0,$$

противоречащее (11). Тем самым установлена справедливость (15). Теорема 1 доказана.

Финансирование работы

Исследование выполнено в рамках Программы фундаментальных научных исследований СО РАН (проект № FWNF–2022–2019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Гранберг А. Г., Суслов В. И., Суспицын С. А.** Многорегиональные системы: Экономико-математическое исследование. Новосибирск: Наука. Сиб. науч. изд-во, 2007. 372 с.
2. **Экланд И.** Элементы математической экономики. М.: Мир, 1983. 248 с.

3. **Vasil'ev V. A.** Walras equilibrium in multiregionl economic systems // Proc. 2017 Int. Multi-Conf. Engineering, Computer, and Informmtion Sciences (IEEE SIBIRCON 2017) (Novosibirsk, Russia, Sept. 18–22, 2017). Piscataway: IEEE, 2017. P. 176–181.
4. **Васильев В. А.** О равновесиях Эдджворта для некоторых видов неклассических рынков // Модели и методы оптимизации. Новосибирск: Наука, 1994. С. 3–52. (Тр. РАН. Сиб. отделение, Инст. математики; Т. 28).
5. **Суслов В. И.** Измерение эффектов межрегиональных взаимодействий: Модели, методы, результаты. Новосибирск: Наука. Сиб. отд-ние, 1991. 250 с.
6. **Рубинштейн А. Г.** Моделирование экономических взаимодействий в территориальных системах. Новосибирск: Наука. Сиб. отд-ние, 1983. 240 с.
7. **Обэн Ж.-П.** Нелинейный анализ и его экономические приложения. М.: Мир, 1988. 264 с.
8. **Голдман А. Дж.** Теоремы разложения и отделимости для многогранных выпуклых множеств // Линейные неравенства и смежные вопросы. М.: Изд-во иностр. лит., 1959. С. 162–171.
9. **Рокафеллар Р. Т.** Выпуклый анализ. М.: Мир, 1973. 472 с.
10. **Иоффе А. Д., Тихомиров В. М.** Теория экстремальных задач. М.: Наука, 1974. 480 с.

Васильев Валерий Александрович

Статья поступила
14 мая 2025 г.
После доработки —
11 июня 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 102–117

UDC 333.1:519.86 DOI: 10.33048/daio.2025.32.835

THE FUZZY CORE AND WALRAS EQUILIBRIA OF A MODEL FOR SPATIAL ECONOMY

V. A. Vasil'ev

Sobolev Institute of Mathematics,
4 Acad. Koptyug, 4, 630090 Novosibirsk, Russia
E-mail: vasilev@math.nsc.ru

Abstract. We consider some assumptions guaranteeing equivalence of unblockable and equilibrium plans in the famous model of spatial economy proposed by Acad. A. G. Granberg and his team. A continuous version of Edgeworth conjecture, namely the coincidence of the fuzzy core and equilibrium allocations, is studied. It is worth emphasizing that, besides having their own value, the coincidence conditions are of substantial interest for the equilibrium existence problem in spatial economies with unbounded regional activities. The fuzzy core equivalence theorem presented in the paper is based on the assumptions of strict regional autarchy, which is an analog of the Slater condition, and so-called regional unboundedness by functional. The main requirement for the regional system as a whole is the absence of "cornucopia". Bibliogr. 10.

Keywords: spatial economy, Walrasian equilibrium, fuzzy core, autarchy, "cornucopia".

References

1. **A. G. Granberg, V. I. Suslov,** and **S. A. Suspitsyn,** *Multi-Regional Systems: Economical and Mathematical Research* (Nauka, Sib. Nauchn. Izd., Novosibirsk, 2007) [Russian].
2. **I. Ekeland,** *Éléments d'Économie Mathématique* (Herman, Paris, 1979 [French]; Mir, Moscow, 1983 [Russian]).
3. **V. A. Vasil'ev,** Walras equilibrium in multiregionl economic systems, in *Proc. 2017 Int. Multi-Conf. Engineering, Computer, and Informmtion Sciences (IEEE SIBIRCON 2017)* (Novosibirsk, Russia, Sept. 18–22, 2017) (IEEE, Piscataway, 2017), pp. 176–181.

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

4. V. A. Vasil'ev, On Edgeworth equilibria for non-classic markets of some kinds, in *Models and Optimization Methods* (Nauka, Novosibirsk, 1994), pp. 3–52 (Tr. RAN, Sib. Otd., Inst. Mat., V. 28) [Russian] [*Sib. Adv. Math.* **6** (3), 96–150 (1996)].

5. V. I. Suslov, *Measuring the Effects of Interregional Interactions: Models, Methods, Results* (Nauka, Sib. Otd., Novosibirsk, 1991) [Russian].

6. A. G. Rubinshtein, *Modeling Economic Interactions in Territorial Systems* (Nauka, Sib. Otd., Novosibirsk, 1983) [Russian].

7. J.-P. Aubin, *L'Analyse Non Linéaire et Ses Motivations Économiques* (Masson, Paris, 1984 [French]; Mir, Moscow, 1988 [Russian]).

8. A. J. Goldman, Resolution and separation theorems for polyhedral convex sets, in *Linear Inequalities and Related Systems* (Princeton Univ. Press, Princeton, NJ, 1956), pp. 41–52; (Izd. Inostr. Lit., Moscow, 1959), pp. 162–171 [Russian].

9. R. T. Rockafellar, *Convex Analysis* (Princeton Univ. Press, Princeton, NJ, 1970; Mir, Moscow, 1973 [Russian]).

10. A. D. Ioffe and **V. M. Tikhomirov,** *Theory of Extremal Problems* (Nauka, Moscow, 1974) [Russian].

Valery A. Vasil'ev

Received May 14, 2025
Revised June 11, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 118–137

УДК 519.8 DOI: 10.33048/daio.2025.32.816

ПРИБЛИЖЁННЫЙ АЛГОРИТМ РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ ПО НЕОДНОРОДНЫМ ПРОЦЕССОРАМ С ЗАДЕРЖКАМИ ПРИ ПЕРЕДАЧЕ ДАННЫХ

А. В. Демаков[1, a], *А. В. Кононов*[2, b]

[1] Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
[2] Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: [a] a.demakov@g.nsu.ru, [b] alvenko@math.nsc.ru

Аннотация. Изучается задача распределения заданий по вычислительным серверам с учётом начальной загрузки данных для их выполнения. Назначение задания на сервер, который не содержит нужного блока данных, ведёт к расходам времени на передачу блока по сети. Чем больше блоков передаётся по сети, тем больше добавка к длительности задания. Требуется минимизировать общее время выполнения всех заданий.

Для рассматриваемой задачи предложен 2-приближённый алгоритм, который использует решение задачи линейного программирования и достройку дробного решения до целого. Для вычислительных экспериментов рассмотрена ускоренная версия алгоритма. Проведены вычислительные эксперименты, которые показали, что алгоритм по качеству ответов сопоставим с известными алгоритмами. Ил. 7, библиогр. 16.

Ключевые слова: теория расписаний, приближённый алгоритм, длина расписания.

Введение

Рассмотрим задачу параллельной обработки данных, в которой требуется распределить задания по параллельным неоднородным процессорам с целью минимизировать общее время обслуживания. Обработка каждого задания требует наличия на сервере определённого блока данных. Назначение задания на сервер, который не содержит нужного блока, ведёт к расходам времени на передачу блока по сети. Такое задание будем называть *удалённым*. Если сервер содержит блок данных, то задание, назначенное на такой сервер, называется *локальным*. Чем больше блоков

передаётся по сети, тем больше добавка к длительности задания. Данная задача в упрощённой форме моделирует различные процессы параллельных вычислений, например обработку большого набора изображений, где каждое изображение может быть обработано независимо.

Задачи оптимального распределения заданий по параллельным процессорам интенсивно изучаются с 1970-х гг. Задача построения расписания минимальной длины NP-трудна уже в системе из параллельных идентичных процессоров [1]. В задачах с неоднородными процессорами время выполнения задания зависит и от задания, и от того процессора, на который оно назначено. В 1985 г. Поттс [2] предложил 2-приближённый алгоритм, основанный на округлении точного дробного решения релаксации соответствующей задачи целочисленного линейного программирования (ЦЛП), в случае, когда число машин фиксировано. Ленстра, Шмойс и Тардош [3] усилили этот результат и показали, что дробное решение может быть преобразовано в хорошее целочисленное за время, ограниченное полиномом от размера входа, и в том случае, когда число машин не фиксированно и является частью входа. Алгоритм, полученный в [3], также 2-приближённый. В той же статье авторы показали, что аппроксимация этой задачи с относительной погрешностью меньше $3/2$ влечёт совпадение классов P и NP.

Задачи, в которых требуется учесть предварительное распределение данных на серверах, изучались в статьях [4, 5], посвящённых оптимизации назначения заданий в системе Hadoop, разработанной в рамке вычислительной парадигмы MapReduce. Процесс MapReduce состоит из двух стадий. На первой стадии Map один из компьютеров получает входные данные и разделяет их на другие компьютеры для обработки. На второй стадии Reduce происходит свёртка обработанных данных. Главный сервер получает ответы и формирует результат. Вторая стадия не может начаться, пока не выполнены все задания на первой стадии.

Для эффективного распределения задач на стадии Map Фишер, Су и Ин [4] представили идеализированную модель Hadoop, которая называется задачей назначения заданий Hadoop. Для упрощения модели авторы предположили, что все серверы одинаковы, а время выполнения заданий на каждом сервере совпадает с загрузкой этого сервера. С учётом размещения входных блоков по серверам цель задачи — найти назначение, минимизирующее общее время выполнения всех заданий. В [4] авторы рассмотрели задачу с одинаковыми длительностями локальных заданий, доказали её NP-трудность и предложили полиномиальный алгоритм с абсолютной оценкой точности. Доказано, что алгоритм может ошибаться от оптимума не больше, чем на длительность одного удалённого задания в оптимальном решении. В [5] рассмотрена эта же задача с дополнением в виде начальной нагрузки серверов и предложен полиномиальный

эвристический алгоритм BAR. Оба алгоритма используют решение задачи о максимальном потоке для назначения локальных заданий и жадное улучшение решения. С учётом специфики модели Hadoop авторы обеих статей предполагают длительности всех заданий одинаковыми, что позволяет строить начальное решение потоковыми алгоритмами.

В [6] указывается, что хотя принято считать, что в модели Hadoop длительности всех заданий одинаковы, на практике часто случается перекос при выполнении заданий разными процессорами, который негативно влияет на общее время выполнения стадии Map. Существуют различные причины возникновения такого перекоса. Например, задачи Map обычно обрабатывают коллекцию записей в форме пар ключ-значение, одну за другой. В идеале время обработки не сильно различается от записи к записи. Однако в зависимости от приложения для обработки некоторых записей может потребоваться больше ресурсов ЦП и памяти, чем для других, например такая ситуация возникает при использовании приложения PageRank [7]. Другая причина в том, что хотя MapReduce — унарный оператор, его можно использовать для эмуляции n-арной операции путём логического объединения нескольких наборов данных в качестве одного входа [8]. Каждый набор данных может потребовать различной обработки, что приводит к многомодальному распределению времени выполнения задач. В [6] перечисляется и ряд других причин, ведущих к перекосу при выполнении заданий на первой стадии. Похожая причина указывается также в работе [9], где авторы рассмотрели обобщённую задачу для модели MapReduce с обеими стадиями и неоднородными серверами. В [9] предложены приближённые онлайн и оффлайн алгоритмы и проведено теоретическое исследование задачи.

В нашей работе рассматривается задача о назначениях заданий на стадии Map с неоднородными серверами и произвольными длительностями. В разд. 1 вводится формальная постановка задачи. В разд. 2 описывается приближённый алгоритм, который использует идею, предложенную в [3]. Метод состоит в решении задачи линейного программирования и достройке дробного решения до целого. В разд. 3 обсуждаются модификации алгоритма с целью улучшения его трудоёмкости, приводятся результаты вычислительных экспериментов и сравнение полученных результатов с результатами известных алгоритмов.

1. Постановка задачи

Пусть $J = \{1, \ldots, n\}$ — множество заданий, $S = \{1, \ldots, m\}$ — множество серверов. Общее время выполнения всех заданий на сервере не зависит от того, в каком порядке они выполняются, поэтому для составления расписания достаточно найти назначение заданий по серверам. Назначением заданий назовём отображение $A\colon J \to S$.

Длительность задания определяется функцией $w\colon S \times J \times \mathbb{Z}^+ \to \mathbb{Z}^+$. Длительность зависит не только от сервера и задания, но и от нагрузки сети, т. е. от количества данных, перемещаемых с одного сервера на другой. Если данные для задания находятся на выбранном сервере, то время выполнения задания будет ниже, чем если их требуется передать по сети с удалённого сервера. Локальное задание t на сервере s имеет длительность w_{ts}, а удалённое — $w_{ts} + f(p)$, где p — число удалённых заданий. Без ограничения общности будем считать, что функция f принимает в целых точках целочисленные значения. Так как количество передаваемых по сети данных увеличивает нагрузку сети, будем считать, что $f(p)$ увеличивается с ростом p. Тогда для задания t, сервера s и назначения A с количеством удалённых заданий p выполняется соотношение

$$w'_{ts} = \begin{cases} w_{ts}, & \text{если задание } t \text{ локальное для сервера } s, \\ w_{ts} + f(p) & \text{иначе.} \end{cases}$$

Размещение данных задано двудольным графом $G(J \cup S, E)$. Между вершинами $t \in J$ и $s \in S$ имеется ребро $(t, s) \in E$, если данные для задания t находятся на сервере s.

Нагрузка сервера s вычисляется по формуле $L_s^A = \sum\limits_{t\colon A(t)=s} w'_{ts} + L_s^{\text{init}}$, где L_s^{init} — нагрузка сервера в начальный момент времени. Требуется найти назначение A, минимизирующее время $C_{\max}(A) = \max\limits_{s \in S} L_s^A$ завершения всех работ. Назовём задачу поиска оптимального назначения задачей П. Отметим, что задача П является обобщением классической задачи минимизации длины расписания на различных параллельных приборах [11], которая NP-трудна в сильном смысле [12].

2. Алгоритм

В этом разделе для решения задачи П предлагается 2-приближённый алгоритм, основанный на решении серии задач линейного программирования и последующем преобразовании дробных решений в целочисленные. Сформулируем задачу П в виде задачи целочисленного линейного программирования (ЦЛП). Для этого введём переменные

$$x_{ts} = \begin{cases} 1, & \text{если задание } t \text{ назначено на сервер } s, \\ 0 & \text{иначе.} \end{cases}$$

С использованием введённых обозначений задача ЦЛП может быть записана следующим образом:

$$C_{\max} \to \min, \tag{1}$$

$$\sum_{t=1}^{|J|} x_{ts} w'_{ts} + L_s^{\text{init}} \leqslant C_{\max}, \quad s \in S, \tag{2}$$

$$\sum_{s=1}^{|S|} x_{ts} = 1, \quad t \in J, \tag{3}$$

$$x_{ts} \in \{0, 1\}, \quad t \in J,\ s \in S. \tag{4}$$

Левая часть неравенств (2) задаёт нагрузку каждого сервера. Равенства (3) гарантируют, что каждое задание назначено ровно на один сервер.

Так как длительность удалённых задач зависит от их числа p, при его изменении требуется пересчитать значения параметров w'_{ts}, поэтому будем решать задачу Π для каждого p отдельно. Обозначим через Π_p задачу Π, в которой число удалённых заданий не превышает p.

Введём двоичную матрицу $\Lambda = (\lambda_{ts})_{t\in J}^{s\in S}$, где

$$\lambda_{ts} = \begin{cases} 1, & \text{если задание } t \text{ локальное для сервера } s, \\ 0 & \text{иначе,} \end{cases}$$

и добавим в задачу (1)–(4) ограничение на число удалённых заданий:

$$\sum_{t=1}^{|J|} \sum_{s=1}^{|S|} x_{ts}(1 - \lambda_{ts}) \leqslant p. \tag{5}$$

Для фиксированного числа p задачу (1)–(5) обозначим через ЦЛП(p). Допустимое решение задачи ЦЛП(p) взаимно однозначно соответствует допустимому решению задачи Π_p, хотя значения целевых функций могут и отличаться. Действительно, пусть $\overline{x} = (x_{ts})_{t\in J}^{s\in S}$ — произвольное допустимое решение задачи ЦЛП(p). Положим $A(t) = s$ для всех $t \in J$ и $s \in S$ таких, что $x_{ts} = 1$. Ограничения (3) и (5) влекут, что назначение A является допустимым решением задачи Π_p. Пусть $C_{\max}(p, \overline{x})$ — значение целевой функции задачи ЦЛП(p) на решении $\overline{x}$. Поскольку $f(p)$ — неубывающая функция, имеем $C_{\max}(A) \leqslant C_{\max}(p, \overline{x})$. Более того, существует $p' \leqslant p$ такое, что $C_{\max}(A) = C_{\max}(p', \overline{x})$, где p' равно числу удалённых заданий в назначении A.

2.1. Параметрическое сокращение. Одним из широко известных методов построения приближённых алгоритмов для комбинаторных задач является решение релаксации задачи ЦЛП с последующим округлением оптимального дробного решения до целочисленного.

К сожалению, как показано в [13], задача (1)–(4) имеет неограниченный разрыв целочисленности, что верно и для задачи ЦЛП(p). Например, пусть есть одно задание, которое имеет одинаковую длительность s

на каждом сервере, и s серверов. В этом случае оптимальное решение релаксированной задачи имеет длину 1, а оптимальное решение целочисленной — s.

Чтобы обойти эту трудность, используем параметрическое сокращение. Пусть $T \in \mathbb{Z}^+$ — верхняя оценка длины расписания. Будем считать, что задание t не может выполняться на сервере s, если $w'_{ts} > T$. Пусть $S_T = \{(t,s) \mid w'_{ts} \leqslant T\}$. Тогда для фиксированных p и T будем искать решение следующей задачи:

$$C_{\max} \to \min, \tag{6}$$

$$\sum_{t\colon (t,s)\in S_T} x_{ts} w'_{ts} + L_s^{\text{init}} \leqslant C_{\max}, \quad s \in S, \tag{7}$$

$$\sum_{s\colon (t,s)\in S_T} x_{ts} = 1, \quad t \in J, \tag{8}$$

$$\sum_{t=1}^{|J|} \sum_{s=1}^{|S|} x_{ts}(1-\lambda_{ts}) \leqslant p, \tag{9}$$

$$x_{ts} \geqslant 0, \quad (t,s) \in S_T, \tag{10}$$

$$C_{\max} \leqslant T. \tag{11}$$

Задачу (6)–(11) обозначим через $\mathrm{LP}(T,p)$. Используя бинарный поиск, можно найти наименьшее значение параметра T, при котором существует допустимое решение. Пусть T^* и есть такое значение. Оно является нижней оценкой оптимального решения задачи Π_p. Найти допустимое решение можно, например, с помощью алгоритма Хачияна [14], который находит экстремальное решение задачи линейного программирования.

2.2. Свойства экстремальных решений. Пусть $|S_T| = m$, $|J| = n$. Определим, какими свойствами обладают экстремальные решения.

Лемма 1. *Любое экстремальное решение* $\mathrm{LP}(T,p)$ *содержит не более* $n+m+1$ *ненулевых переменных.*

Доказательство. Заметим, что решение экстремально, если $|S_T|+1$ линейно независимых ограничений обращаются в равенство. По крайней мере $|S_T|+1-(n+m+2)$ из них будут выбраны из четвёртого множества ограничений. Значит, $|S_T|+1-(n+m+2)$ переменных равны нулю. Следовательно, экстремальное решение имеет не более $n+m+1$ ненулевых переменных. Лемма 1 доказана.

Пусть x^* — экстремальное решение $\mathrm{LP}(T,p)$. Представим его двудольным графом $H = ((J,S),E)$ с множеством вершин $J \cup S$, при этом ребро

$(t, s) \in E$, если $x^*_{ts} > 0$. Далее вершины в доле S будем называть *вершинами-серверами*, а вершины в доле J — *вершинами-заданиями*.

Лемма 2. *Пусть n_c — число вершин-заданий, m_c — число вершин-серверов в некоторой компоненте связности C графа H. Тогда число рёбер в C не превышает $n_c + m_c + 1$.*

Доказательство. Пусть $S_c \subset S$ и $J_c \subset J$ — подмножества вершин-заданий и вершин-серверов, попавших в компоненту связности C. Положим $y_{ts} = x^*_{ts}$, $i \in J_c$, $j \in S_c$. Тогда y — экстремальное решение задачи $LP_C(T)$, ограниченной множествами S_c и J_c. По лемме 1 число рёбер в C не превышает $n_c + m_c + 1$. Лемма 2 доказана.

Если в решении x^* переменная x^*_{ts} равна 1 для некоторого s, то задание t называется *целым*, иначе — *дробным*. Соответствующие им вершины также будем называть *целыми* или *дробными*. Пусть задание t целое и $x^*_{ts} = 1$. Положим $A(t) = s$, т. е. назначим целое задание на ту же машину, что и в решении x^*.

Каждая вершина графа H, соответствующая целому заданию, инцидентна ровно одному ребру. Удалим из графа H все целые вершины-задания с инцидентными им рёбрами. Назовём полученный граф H'. Так как каждая компонента связности C' в графе H' получена из соответственной компоненты C связности графа H удалением одинакового числа вершин и рёбер, утверждение леммы 2 остаётся в силе и для компоненты C'.

Ребра (t, s) графа H' будем называть *лёгкими*, если $1 - x_{ts} \leqslant 1/2$, иначе — *тяжёлыми*. Заметим, что каждая вершина-задание имеет не более одного инцидентного тяжёлого ребра. Сначала докажем две вспомогательные леммы.

Лемма 3. *Пусть C — произвольный цикл в графе H'. Тогда либо тяжёлые и лёгкие рёбра в нём чередуются, либо существует вершина-сервер с двумя лёгкими рёбрами.*

Доказательство. По построению граф H двудольный, следовательно, все циклы в нём содержат чётное число вершин и рёбер. Предположим, что в цикле C нет вершины-сервера с двумя лёгкими рёбрами. Напомним, что каждая вершина-задание содержит по крайней мере одно лёгкое ребро. Занумеруем все рёбра по циклу в направлении от вершины-задания к вершине-серверу, начиная с этого ребра. Тогда по предположению второе ребро, идущее от вершины-сервера, будет тяжёлым, а следующее за ним ребро, идущее от следующей вершины-задания, опять лёгким. В итоге получим, что все нечётные рёбра лёгкие, а чётные — тяжёлые. Лемма 3 доказана.

Лемма 4. *Пусть C — произвольная цепь в графе H' с чётным числом рёбер. Пусть крайние вершины цепи — вершины-серверы, а крайние рёбра тяжёлые. Тогда существует вершина-сервер с двумя лёгкими рёбрами.*

Доказательство. Так как вершины-задания не крайние, каждому из них инцидентны два ребра, и у них нет общих инцидентных рёбер. Поскольку каждая вершина-задание имеет не более одного инцидентного тяжёлого ребра, тяжёлых рёбер в C не больше, чем лёгких рёбер. Пусть не существует вершины-сервера с двумя лёгкими рёбрами. Тогда у каждой вершины-сервера кроме крайних одно ребро тяжёлое и одно лёгкое, т. е. количество тяжёлых и лёгких внутренних рёбер совпадает. Добавляя к количеству тяжёлых рёбер два крайних ребра, получаем противоречие с утверждением, что тяжёлых рёбер в C не больше, чем лёгких. Лемма 4 доказана.

Покажем, что для всех s существует назначение дробных заданий такое, что

$$\sum_{t\colon A(t)=s} (1 - x_{ts}) \leqslant 1. \tag{12}$$

Назовём такое назначение *равномерным*.

Лемма 5. *Пусть x^* — экстремальное решение $\mathrm{LP}(T, p)$. Тогда существует равномерное назначение дробных заданий.*

Доказательство. Достаточно показать, что на каждый сервер можно назначить не более одного дробного задания, связанного с сервером тяжёлым ребром, или не более двух заданий, связанных с сервером лёгкими рёбрами.

Рассмотрим произвольную связную непустую компоненту $C \subseteq H'$. Пусть r — число вершин в компоненте C. Лемма 2 влечёт, что число рёбер компоненты C ограничено числом $r + 1$. Следовательно, компонента C является либо деревом, либо псевдодеревом, либо графом с $r+1$ рёбрами. В последнем случае каждый такой граф может быть получен добавлением двух рёбер к некоторому остовному дереву на вершинах из C. После добавления первого ребра образуется псевдодерево, т. е. связный граф с одним циклом. Добавление второго ребра либо породит новый цикл, либо соединит путём две вершины первого цикла. Граф на r вершинах назовём *двуциклическим*, если он содержит $r+1$ рёбер и в нём нет вершин степени 1. Варианты таких графов представлены на рис. 1.

Построение равномерного назначения разобьём на несколько этапов.

Этап 1. Если в графе H' есть вершина степени 1, то это вершина-сервер, скажем, s. Тогда в решении x^* существует ровно одно дробное

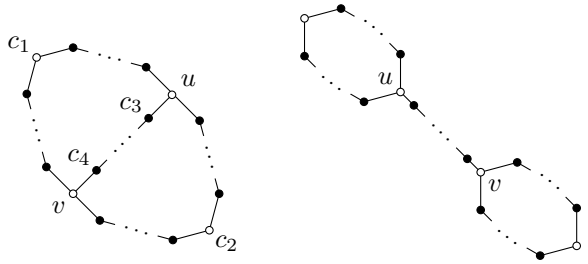


Рис. 1. Виды компонент связности графа H':
вершины-задания выделены чёрным, вершины-серверы — белым

задание t такое, что $0 < x^*_{ts} < 1$. Назначим это задание на сервер s и удалим из H' вершину-сервер, вершину-задание и все инцидентные ей рёбра. Повторим эту процедуру, пока в графе H' не останется вершин степени 1. Назовём полученный граф H''.

При удалении вершин и рёбер не образуется новых циклов. Следовательно, каждая связная компонента графа H'' либо простой цикл, либо двуциклический граф.

Этап 2. Если в результате удаления вершин степени 1 получился простой цикл, то выберем в нём произвольное совершенное паросочетание M и назначим каждое задание на тот сервер, который связан с ним ребром в M.

Осталось разобрать случай, когда после удаления вершин степени 1 получился двуциклический граф. Обозначим через u и v вершины степени больше чем 2 и назовём их *выделенными*. Отметим, что возможна ситуация, когда вершины u и v совпадают. В этом случае имеем одну вершину степени 4, скажем, v. Если выделенные вершины различны, то степень каждой из них равна 3.

Этап 3. Пусть одна из этих вершин (или обе) является вершиной-заданием. Тогда рассмотрим произвольный цикл, в который она входит. Найдём в этом цикле совершенное паросочетание M и назначим каждое задание на тот сервер, который связан с ним ребром в M. После удаления всех вершин цикла и инцидентных им рёбер получим компоненту, которая является либо деревом, либо псевдодеревом. Последовательно повторяя этапы 1 и 2, получим требуемое назначение.

Пусть обе вершины будут вершинами-серверами. Назовём вершину-сервер *особенной*, если ей инцидентно два лёгких ребра.

Этап 4. Предположим, что в двуциклическом графе вершины u и v лежат в разных циклах (правый граф на рис. 1) и в нём нет особенной вершины. Тогда по лемме 3 тяжёлые и лёгкие рёбра в циклах чередуются, следовательно, в каждом цикле существует паросочетание, состоящее

из лёгких рёбер. Назначим задания на серверы согласно двум таким паросочетаниям. Рассмотрим вершины задания на u-v-пути. По лемме 4 одно из крайних рёбер в этом пути лёгкое. Выберем совершенное паросочетание, содержащее это ребро, и назначим задания на серверы согласно этому паросочетанию.

Покажем, что если в двуциклическом графе вершины u и v лежат в одном цикле (левый граф на рис. 1), то всегда существует особенная вершина. Пусть это не так. Рассмотрим произвольный цикл C, в котором лежат вершины u и v. По предположению в нём нет особенной вершины. Тогда по лемме 3 тяжёлые и лёгкие рёбра в цикле C чередуются. Следовательно, в этом цикле в каждую вершину входят одно лёгкое и одно тяжёлое ребро. Однако, вершина v имеет степень три и входит в три цикла. При этом каждая пара рёбер, инцидентных вершине v, попадает в один из трёх циклов. Следовательно, существует цикл, в котором тяжёлые и лёгкие рёбра не чередуются, и согласно лемме 3 получаем противоречие с предположением, что в рассматриваемом графе нет особенной вершины.

Этап 5. Пусть в двуциклическом графе есть особенная вершина z, отличная от u и v. Рассмотрим путь из вершины z в одну из выделенных вершин, который не содержит вторую выделенную вершину. Пусть это будет путь $P_{[zu]}$ из вершины z в вершину u. Так как u и z — вершины-серверы, в $P_{[zu]}$ чётное число рёбер, следовательно, в нём существует два совершенных паросочетания. Выберем то паросочетание, которое содержит лёгкое ребро, инцидентное z. Назначим задания на серверы согласно этому паросочетанию и удалим все назначенные вершины-задания и инцидентные им рёбра. В результате получится либо псевдограф, либо две отдельные компоненты — псевдограф и простой цикл. Последовательно повторяя этапы 1 и 2, получим требуемое назначение.

Этап 6. Пусть в двуциклическом графе вершины u и v лежат в разных циклах и нет особенных вершин, отличных от u и v. По леммам 3 и 4 тяжёлые и лёгкие рёбра чередуются в каждом из циклов и на пути, соединяющем вершины u и v. Следовательно, в каждом цикле и пути есть совершенное паросочетание, состоящее из лёгких рёбер. Назначим задания на серверы согласно этим паросочетаниям. При этом на одну из выделенных вершин придётся два лёгких ребра, а все остальные вершины-серверы получат по одному лёгкому ребру.

Этап 7. Пусть в двуциклическом графе вершины u и v лежат в одном цикле, u — особенная вершина и нет особенных вершин, отличных от u и v. Так как u — особенная вершина, существует цикл C, в котором вершине u инцидентно два лёгких ребра. Пусть P_1 и P_2 — два пути из вершины v в вершину u в этом цикле. Оба пути имеют чётное число

рёбер. Выберем в каждом из них паросочетание, которое содержит ребро, инцидентное вершине u. Назначим задания на серверы согласно этим паросочетаниям. Пусть P_3 — путь из вершины v в вершину u, не лежащий в цикле C. Так как вершины u и v — вершины-серверы, в P_3 чётное число рёбер, следовательно, в нём существует два совершенных паросочетания. Выберем то паросочетание, которое не содержит ребра, инцидентного u. Назначим задания на серверы согласно этому паросочетанию.

В итоге получаем, что по экстремальному решению x^* на каждый сервер можно назначить не более одного дробного задания, связанного с сервером тяжёлым ребром, или не более двух заданий, связанных с сервером лёгкими рёбрами. Лемма 5 доказана.

Положим

$$\alpha = \frac{1}{m}\sum_{t\in J}\min_{s\in S} w_{ts}, \quad \beta = \max_{s\in S}\Big\{\sum_{t\colon (t,s)\in E} w_{ts} + L_s^{\text{init}}\Big\}.$$

Легко проверить, что α и β являются нижней и верхней оценкой длины оптимального расписания соответственно.

Алгоритм 1.

Вход: J, S, G, w_{ij}, f.

1: **for** $p \in [0, n]$ **do**
2: используя бинарный поиск, найти наименьшее значение $T \in [\alpha, \beta]$, для которого $\mathrm{LP}(T, p)$ имеет допустимое решение;
3: обозначить найденное значение через $T^*(p)$;
4: найти экстремальное решение задачи $\mathrm{LP}(T^*(p), p)$;
5: все целые работы назначить по найденному решению;
6: построить равномерное назначение дробных работ;
7: обозначить полученное назначение через A_p;
8: **return** $A^* = \{A_r \mid C_{\max}(A_r) = \min\limits_{p\in\overline{1,n}} C_{\max}(A_p)\}$;

Пусть

$$W(p) = \max_{s,t} w_{ts} + f(p). \tag{13}$$

Обозначим через OPT значение целевой функции в оптимальном решении.

Теорема 1. *Пусть число удалённых заданий в оптимальном решении равно p. Тогда $C_{\max}(A^*) \leqslant \min\{\mathrm{OPT} + W(p), 2\mathrm{OPT}\}$.*

Доказательство. Рассмотрим назначение работ A_p, найденное алгоритмом. Заметим, что если для фиксированного p существует решение x задачи $\mathrm{LP}(T', p)$ такое, что $C_{\max}(x) < T'$, то оно также является

допустимым решением задачи $\mathrm{LP}(T,p)$ с $T = C_{\max}(x)$. Следовательно, согласно шагу 2 алгоритма $T^*(p) \leqslant \mathrm{OPT}$.

Оценим нагрузку каждого сервера s в решении A_p. Пусть x^* — экстремальное решение $\mathrm{LP}(T^*(p),p)$. Имеем

$$L_s^{A_p} = L_s^{\mathrm{init}} + \sum_{t\colon A_p(t)=s} w'_{ts} = \\ = L_s^{\mathrm{init}} + \sum_{t\colon A_p(t)=s} w'_{ts}x^*_{ts} + \sum_{t\colon A_p(t)=s} w'_{ts}(1-x^*_{ts}) \leqslant \\ \leqslant T^*(p) + \sum_{t\colon A_p(t)=s} w'_{ts}(1-x^*_{ts}). \quad (14)$$

Последнее неравенство следует из ограничений (1) и (5) задачи $\mathrm{LP}(T,p)$. Так как на сервер s могут быть назначены только задания из множества $S_{T^*(p)}$, для любого задания t такого, что $A_p(t) = s$, имеем $w'_{ts} \leqslant T^*(p)$. Подставляя это неравенство в (14) и учитывая (12) и (13), получим

$$L_s^{A_p} \leqslant T^*(p) + \min\{T^*(p), W(p)\}.$$

Поскольку последнее неравенство выполнено для нагрузки каждого сервера, имеем

$$C_{\max}(A^*) \leqslant C_{\max}(A_p) \leqslant T^*(p) + \min\{T^*(p), W(p)\} \leqslant \\ \leqslant \mathrm{OPT} + \min\{\mathrm{OPT}, W(p)\},$$

откуда следует утверждение теоремы. Теорема 1 доказана.

Замечание 1. Отметим, что первый результат теоремы 1 обобщает результат из [4], полученный для случая, когда $w_{ts} = 1$ для всех s и t.

Трудоёмкость алгоритма определяется трудоёмкостью решения задачи линейного программирования на каждой итерации бинарного поиска для каждого возможного значения параметра p. Пусть $O(\tau)$ — трудоёмкость решения $\mathrm{LP}(T,p)$. Тогда алгоритм 1 имеет трудоёмкость $O(\tau n \times \log(\beta-\alpha))$, которая ограничена полиномом от размера входа задачи.

2.3. Модификации алгоритма при решении тестовых примеров. В предыдущем разделе было доказано, что алгоритм 1 строит расписание, длина которого не превышает двух длин оптимального расписания, и время работы алгоритма ограничено полиномом от размера входа индивидуальной задачи. Однако с практической точки зрения алгоритм 1 имеет два недостатка. Во-первых, он универсален и не учитывает специфику конкретного примера. Во-вторых, время его работы достаточно велико.

В алгоритме 1 после решения задачи линейного программирования происходит округление дробного решения до целого. Дробное решение представляется двудольным графом. Округление производится с помощью разбора висячих вершин. В тестовых примерах часто возникает ситуация, когда вершина-задание смежна с несколькими висячими вершинами-серверами. В этом случае выберем из них сервер с минимальной нагрузкой. Эта простая модификация не улучшает теоретическую оценку точности алгоритма, но приводит к улучшению решений для большинства примеров. Назовём алгоритм 1 с такой модификацией алгоритмом 2.

Более существенный недостаток алгоритма 1 — его трудоёмкость. Перебор решений для каждого значения параметра p требует много времени. В целях ускорения алгоритма было решено заменить перебор количества удалённых заданий жадной балансировкой нагрузки. На первой стадии решается задача $\mathrm{LP}(T, 0)$, т. е. строится дробное решение, в котором нет удалённых заданий. Далее по дробному решению одним из ранее описанных методов строится целочисленное решение. Затем применяется процедура локального спуска. Процедура состоит в перемещении задания с самого нагруженного сервера на любой менее нагруженный, которое не увеличивает длины расписания. Такую модификацию назовём ускоренным алгоритмом 2.

Эффект от обеих модификаций обсуждается в разд. 3.

3. Вычислительные эксперименты

В этом разделе представлены результаты вычислительных экспериментов для оценки эффективности представленного алгоритма.

Во всех тестовых примерах число серверов равняется 20, а число заданий изменяется от 60 до 200. Предполагается, что начальная нагрузка серверов равна нулю. Каждое задание выступает локальным ровно для трёх серверов. Для каждого задания случайно генерируем числа из усечённого нормального распределения со средним 10 и дисперсией 4. Числа округляем вниз до целого и применяем в качестве номеров серверов для размещения блоков данных. Выбор нормального распределения для размещения данных обусловлен необходимостью получить неравномерную загрузку серверов. В случае равномерной загрузки серверов данными задача сильно упрощается и, как правило, все применяемые далее алгоритмы находят оптимальное решение.

Рассматриваются два класса примеров: с одинаковыми локальными длительностями и с произвольными локальными длительностями. В первом случае все локальные длительности равны 20, длительность удалённых заданий вычисляется как $w'_{ij} = w_{ij} + C \cdot r$, где C — фактор сети, а r — число удалённых заданий в назначении. В построенных примерах

фактор сети изменяется от 0,1 до 3. Для примеров с произвольными длительностями значения параметров w_{ij} генерируются на отрезке $[1, 50]$ с равномерным распределением.

Алгоритмы для сравнения реализованы в виде компьютерной программы на языке C++ и решателя SCIP 8.0.3 под лицензией Apache 2.0 [15]. Расчёты проведены на персональном компьютере с процессором Intel (R) Core (TM) i5 9400 2,9 ГГц и 8 ГБ ОЗУ.

3.1. Сравнение методов округления дробного решения. В п. 2.3 описана модификация округления дробного решения. Как видно из графиков, представленных на рис. 2 и 3, за счёт более аккуратного назначения дробных заданий алгоритм 2 улучшает решения в среднем на 3% для задач с одинаковыми локальными длительностями работ и на 6% — для задач с произвольными локальными длительностями работ.

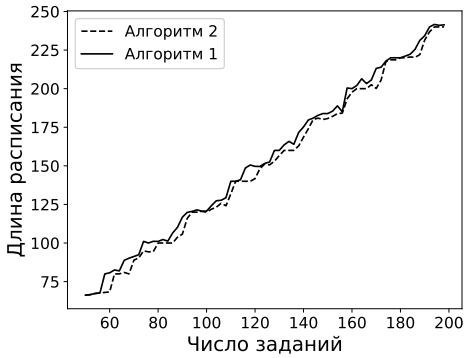


Рис. 2. Сравнение решений, полученных алгоритмами 1 и 2: одинаковые длительности

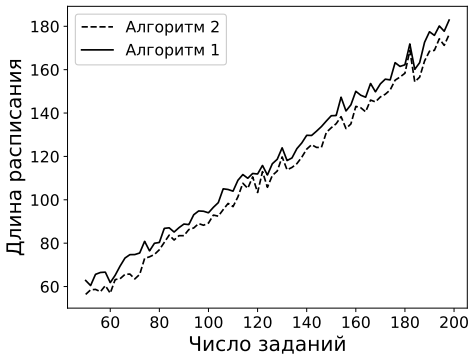


Рис. 3. Сравнение решений, полученных алгоритмами 1 и 2: произвольные длительности

3.2. Сравнение эффективности алгоритмов. Так как для рассматриваемой задачи в литературе известны алгоритмы только для примеров с единичными длительностями работ, проведём сравнение алгоритмов, представленных в этой работе, со стандартным алгоритмом системы Hadoop.

Стандартный алгоритм системы Hadoop (Hadoop Default Scheduler, HDS) [16] работает в режиме онлайн. Когда сервер простаивает, алгоритм выбирает локальное задание. Если такого нет, то случайное. Для сравнения с остальными алгоритмами была реализована оффлайн версия этого алгоритма. Для назначения очередного задания алгоритм выбирает наименее загруженный сервер. Затем нагрузки обновляются.

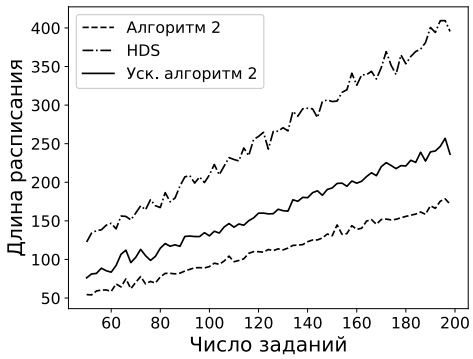


Рис. 4. Сравнение точности решений, $C = 0{,}1$

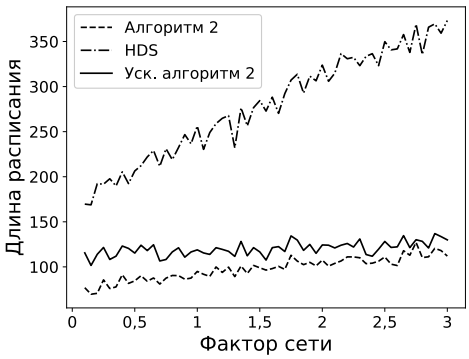


Рис. 5. Влияние фактора сети на точность различных алгоритмов

На рис. 4 приведено сравнение алгоритма 2 и его ускоренной версии с алгоритмом HDS. На диаграмме хорошо видно, что алгоритм HDS сильно проигрывает в точности даже ускоренной версии алгоритма 2.

Чтобы исследовать влияние перегруженности сети, сравниваем точность алгоритмов, изменяя сетевой коэффициент C от 0,1 до 3,0. Число работ в этих тестовых примерах равнялось 80. На рис. 7 видно, что увеличение коэффициента C приводит приводит к увеличению разрыва между точностью решений, полученных алгоритмом HDS и алгоритмом 2, в то время как разница целевых значений, полученных алгоритмом 2 и его ускоренной версией, снижается.

Для полноты картины сравним алгоритмы 1 и 2 со специализированными алгоритмами, разработанными для примеров с единичными длительностями работ.

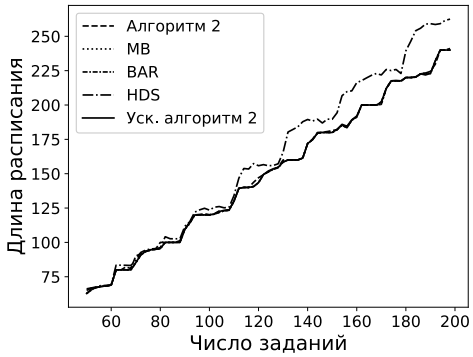


Рис. 6. Сравнение точности решений: одинаковые длительности

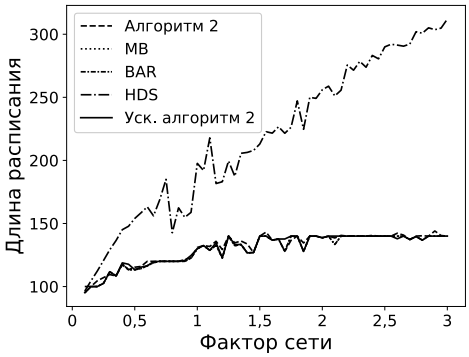


Рис. 7. Влияние фактора сети на точность различных алгоритмов: одинаковые длительности

Алгоритм Balance-Reduce (BAR) [5] стартует с решения задачи о максимальном потоке для назначения локальных заданий. Затем с помощью локальной процедуры Reduce решение улучшается перемещением заданий с наиболее загруженных серверов на менее загруженные. Алгоритм применим только для случая одинаковых локальных длительностей.

Алгоритм MaxCover-BalAssign (MB) [4] работает итеративно, создавая назначения, и выбирает из них наилучшее с точки зрения длины расписания. Каждая итерация состоит из двух частей. На первом этапе фиксируется ограничение на количество локальных заданий для серверов и решается задача о максимальном потоке. На втором этапе неназначенные задания распределяются жадной эвристикой. Алгоритм применим только для случая одинаковых локальных длительностей.

Проведённые эксперименты показывают, что алгоритм 2 и его ускоренная версия для примеров с одинаковой локальной длительностью заданий не уступают в точности алгоритмам HDS и BAR. Однако необходимо признать, что алгоритм 2 и даже его ускоренная версия сильно проигрывают потоковым алгоритмам в скорости нахождения решений.

Заключение

В настоящей работе рассмотрена задача распределения заданий по вычислительным серверам, в которой время обработки задания зависит как от нагрузки сети при передачи исходных данных, так и от машины, на которую назначается задание. Для рассматриваемой задачи предложен 2-приближённый алгоритм полиномиальной трудоёмкости, основанный на решении соответствующей задачи линейного программирования и последующем округлении полученного решения. Проведённые вычислительные эксперименты показали, что алгоритм сопоставим по качеству получаемых решений с известными специализированными алгоритмами, даже если локальные длительности всех работ одинаковы. Так как время работы алгоритма значительно проигрывает времени работы алгоритмов, основанных на решении задачи о максимальном потоке, предложена и протестирована ускоренная версия алгоритма.

Финансирование работы

Работа первого автора выполнена в рамках подготовки ВКР магистра под руководством второго автора в Новосибирском гос. университете. Работа второго автора выполнена в рамках гос. задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Garey M. R., Johnson D. S.** "Strong" NP-completeness results: Motivation, examples, and implication // J. ACM. 1978. V. 25, No. 3. P. 499–508.
2. **Potts C. N.** Analysis of a linear programming heuristic for scheduling unrelated parallel machines // Discrete Appl. Math. 1985. V. 10, No. 2. P. 155–164.
3. **Lenstra J. K., Shmoys D. B., Tardos É.** Approximation algorithms for scheduling unrelated parallel machines // Math. Program. 1990. V. 46. P. 259–271.
4. **Fischer M. J., Su X., Yin Y.** Assigning tasks for efficiency in Hadoop: Extended abstract // Proc. 22nd Annu. ACM Symp. Parallelism in Algorithms and Architectures (Thira Santorini, Greece, June 13–15, 2010). New York: ACM, 2010. P. 30–39. DOI: 10.1145/1810479.1810484.
5. **Jin J., Luo J., Song A., Dong F., Xiong R.** BAR: An efficient data locality driven task scheduling algorithm for cloud computing // Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing (Newport Beach, USA, May 23–26, 2011). Washington: IEEE Comput. Soc., 2011. P. 295–304. DOI: 10.1109/CCGrid.2011.55.
6. **Kwon Y., Balazinska M., Howe B., Rolia J.** A study of skew in MapReduce applications // Proc. Open Cirrus Summit 2011 (Moscow, Russia, June 1–3, 2011). Piscataway: IEEE, 2011. P. 1–5.
7. **Brin S., Page L.** The anatomy of a large-scale hypertextual Web search engine // Comput. Netw. ISDN Syst. 1998. V. 30, No. 1–7. P. 107–117.
8. **Gates A. F., Natkovich O., Chorpa S.** [et al.]. Building a high-level dataflow system on top of Map-Reduce: The Pig experience // Proc. VLDB Endow. 2009. V. 2, No. 2. P. 1414–1425.
9. **Moseley B., Dasgupta A., Kumar R., Sarlós T.** On scheduling in map-reduce and flow-shops // Proc. 23rd Annu. ACM Symp. Parallelism in Algorithms and Architectures (San Jose, CA, USA, June 4–6, 2011). New York: ACM, 2011. P. 289–298. DOI: 10.1145/1989493.1989540.
10. **Dean J., Ghemawat S.** MapReduce: Simplified data processing on large clusters // Commun. ACM. 2008. V. 51, No. 1. P. 107–113. DOI: 10.1145/1327452.1327492.
11. **Танаев В. С., Гордон В. С., Шафранский Я. М.** Теория расписаний: Одностадийные системы. М.: Наука, 1984. 384 с.
12. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
13. **Vazirani V. V.** Approximation algorithms. Heidelberg: Springer, 2001. 396 p.
14. **Хачиян Л. Г.** Полиномиальные алгоритмы в линейном программировании // Журн. вычисл. математики и мат. физики. 1980. Т. 20, № 1. С. 51–68.
15. **Bestuzheva K., Besançon M., Chen W.-K.** [et al.]. Enabling research through the SCIP Optimization Suite 8.0 // ACM Trans. Math. Softw. 2023. V. 49, No. 2. Article ID 22. 21 p. DOI: 10.1145/3585516.

16. White T. Hadoop: The definitive guide. Sebastopol, CA: O'Reilly Media, 2015. 756 p.

Демаков Алексей Владимирович
Кононов Александр Вениаминович

Статья поступила
22 октября 2024 г.
После доработки —
20 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 118–137

UDC 519.8 DOI: 10.33048/daio.2025.32.816

AN APPROXIMATE ALGORITHM FOR TASK ASSIGNMENT TO HETEROGENEOUS PROCESSORS WITH DELAYS IN DATA TRANSMISSION

A. V. Demakov[1, a] and *A. V. Kononov*[2, b]

[1] Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
[2] Sobolev Institute of Mathematics,
4 Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: [a] a.demakov@g.nsu.ru, [b] alvenko@math.nsc.ru

Abstract. We consider a problem of assigning tasks to computing servers taking into account the initial download of data for their execution. Assigning a task to a server that does not contain a required data block will take time to pass the block over the network. The more blocks are transmitted over the network, the more time is added to the task. The total time of all tasks is to be minimized.

For the problem under consideration, we propose a 2-approximate algorithm. Using a solution of some linear programming problem, this algorithm completes a fractional solution to an integer one. For computational experiments an enhanced version of the algorithm is considered. The computational experiments showed that the algorithm is comparable with some known algorithms in the quality of solution obtained. Illustr. 7, bibliogr. 16.

Keywords: scheduling theory, approximate algorithm, makespan.

References

1. **S. Brin** and **L. Page,** The anatomy of a large-scale hypertextual Web search engine, *Comput. Netw. ISDN Syst.* **30** (1–7), 107–117 (1998).
2. **M. R. Garey** and **D. S. Johnson,** "Strong" NP-completeness results: Motivation, examples, and implication, *J. ACM* **25** (3), 499–508 (1978).
3. **A. F. Gates, O. Natkovich, S. Chorpa,** [et al.], Building a high-level dataflow system on top of Map-Reduce: The Pig experience, *Proc. VLDB Endow.* **2** (2), 1414–1425 (2009).

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

4. **J. Dean** and **S. Ghemawat,** MapReduce: Simplified data processing on large clusters, *Commun. ACM* **51** (1), 107–113 (2008), DOI: 10.1145/1327452.1327492.
5. **M. J. Fischer, X. Su,** and **Y. Yin,** Assigning tasks for efficiency in Hadoop: Extended abstract, in *Proc. 22nd Annu. ACM Symp. Parallelism in Algorithms and Architectures* (Thira Santorini, Greece, June 13–15, 2010) (ACM, New York, 2010), pp. 30–39, DOI: 10.1145/1810479.1810484.
6. **J. Jin, J. Luo, A. Song, F. Dong,** and **R. Xiong,** BAR: An efficient data locality driven task scheduling algorithm for cloud computing, in *Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing* (Newport Beach, USA, May 23–26, 2011) (IEEE Comput. Soc., Washington, 2011), pp. 295–304, DOI: 10.1109/CCGrid.2011.55.
7. **Y. Kwon, M. Balazinska, B. Howe,** and **J. Rolia,** A study of skew in MapReduce applications, in *Proc. Open Cirrus Summit 2011* (Moscow, Russia, June 1–3, 2011) (IEEE, Piscataway, 2011), pp. 1–5.
8. **B. Moseley, A. Dasgupta, R. Kumar,** and **T. Sarlós,** On scheduling in map-reduce and flow-shops, in *Proc. 23rd Annu. ACM Symp. Parallelism in Algorithms and Architectures* (San Jose, CA, USA, June 4–6, 2011) (ACM, New York, 2011), pp. 289–298, DOI: 10.1145/1989493.1989540.
9. **J. K. Lenstra, D. B. Shmoys,** and **É. Tardos,** Approximation algorithms for scheduling unrelated parallel machines, *Math. Program.* **46**, 259–271 (1990).
10. **C. N. Potts,** Analysis of a linear programming heuristic for scheduling unrelated parallel machines, *Discrete Appl. Math.* **10** (2), 155–164 (1985).
11. **V. S. Tanaev, V. S. Gordon,** and **Ya. M. Shafranskii,** *Scheduling Theory: Single-Stage Systems* (Nauka, Moscow, 1984) [Russian].
12. **M. R. Garey** and **D. S. Johnson,** *Computers and Intractability. A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979; Mir, Moscow, 1982 [Russian]).
13. **V. V. Vazirani,** *Approximation Algorithms* (Springer, Heidelberg, 2001).
14. **L. G. Khachiyan,** Polynomial algorithms in linear programming, *Zh. Vychisl. Mat. Mat. Fiz.* **20** (1), 51–68 (1980) [Russian] [*USSR Comput. Math. Math. Phys.* **20** (1), 53–72 (1980), DOI: 10.1016/0041-5553(80)90061-0].
15. **K. Bestuzheva, M. Besançon, W.-K. Chen,** [et al.], Enabling research through the SCIP Optimization Suite 8.0, *ACM Trans. Math. Softw.* **49** (2), ID 22 (2023), DOI: 10.1145/3585516.
16. **T. White,** *Hadoop: The Definitive Guide* (O'Reilly Media, Sebastopol, CA, 2015).

Aleksey V. Demakov
Aleksandr V. Kononov

Received October 22, 2024
Revised August 20, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 138–145

УДК 519.178 DOI: 10.33048/daio.2025.32.836

ПОИСК ОСТОВНОГО ДЕРЕВА ГРАФА-КАКТУСА С МИНИМАЛЬНЫМ ИНДЕКСОМ ВИНЕРА

В. А. Клячин[a], *Е. В. Хижнякова*[b]

Волгоградский гос. университет,
Университетский пр., 100, 400062 Волгоград, Россия
E-mail: [a] klyachin.va@volsu.ru, [b] yakovleva.e.v@volsu.ru

Аннотация. Рассматривается задача построения остовного дерева в графе типа «кактус», минимизирующего значение индекса Винера — топологического индекса графа, который определяется суммой расстояний между всеми парами вершин. Граф-кактус представляет собой особый класс связных графов, в которых любые два простых цикла имеют не более одной общей вершины. Показывается, что в общем виде задача поиска остовного дерева с минимальным индексом Винера NP-полна, но для графов типа «кактус» данная задача решается за полиномиальное время. Приведён соответствующий алгоритм. Ил. 1, библиогр. 7.

Ключевые слова: индекс Винера, граф-кактус, остовное дерево.

Введение

Индекс Винера является важной характеристикой графов, определяемой как сумма длин кратчайших путей между всеми парами вершин. Существует немало работ, посвящённых исследованию и минимизации индекса Винера для различных классов графов. Например, работа [1] посвящена исследованию геометрических остовных деревьев, минимизирующих индекс Винера. В ней показано, что любое такое остовное дерево не имеет пересекающихся рёбер, и предложен алгоритм построения остовного дерева с минимальным индексом Винера для вершин, расположенных в выпуклой позиции. В [2] описываются графы, которые имеют экстремальный индекс Винера среди всех графов на n вершинах с k висячими вершинами.

В данной работе рассматривается задача поиска остовного дерева графа-кактуса с минимальным индексом Винера. Данная задача находит применение в различных сферах: например некоторые транспортные сети построены в виде графов-кактусов. Такие сети встречаются преимущественно в исторических центрах малых и средних городов.

Основная цель минимизации индекса Винера связана с поиском эффективного способа связи всех вершин графа такими путями, чтобы обеспечить наилучшую связь и минимум затрат на обслуживание сети. Этот показатель важен в различных приложениях, включая проектирование транспортных систем, электросетей, телекоммуникационных линий и логистику. Чем меньше индекс Винера, тем короче средние пути транспортировки товаров, пассажиров или услуг между различными пунктами назначения. Следовательно, снижаются расходы на топливо, рабочую силу и износ оборудования.

В органической химии графы-кактусы применяются для моделирования структуры полициклических соединений, таких как ароматические углеводороды (бензол, нафталин и др.) [3]. Индекс Винера коррелирует с температурой плавления, кипением, плотностью вещества и другими важными характеристиками. Чем больше значение индекса Винера, тем сложнее строение молекулы и, следовательно, сильнее проявляются физические эффекты [4–6].

Минимизация индекса Винера при выборе подходящего остовного дерева обеспечивает точное прогнозирование этих характеристик.

1. Постановка задачи

Граф-кактус — это связный неориентированный граф, в котором любые два простых цикла имеют не более одной общей вершины. Пусть $G(V, E)$ — кактус, где V и E — множества вершин и рёбер в G. Обозначим через $\delta_G(u, v)$ расстояние между вершинами u и v в G.

Индексом Винера $W(G)$ называется инвариант, определяемый как сумма кратчайших путей между всеми парами вершин графа:

$$W(G) = \sum_{\{u,v\} \in V(G)} \delta_G(u, v).$$

Для деревьев формула может выглядеть так:

$$W(T) = \sum_{e=(a,b) \in E(T)} |e| \cdot n(a) \cdot n(b), \tag{1}$$

где $n(a)$ — число вершин, которые в графе ближе к a, чем к b, $n(b)$ — число вершин, которые ближе к b, чем к a [4].

Задача 1. *Дан граф. Найти его остовное дерево с минимальным индексом Винера.*

Отметим, что задача 1 NP-полна. В [7] показано, что эта задача является NP-полной даже в случае, когда веса всех рёбер равны между собой.

Однако граф-кактус представляет собой особый граф, обладающий следующей характеристикой: любая вершина входит не более чем в один простой цикл. Другими словами, в таком графе нельзя построить два цикла, пересекающихся по общему ребру. Простейшие примеры графов-кактусов включают линейные цепочки, кольца, звёзды и некоторые комбинации этих структур.

Благодаря данному свойству задача 1 может быть решена за полиномиальное время для графов-кактусов.

Задача 2. *Дан граф-кактус. Найти его остовное дерево с минимальным индексом Винера.*

2. Индекс Винера остовных деревьев графа-кактуса

Для получения остовного дерева графа-кактуса необходимо разорвать каждый из его циклов, при этом из цикла удаляется ровно одно ребро.

Число остовных деревьев для кактусов вычисляется по формуле

$$\tau(G) = m_1 m_2 \cdots m_l,$$

где m_i — число рёбер в i-м цикле, l — число циклов.

Рассмотрим удаление ребра в одном из циклов с целью минимизации $W(G)$. Предположим, что из цикла $C = \{p_0, p_1, \ldots, p_{n-1}\}$ удаляется ребро (p_k, p_{k+1}). К любой из вершин p_i цикла может присоединяться подграф T_i (рис. 1).

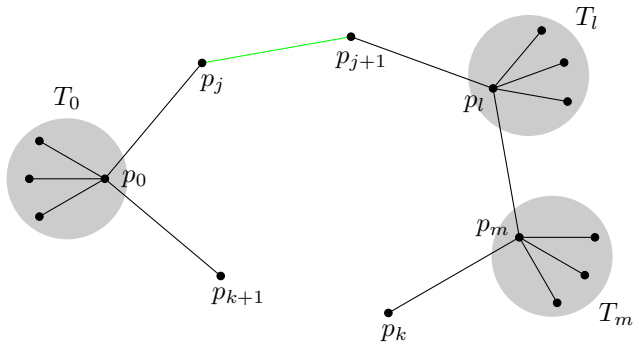


Рис. 1. Удаление ребра в одном из циклов графа-кактуса

Обозначим через $\delta_{p_i}(T_i)$ сумму длин путей от вершины p_i до каждой вершины из подграфа T_i:

$$\delta_{p_i}(T_i) = \sum_{q \in T_i} \delta_{T_i}(p_i, q).$$

Тогда индекс Винера графа G можно вычислить по формуле

$$W = \sum_{i=1}^{n-1} W(T_i) + \frac{1}{2}\sum_{i=1}^{n-1} \delta_{p_i}(T_i) \cdot (|V| - n(T_i)) + \\ + \sum_{i \in \overline{0,n-1} \setminus \{k\}} |u_i u_{i+1}| \cdot n(u_i) \cdot n(u_{i+1}). \quad (2)$$

Получается, что в формуле (2) только третье слагаемое зависит от выбора ребра, подлежащего удалению в текущем цикле, значит, именно его нужно сделать минимальным, чтобы минимизировать индекс Винера.

Следовательно, минимальный индекс Винера, который можно получить при разрыве одного из циклов графа-кактуса, может быть вычислен по формуле

$$W(G) = \sum_{i=1}^{n-1} W(T_i) + \frac{1}{2}\sum_{i=1}^{n-1} \delta_{p_i}(T_i) \cdot (|V| - n(T_i)) + \\ + \min_{k \in [0,n-1]} \sum_{i \in \overline{0,n-1} \setminus \{k\}} |u_i u_{i+1}| \cdot n(u_i) \cdot n(u_{i+1}). \quad (3)$$

Так как для решения задачи 2 необходимо удалить ровно одно ребро в каждом из циклов, с учётом формулы (1) получаем, что минимальный индекс Винера среди всех остовных деревьев графа-кактуса может быть вычислен по формуле

$$W(G) = \sum_{i=0}^{i<l} \min_{k \in C_i} \sum_{e=(a,b) \in C_i \setminus \{k\}} |e| \cdot n(a) \cdot n(b) + \sum_{e=(a,b) \notin C} |e| \cdot n(a) \cdot n(b), \quad (4)$$

где C — циклы в исходном графе-кактусе, l — число циклов, C_i — рёбра, принадлежащие i-му циклу, k — ребро, подлежащее удалению.

3. Алгоритм поиска остовного дерева с минимальным индексом Винера для графа-кактуса

Дан граф-кактус $G = (V, E)$.

Шаг 1. Найти все простые циклы в графе G. К списку C отнести рёбра, принадлежащие какому-либо циклу. Сгруппировать рёбра по циклам.

Шаг 2. Вычислить сумму для рёбер, не принадлежащих ни одному циклу:

$$W_1 = \sum_{(a,b) \notin \mathrm{C}} |ab| \cdot n(a) \cdot n(b).$$

Шаг 3. Для каждого цикла C_i из C найти ребро, подлежащее удалению. Для этого для каждого ребра k цикла C_i

Шаг 3.1. Вычислить

$$W_{2_k} = \sum_{e=(a,b)\in C_i \setminus k} |e| \cdot n(a) \cdot n(b).$$

Шаг 3.2. Взять минимальную из сумм, полученных на шаге 3.1. Соответствующее ребро удалить.

Шаг 3.3. Вычислить

$$W_2 = \sum_{i=0}^{i<l} \min_{k\in C_i} W_{2_k}.$$

Шаг 4. Минимальный индекс Винера положить равным

$$W = W_1 + W_2.$$

Все лишние рёбра удалены на шаге 3.2. Конец алгоритма.

Таким образом, удаляя по одному ребру из каждого цикла и минимизируя индекс Винера по формуле (4), решаем задачу 2.

Проведём оценку сложности алгоритма. Поиск всех простых циклов в графе-кактусе можно осуществить посредством поиска в глубину, сложность которого равна $O(V+E)$. Подсчёт числа вершин в двух компонентах связности после удаления ребра (a,b), т. е. величины $n(a)$ и $n(b)$, также можно найти алгоритмами поиска в ширину или в глубину. Таким образом, сложность алгоритма равна $O(4EV+4E^2)$. Учитывая, что в алгоритме рассматривается только граф-кактус, в котором число рёбер ограничено числом вершин, $E \leqslant \left\lfloor \frac{3}{2}(V-1) \right\rfloor$, оценка принимает более простой вид: $O(V^2)$.

Заключение

В работе рассмотрена задача поиска остовного дерева графа-кактуса с минимальным значением индекса Винера. Для её решения разработан специальный алгоритм, учитывающий структурные особенности рассматриваемого класса графов. Этот алгоритм обладает полиномиальной сложностью относительно размера входного графа, что делает его применимым для обработки больших наборов данных.

Предложенная методика имеет практическое применение в различных областях науки и техники, таких как химия (оценка физико-химических характеристик соединений), биология (анализ белковых сетей), транспортная логистика и телекоммуникационные сети.

Таким образом, полученные результаты вносят значительный вклад в развитие теории оптимизации графовых структур и позволяют эффективно решать прикладные задачи на графах специального вида.

Финансирование работы

Исследование выполнено за счёт бюджета Волгоградского гос. университета. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Abu-Affash A. K., Carmi P., Luwisch O., Mitchell J. S. B.** Geometric spanning trees minimizing the Wiener index // Algorithms and data structures. Proc. 18th Int. Symp. (Montreal, QC, Canada, July 31 – Aug. 2, 2023). Cham: Springer, 2023. P. 1–14. (Lect. Notes Comput. Sci.; V. 14079).
2. **Pandey D., Patra K. L.** Wiener index of graphs with fixed number of pendant or cut-vertices // Czechoslov. Math. J. 2022. V. 72, No. 2. P. 411–431.
3. **Klein D. J., Randić M.** Resistance distance // J. Math. Chem. 1993. V. 12. P. 81–95.
4. **Wiener H.** Structural determination of paraffin boiling points // J. Am. Chem. Soc. 1947. V. 69, No. 1. P. 17–20.
5. **Gutman I., Trinajstic N.** Graph theory and molecular orbitals. Total φ-electron energy of alternant hydrocarbons // Chem. Phys. Lett. 1972. V. 17, No. 4. P. 535–538.
6. **Bonchev D., Rouvray D. H.** Complexity in chemistry, biology, and ecology. New York: Springer, 2005. 348 p.
7. **Johnson D. S., Lenstra J. K., Rinnooy Kan A. H. G.** The complexity of the network design problem // Networks. 1978. V. 8, No. 4. P. 279–285.

Клячин Владимир Александрович
Хижнякова Екатерина Владимировна

Статья поступила
28 мая 2025 г.
После доработки —
11 июня 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 138–145

UDC 519.178 DOI: 10.33048/daio.2025.32.836

SEARCH FOR A SPANNING TREE IN A CACTUS GRAPH WITH THE MINIMUM WIENER INDEX

V. A. Klyachin [a] and *E. V. Khizhnyakova* [b]

Volgograd State University,
100 Universitetskiy Avenue, 400062, Volgograd, Russia
E-mail: [a] klyachin.va@volsu.ru, [b] yakovleva.e.v@volsu.ru

Abstract. We consider the problem of constructing a spanning tree in a cactus graph that minimizes the value of the Wiener index, which is a topological graph index determined by the sum of distances between all pairs of vertices. The cactus graph is a special class of connected graphs in which any two simple cycles have at most one vertex in common. It is shown that the problem of finding a spanning tree with the minimum Wiener index is NP-complete, while for cactus-type graphs this problem can be solved in polynomial time. The corresponding algorithm is provided. Illustr. 1, bibliogr. 7.

Keywords: Wiener index, cactus graph, spanning tree.

References

1. **A. K. Abu-Affash, P. Carmi, O. Luwisch,** and **J. S. B. Mitchell,** Geometric spanning trees minimizing the Wiener index, in *Algorithms and Data Structures*, Proc. 18th Int. Symp. (Montreal, QC, Canada, July 31 – Aug. 2, 2023) (Springer, Cham, 2023), pp. 1–14 (Lect. Notes Comput. Sci., V. 14079).
2. **D. Pandey** and **K. L. Patra,** Wiener index of graphs with fixed number of pendant or cut-vertices, *Czechoslov. Math. J.* **72** (2), 411–431 (2022).
3. **D. J. Klein** and **M. Randić,** Resistance distance, *J. Math. Chem.* **12**, 81–95 (1993).
4. **H. Wiener,** Structural determination of paraffin boiling points, *J. Am. Chem. Soc.* **69** (1), 17–20 (1947).
5. **I. Gutman** and **N. Trinajstic,** Graph theory and molecular orbitals. Total φ-electron energy of alternant hydrocarbons, *Chem. Phys. Lett.* **17** (4), 535–538 (1972).
6. **D. Bonchev** and **D. H. Rouvray,** *Complexity in Chemistry, Biology, and Ecology* (Springer, New York, 2005).

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

7. D. S. Johnson, J. K. Lenstra, A. H. G. Rinnooy Kan, The complexity of the network design problem, *Networks* **8** (4), 279–285 (1978).

Vladimir A. Klyachin
Ekaterina V. Khizhnyakova

Received May 28, 2025
Revised June 11, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 146–171

УДК 519.71 DOI: 10.33048/daio.2025.32.838

О СЛОЖНОСТИ РЕАЛИЗАЦИИ СИСТЕМЫ ИЗ ТРЁХ МОНОМОВ СХЕМАМИ КОМПОЗИЦИИ

С. А. Корнеев[1,2]

[1] Московский гос. университет им. М. В. Ломоносова,
Ленинские горы, 1, 119991 Москва, Россия
[2] Московский центр фундаментальной и прикладной математики,
Ленинские горы, 1, 119991 Москва, Россия
E-mail: korneev.sa.42@gmail.com

Аннотация. Исследуется сложность реализации систем мономов схемами композиции. Под сложностью в этой модели понимается минимальное число операций, необходимое для вычисления системы мономов по переменным, при этом допускается многократное использование результатов промежуточных вычислений. Основные результаты данной работы: для произвольной системы из трёх мономов без нулевых степеней установлена асимптотика сложности их совместной реализации схемами композиции; для произвольной системы из трёх мономов от трёх переменных без нулевых степеней установлена формула, выражающая сложность их совместной реализации схемами композиции с точностью до единицы. Табл. 1, библиогр. 21.

Ключевые слова: схема композиции, схема из функциональных элементов, система мономов, вычислительная сложность, сложность схемы.

В работе исследуется сложность (величина, равная минимальному числу операций) совместной реализации трёх мономов схемами композиции. Схемы композиции представляют собой вычислительную модель, в которой разрешено многократное применение результатов промежуточных вычислений, а в качестве единственной операции используется операция композиции двух мономов, предложенная А. И. Ширшовым [1]. Эта вычислительная модель исследовалась, например, в [2–8]. В данной работе получена формула, устанавливающая сложность реализации системы из трёх мономов от q переменных без нулевых степеней с точностью до слагаемого порядка q. Также для сложности реализации системы из трёх мономов от трёх переменных без нулевых степеней удалось получить верхнюю и нижнюю оценки, отличающиеся не более чем на единицу.

Введём основные определения, связанные со схемами композиции.

Мономом над множеством переменных $X = \{x_1, x_2, \ldots, x_q\}$ будем называть выражение вида $x_1^{a_1} x_2^{a_2} \cdots x_q^{a_q}$, где $a_1, a_2, \ldots, a_q$ — целые неотрицательные числа, причём $a_1 + a_2 + \cdots + a_q \geqslant 0$. Если $a_1 + a_2 + \cdots + a_q = 0$, то моном будем называть *нулевым* (с нулевым набором степеней) и обозначать символом 1. В дальнейшем будем рассматривать мономы над фиксированным множеством переменных $X = \{x_1, x_2, \ldots, x_q\}$. Будем говорить, что *моном* U_1 *содержится в мономе* U_2 (или что *моном* U_2 *содержит моном* U_1) и использовать обозначение $U_1 \leqslant U_2$, если все показатели степеней монома U_1 не превосходят соответствующих показателей степеней монома U_2. Если же это условие не выполнено, то будем говорить, что *моном* U_1 *не содержится в мономе* U_2 (*моном* U_2 *не содержит моном* U_1) и использовать обозначение $U_1 \not\leqslant U_2$.

Следуя [1] (см. также [2, 5]), определим понятие композиции мономов. Если для мономов

$$U = x_1^{a_{11}} x_2^{a_{12}} \cdots x_q^{a_{1q}}, \quad V = x_1^{a_{21}} x_2^{a_{22}} \cdots x_q^{a_{2q}}, \quad R = x_1^{b_1} x_2^{b_2} \cdots x_q^{b_q}$$

(возможно, $R = 1$) выполнены условия $R \leqslant U$ и $R \leqslant V$, то моном

$$(U, V)_R = \frac{UV}{R} = x_1^{a_{11}+a_{21}-b_1} x_2^{a_{12}+a_{22}-b_2} \cdots x_q^{a_{1q}+a_{2q}-b_q}$$

называется *композицией мономов* U *и* V *относительно монома* R. Отметим, что операция умножения мономов является частным случаем операции композиции, в этом случае R — нулевой моном. В дальнейшем будем считать, что применение операции композиции подразумевает выполнение условий $R \leqslant U$ и $R \leqslant V$.

Схемой композиции над системой мономов $M = \{U_1, \ldots, U_r\}$ будем называть такую последовательность мономов

$$(U_1, \ldots, U_r, U_{r+1}, \ldots, U_{r+n}), \tag{1}$$

что для любого $k = r+1, \ldots, r+n$ найдутся такие натуральные числа s и t и такой моном R_k (возможно, нулевой), что $s < k$, $t < k$ и

$$U_k = (U_s, U_t)_{R_k}. \tag{2}$$

Равенство (2) будем называть *правилом вычисления монома* U_k.

Если из контекста понятно, над какой системой мономов рассматривается схема композиции, или если это не имеет значения, то вместо «схема композиции над системой мономов» будем говорить «схема композиции» или просто «схема». Также отметим, что для монома из схемы композиции, вообще говоря, может существовать несколько разных правил вычисления. В дальнейшем, рассматривая любую схему композиции, будем считать, что соответствующий ей набор правил вычисления зафиксирован.

Если S — схема композиции вида (1), то под *сложностью* $l_{\mathrm{Sh}}(S)$ *схемы композиции* S будем понимать число n. Будем говорить, что *схема композиции* S *реализует моном* U, если $U \in S$. Аналогично будем говорить, что *схема композиции* S *реализует систему мономов* M, если для каждого монома U из системы M выполнено условие $U \in S$. Проще говоря, сложность схемы — это число мономов в ней, не считая мономов исходной системы, и схема реализует те мономы, которые она содержит.

Пусть M и M_0 — системы мономов. Следуя [2], положим $l_{\mathrm{Sh}}(M) = \min l_{\mathrm{Sh}}(S)$, где минимум берётся по всем схемам композиции, реализующим систему M над множеством переменных $\{x_1, \ldots, x_q\}$. Величину $l_{\mathrm{Sh}}(M)$ будем называть *сложностью реализации системы мономов* M *схемами композиции*. Аналогично величину $l_{\mathrm{Sh}}(M \mid M_0)$, определяемую равенством $l_{\mathrm{Sh}}(M \mid M_0) = \min l_{\mathrm{Sh}}(S)$, где минимум берётся по всем схемам композиции, реализующим систему мономов M над системой мономов M_0, будем называть *сложностью реализации системы мономов* M *над системой мономов* M_0 *схемами композиции*. Если выполнено условие $l_{\mathrm{Sh}}(S) = l_{\mathrm{Sh}}(M \mid M_0)$, то схему S будем называть *минимальной схемой композиции* (реализующей систему мономов M над системой мономов M_0). Заметим, что минимальная схема композиции не может содержать двух одинаковых элементов.

Для схемы (1) мономы $U_{r+1}, \ldots, U_{r+n}$ будем называть *существенными*, так как только они учитываются при подсчёте сложности схемы.

Заметим, что любой матрице из целых неотрицательных чисел можно поставить в соответствие систему мономов, в которой мономы соответствуют строкам матрицы, а переменные — столбцам, при этом элемент на пересечении i-й строки и j-го столбца равен степени j-й переменной в i-м мономе. Так, матрице

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1q} \\ a_{21} & a_{22} & \ldots & a_{2q} \\ \vdots & \vdots & \ldots & \vdots \\ a_{p1} & a_{p2} & \ldots & a_{pq} \end{pmatrix}$$

будет соответствовать система мономов

$$M_A = \left\{ x_1^{a_{11}} x_2^{a_{12}} \cdots x_q^{a_{1q}},\ x_1^{a_{21}} x_2^{a_{22}} \cdots x_q^{a_{2q}},\ \ldots,\ x_1^{a_{p1}} x_2^{a_{p2}} \cdots x_q^{a_{pq}} \right\}.$$

Сложностью $l_{\mathrm{Sh}}(A)$ *реализации матрицы* A будем называть сложность реализации соответствующей этой матрице системы мономов M_A. Таким образом, если матрице A соответствует система мономов M_A, то $l_{\mathrm{Sh}}(A) = l_{\mathrm{Sh}}(M_A)$. В дальнейшем будем иногда допускать «вольные» конструкции типа «сложность реализации матрицы над системой мономов» и соответствующие обозначения. Рассматривая системы из одного монома,

в соответствующих обозначениях будем опускать скобки и писать, например, $l_{\mathrm{Sh}}(U \,|\, V)$ вместо $l_{\mathrm{Sh}}(\{U\} \,|\, \{V\})$. Далее, говоря об асимптотике сложности реализации матриц, будем подразумевать выполнение условия $\sum a_{ij} \to \infty$.

Схемы композиции также можно определить как схемы из двухвходовых функциональных элементов (см., например, [9–11]), реализующих композицию мономов. При таком подходе каждый элемент вычисляет композицию двух мономов относительно некоторого монома R (возможно, нулевого, и, вообще говоря, своего для каждого элемента). Так как моном R существенно влияет на результат вычисления, функциональные элементы с разными мономами R считаются разными. Таким образом, рассматривая схемы композиции как схемы из функциональных элементов, мы, вообще говоря, рассматриваем схемы в бесконечном базисе. Кроме того, считается, что схема реализует все мономы, вычисляемые на выходах элементов, а также подаваемые на входы; другими словами, каждая вершина схемы считается её выходом.

Задача об исследовании различных свойств известной в алгебре операции композиции возникает естественным образом. В частности, самостоятельный интерес представляет изучение соответствующей вычислительной модели, так как она обладает различными интересными свойствами (см., например, [5, 8]). Заметим, что операция композиции является обобщением операции умножения: операция умножения является частным случаем операции композиции при $R = 1$. Тем самым имеет смысл сравнивать схемы композиции со схемами умножения, а также с другими вычислительными моделями (см., например, [12–14]). Важным достижением, полученным в этом направлении исследований, является перенос результата об асимптотике одной функции шенноновского типа из модели схем композиции на модель схем умножения [7]. В контексте данной статьи также важно отметить, что для схем умножения известны результаты об асимптотике сложности реализации матрицы размера $p \times 2$ [15], $2 \times q$ (аналогичный результат, вытекающий из предыдущего на основе принципа двойственности, см., например, [16–18]) и 3×3 [19] (см. также [20]); вместе с тем асимптотика сложности реализации матрицы размера $3 \times q$ пока неизвестна. Кроме того, во всех трёх указанных случаях, как и для матриц меньших размеров, асимптотика сложности имеет вид $\log_2 D(A)$, где $D(A)$ — максимум абсолютных величин миноров матрицы A, но уже для матрицы размера 4×4 известен пример [21] последовательности матриц, для которой асимптотика сложности имеет вид $\frac{4}{3} \log_2 D(A)$, поэтому неудивительно, что для матриц большего размера пока не удалось сформулировать правдоподобную гипотезу об асимптотике сложности их реализации. Таким образом, при увеличении размера

матриц трудность получения соответствующих результатов значительно возрастает. Это замечание справедливо и для модели схем композиции.

Введём несколько обозначений, которые будут использоваться в дальнейшем.

Если задана матрица $A = (a_{ij})$ размера $p \times q$, то положим

$$t_i(A) = \lceil \log \max_{1 \leqslant k \leqslant q} a_{ik} \rceil, \quad t_{ij}(A) = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{ik}}{a_{jk}} \right\rceil.$$

Если из контекста ясно, из какой матрицы берутся коэффициенты, то будем для краткости писать просто t_i, t_{ij}.

Для произвольных мономов

$$\begin{aligned} U_1 &= x_1^{a_{11}} x_2^{a_{12}} \cdots x_q^{a_{1q}}, \\ U_2 &= x_1^{a_{21}} x_2^{a_{22}} \cdots x_q^{a_{2q}}, \\ &\cdots\cdots\cdots\cdots\cdots \\ U_p &= x_1^{a_{p1}} x_2^{a_{p2}} \cdots x_q^{a_{pq}} \end{aligned}$$

введём обозначения

$$\begin{aligned} U_{\max}(U_1, U_2, \ldots, U_p) &= x_1^{\max a_{k1}} x_2^{\max a_{k2}} \cdots x_q^{\max a_{kq}}, \\ U_{\min}(U_1, U_2, \ldots, U_p) &= x_1^{\min a_{k1}} x_2^{\min a_{k2}} \cdots x_q^{\min a_{kq}}, \end{aligned}$$

где максимумы и минимумы берутся по всем $k = 1, 2, \ldots, p$. Если данная система мономов $\{U_1, U_2, \ldots, U_p\}$ обозначена через M, то будем также использовать обозначения

$$U_{\max}(M) = U_{\max}(U_1, U_2, \ldots, U_p), \quad U_{\min}(M) = U_{\min}(U_1, U_2, \ldots, U_p).$$

Сделаем ещё одно замечание, касающееся интерпретации используемых обозначений. Если задана некоторая последовательность мономов S, то можно рассматривать её по крайней мере в трёх разных контекстах.

1. Можно рассматривать последовательность мономов S просто как последовательность мономов. Здесь важно, что мономы идут в определённом порядке. При выделении подпоследовательности будем считать, что этот порядок сохраняется.

2. Можно рассматривать последовательность мономов S как схему над некоторой системой мономов M. При этом должен существовать соответствующий набор правил вычисления. Заметим, что сложность схемы S зависит от выбора системы мономов M. Таким образом, одной и той же последовательности мономов может соответствовать несколько схем разной сложности.

3. Можно рассматривать последовательность мономов S как систему мономов (множество, состоящее из элементов последовательности).

Здесь порядок мономов не важен, но важно, что мы можем говорить о схемах над системой мономов S и определять мономы $U_{\max}(S)$ и $U_{\min}(S)$.

Для доказательства основной теоремы потребуется ряд вспомогательных утверждений.

Первая лемма очевидна. По сути, она говорит о возможности синтеза схемы из двух (или нескольких) частей.

Лемма 1. *Если существуют схемы, реализующие систему мономов M_1 над системой мономов M_0 и систему мономов M над системой мономов M_1, то*

$$l_{\mathrm{Sh}}(M \mid M_0) \leqslant l_{\mathrm{Sh}}(M_1 \mid M_0) + l_{\mathrm{Sh}}(M \mid M_1).$$

В следующей лемме, доказанной в работе [6], сформулированы необходимые и достаточные условия существования правила вычисления.

Лемма 2. *Для мономов U, U_1 и U_2 совокупность условий*

$$U_1 \leqslant U, \quad U_2 \leqslant U, \quad U \leqslant U_1 U_2$$

выполняется тогда и только тогда, когда найдётся моном U_3 (возможно, нулевой) такой, что $U = (U_1, U_2)_{U_3}$.

Следующие две леммы устанавливают сложность реализации монома над мономом и монома над системой его переменных соответственно. Лемма 4 доказана в [2]. Доказательство леммы 3, а также другое доказательство леммы 4, можно найти в [6].

Лемма 3. *Пусть мономы $U = x_1^{a_1} x_2^{a_2} \cdots x_q^{a_q}$ и $V = x_1^{b_1} x_2^{b_2} \cdots x_q^{b_q}$ удовлетворяют условиям $a_k \geqslant b_k > 0$, $k = 1, \ldots, q$. Тогда*

$$l_{\mathrm{Sh}}(U \mid V) = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{b_k} \right\rceil.$$

Лемма 4. *Для монома $U = x_1^{a_1} \cdots x_q^{a_q}$ справедливо равенство*

$$l_{\mathrm{Sh}}(U) = \lceil \log \max_{1 \leqslant k \leqslant q} a_k \rceil + q - 1.$$

Следующая лемма позволяет заменять систему мономов одним мономом, что в некоторых случаях существенно упрощает доказательство нижних оценок сложности.

Лемма 5. *Пусть существует схема, реализующая систему мономов M над системой мономов $M_0 = \{U_0, U_1, U_2, \ldots, U_q\}$, моном U_0 содержится в каждом из мономов системы M, а каждый моном системы M_0 содержится в мономе U_0. Тогда $l_{\mathrm{Sh}}(M \mid M_0) = l_{\mathrm{Sh}}(M \mid U_0)$.*

Доказательство. Неравенство $l_{\mathrm{Sh}}(M \mid M_0) \leqslant l_{\mathrm{Sh}}(M \mid U_0)$ очевидно. Докажем обратное неравенство.

Пусть S — минимальная схема, реализующая систему мономов M над системой мономов M_0. Заменим каждый моном U в схеме S мономом $U' = U_{\max}(U, U_0)$ и удалим из полученной последовательности все мономы U_0, кроме первого. Полученную в результате последовательность мономов обозначим через S'.

Докажем, что последовательность S' образует схему, реализующую систему мономов M над мономом U_0. Ясно, что она начинается с монома U_0. Моном U_0 содержится в каждом из мономов системы M, значит, при такой замене мономы системы M не изменятся, поэтому каждый из них входит в последовательность S'. Осталось показать, что для последовательности S' существует подходящий набор правил вычисления. Пусть $U' \in S' \setminus \{U_0\}$ и правило вычисления для монома U (из которого был получен моном U') в схеме S имеет вид $U = (U_1, U_2)_{U_3}$. Тогда по лемме 2 выполнены условия $U_1 \leqslant U$, $U_2 \leqslant U$ и $U \leqslant U_1 U_2$. Легко проверить, что тогда $U'_1 \leqslant U'$, $U'_2 \leqslant U'$ и $U' \leqslant U'_1 U'_2$. Снова применяя лемму 2, получаем, что существует такой моном U'_3 (возможно, нулевой), что $U' = (U'_1, U'_2)_{U'_3}$. Таким образом, последовательность S' действительно образует схему, реализующую систему мономов M над мономом U_0. Нетрудно заметить, что $l_{\mathrm{Sh}}(S') \geqslant l_{\mathrm{Sh}}(S)$. Отсюда получаем

$$l_{\mathrm{Sh}}(M \mid M_0) = l_{\mathrm{Sh}}(S) = l_{\mathrm{Sh}}(S') \geqslant l_{\mathrm{Sh}}(M \mid U_0),$$

что и требовалось. Лемма 5 доказана.

Для доказательства основной теоремы будет использоваться лишь один частный случай леммы 5. Сформулируем соответствующее утверждение отдельно.

Лемма 6. *Пусть существует схема, реализующая моном U над системой мономов M_0, причём каждый моном системы M_0 содержится в мономе U. Тогда*

$$l_{\mathrm{Sh}}(U \mid M_0) \geqslant l_{\mathrm{Sh}}(U \mid U_{\max}(M_0)).$$

Доказательство. Из условия следует, что моном $U_{\max}(M_0)$ содержится в мономе U. Также очевидно, что каждый моном системы M_0 содержится в мономе $U_{\max}(M_0)$. Отсюда, применяя лемму 5, получаем

$$l_{\mathrm{Sh}}(U \mid M_0) \geqslant l_{\mathrm{Sh}}(U \mid M_0 \cup \{U_{\max}(M_0)\}) = l_{\mathrm{Sh}}(U \mid U_{\max}(M_0)),$$

что и требовалось. Лемма 6 доказана.

Следующая лемма устанавливает верхнюю и нижнюю оценки сложности реализации монома над системой мономов.

Лемма 7. *Пусть матрице $A = (a_{ij})$ размера $p \times q$, не содержащей нулевых столбцов, соответствует система мономов $M = \{U_1, U_2, \ldots, U_p\}$,*

каждый моном которой содержится в мономе $U_0 = x_1^{a_1} x_2^{a_2} \cdots x_q^{a_q}$. *Тогда*

$$l_{\mathrm{Sh}}(U_0 \,|\, M) \leqslant \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\max(a_{1k}, a_{2k}, \ldots a_{qk})} \right\rceil + \min(p, q) - 1,$$

$$l_{\mathrm{Sh}}(U_0 \,|\, M) \geqslant \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\max(a_{1k}, a_{2k}, \ldots a_{qk})} \right\rceil.$$

Доказательство. Верхняя оценка. Для каждого $j = 1, 2, \ldots, q$ выберем из системы M моном с наибольшим показателем переменной x_j и обозначим через M_0 систему, состоящую из всех таких мономов. Ясно, что число мономов в этой системе не превосходит $\min(p, q)$. Пусть $M_0 = \{V_1, V_2, \ldots, V_n\}$, и пусть этой системе мономов соответствует матрица $B = (b_{ij})$. Положим $V_0 = U_{\min}(V_1 \cdot V_2 \cdot \cdots \cdot V_n, U_0)$. Тогда каждый моном из системы M не превосходит монома V_0, который, в свою очередь, не превосходит монома U_0. Легко видеть, что $l_{\mathrm{Sh}}(V_0 \,|\, M_0) \leqslant n - 1$.

Если $U_0 = V_0$, то логарифм в формулировке леммы равен нулю. Тогда

$$l_{\mathrm{Sh}}(U_0 \,|\, M) = l_{\mathrm{Sh}}(V_0 \,|\, M) \leqslant n - 1 \leqslant \min(p, q) - 1,$$

что и требовалось.

Если же $U_0 \neq V_0$, то, используя леммы 1 и 3, получаем

$$\begin{aligned} l_{\mathrm{Sh}}(U_0 \,|\, M) &\leqslant l_{\mathrm{Sh}}(U_0 \,|\, M_0) \leqslant l_{\mathrm{Sh}}(U_0 \,|\, V_0) + l_{\mathrm{Sh}}(V_0 \,|\, M_0) \leqslant \\ &\leqslant \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\min(b_{1k} + b_{2k} + \cdots + b_{nk}, a_k)} \right\rceil + n - 1 \leqslant \\ &\leqslant \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\min(\max(b_{1k}, b_{2k}, \ldots, b_{nk}), a_k)} \right\rceil + \min(p, q) - 1 = \\ &= \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\max(a_{1k}, a_{2k}, \ldots, a_{pk})} \right\rceil + \min(p, q) - 1, \end{aligned}$$

что и требовалось.

Нижняя оценка. Положим $V_0 = U_{\max}(U_1, U_2, \ldots U_p)$. Тогда каждый моном из системы M не превосходит монома V_0, который, в свою очередь, не превосходит монома U_0. Используя леммы 3 и 5, получаем

$$\begin{aligned} l_{\mathrm{Sh}}(U_0 \,|\, M) &\geqslant l_{\mathrm{Sh}}(U_0 \,|\, M \cup \{V_0\}) = \\ &= l_{\mathrm{Sh}}(U_0 \,|\, V_0) = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_k}{\max(a_{1k}, a_{2k}, \ldots, a_{qk})} \right\rceil, \end{aligned}$$

что и требовалось. Лемма 7 доказана.

Следующие две леммы основные — они устанавливают соответственно верхнюю и нижнюю оценки сложности реализации системы из трёх мономов без нулевых степеней над мономом, состоящим из всех переменных.

Введём несколько дополнительных обозначений. Во-первых, для матрицы $A = (a_{ij})$ положим

$$X_{ijk} = 2^{\max(t_i - \max(t_{ij}, t_{ik}),\, t_j - \max(t_{ji}, t_{jk}))},$$
$$Y_{ijk} = 2^{\max(t_{ij}, t_{ik}, 0) - \max(t_{ik}, 0)},$$
$$d^{ij}_{kl} = \min(a_{il}, a_{jl}Y_{ijk}, a_{kl}, X_{ijk}Y_{ijk}).$$

Во-вторых, обобщим обозначения t_i и t_{ij} следующим образом с учётом добавления вспомогательного монома $U_0 = x_1^{a_{01}} x_2^{a_{02}} \cdots x_q^{a_{0q}}$:

$$t_0(A) = \lceil \log \max_{1 \leqslant k \leqslant q} a_{0k} \rceil, \quad t_{i0}(A) = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{ik}}{a_{0k}} \right\rceil, \quad i = 1, 2, 3.$$

Лемма 8. *Пусть $A = (a_{ij})$ — матрица размера $3 \times q$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) \leqslant \max_{(i_1,i_2,i_3) \in S_3} \Bigg(t_{i_1} + t_{i_2 i_1} + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d^{i_1 i_2}_{i_3 l}, d^{i_2 i_1}_{i_3 l}\right)} \right\rceil \Bigg) + 1.$$

Доказательство. Пусть $\{U_1, U_2, U_3\}$ — соответствующая матрице A система мономов. Сначала докажем вспомогательное неравенство

$$t_i - t_{ij} \leqslant t_j, \quad i, j = 1, 2, 3,\ i \neq j. \tag{3}$$

Пусть максимум в выражении для t_i достигается при $k = s$, т. е.

$$t_i = \lceil \log \max_{1 \leqslant k \leqslant q} a_{ik} \rceil = \lceil \log a_{is} \rceil.$$

Тогда, используя неравенство $\lceil x \rceil - \lceil y \rceil \leqslant \lceil x - y \rceil$, получаем

$$t_i - t_{ij} = \lceil \log a_{is} \rceil - \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{ik}}{a_{jk}} \right\rceil \leqslant \lceil \log a_{is} \rceil - \left\lceil \log \frac{a_{is}}{a_{js}} \right\rceil \leqslant \lceil \log a_{js} \rceil \leqslant t_j.$$

Неравенство (3) доказано.

Так как порядок строк матрицы никак не влияет на сложность её реализации, без ограничения общности будем считать, что выполнено условие

$$t_1 - t_{12} = \min_{\substack{i,j=1,2,3,\\ i \neq j}} (t_i - t_{ij}). \tag{4}$$

Тогда из (3) и (4) получаем

$$t_{12} \geqslant t_1 - (t_2 - t_{21}) \geqslant 0. \tag{5}$$

Положим

$$a_{0k} = \min(a_{1k}, a_{2k}, a_{3k}, 2^{t_2 - \max(t_{21}, t_{23})}), \quad U_0 = x_1^{a_{01}} x_2^{a_{02}} \cdots x_q^{a_{0q}},$$

$$b_{1k} = \min\left(a_{0k} \cdot 2^{t_{10} - \max(t_{13}, 0)}, a_{1k}, a_{3k}\right), \quad V_1 = x_1^{b_{11}} x_2^{b_{12}} \cdots x_q^{b_{1q}},$$
$$b_{2k} = \min\left(a_{0k} \cdot 2^{t_{20} - \max(t_{23}, 0)}, a_{2k}, a_{3k}\right), \quad V_2 = x_1^{b_{21}} x_2^{b_{22}} \cdots x_q^{b_{2q}},$$
$$k = 1, 2, \ldots, q.$$

Нетрудно заметить, что для мономов U_1, U_2, U_3, U_0, V_1, V_2 выполнены условия $U_0 \leqslant V_1$, $U_0 \leqslant V_2$, $V_1 \leqslant U_1$, $V_2 \leqslant U_2$, $V_1 \leqslant U_3$, $V_2 \leqslant U_3$. Из этих неравенств и отсутствия нулевых степеней во всех рассматриваемых мономах следует существование соответствующих схем. Отсюда, применяя лемму 1, получаем неравенство

$$l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) \leqslant l_{\mathrm{Sh}}(U_0 \,|\, x_1 x_2 \cdots x_q) + l_{\mathrm{Sh}}(V_1 \,|\, U_0) + \\ + l_{\mathrm{Sh}}(U_1 \,|\, V_1) + l_{\mathrm{Sh}}(V_2 \,|\, U_0) + l_{\mathrm{Sh}}(U_2 \,|\, V_2) + l_{\mathrm{Sh}}(U_3 \,|\, \{V_1, V_2\}). \quad (6)$$

Обозначим через Σ сумму первых пяти слагаемых в этой формуле. Тогда

$$l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) \leqslant \Sigma + l_{\mathrm{Sh}}(U_3 \,|\, \{V_1, V_2\}). \quad (7)$$

Используя лемму 3, будем оценивать сверху величину Σ. Для первого слагаемого получаем

$$l_{\mathrm{Sh}}(U_0 \,|\, x_1 x_2 \cdots x_q) = \lceil \log \max_{1 \leqslant k \leqslant q} a_{0k} \rceil \leqslant$$
$$\leqslant \lceil \log \max_{1 \leqslant k \leqslant q} \min\left(a_{2k}, 2^{t_2 - \max(t_{21}, t_{23})}\right) \rceil \leqslant \lceil \log \min\left(2^{t_2}, 2^{t_2 - \max(t_{21}, t_{23})}\right) \rceil =$$
$$= \lceil \log 2^{\min(t_2,\, t_2 - \max(t_{21}, t_{23}))} \rceil = t_2 - \max(t_{21}, t_{23}, 0). \quad (8)$$

Для второго и третьего слагаемых имеем

$$l_{\mathrm{Sh}}(V_1 \,|\, U_0) + l_{\mathrm{Sh}}(U_1 \,|\, V_1) =$$
$$= \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{b_{1k}}{a_{0k}} \right\rceil + \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{1k}}{b_{1k}} \right\rceil \leqslant \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{0k} \cdot 2^{t_{10} - \max(t_{13}, 0)}}{a_{0k}} \right\rceil +$$
$$+ \left\lceil \log \max_{1 \leqslant k \leqslant q} \max\left(\frac{a_{1k}}{a_{0k}} \cdot 2^{-(t_{10} - \max(t_{13}, 0))}, \frac{a_{1k}}{a_{1k}}, \frac{a_{1k}}{a_{3k}}\right) \right\rceil =$$
$$= (t_{10} - \max(t_{13}, 0)) + \max(t_{13}, 0) = t_{10}. \quad (9)$$

Для четвёртого и пятого слагаемых аналогично

$$l_{\mathrm{Sh}}(V_2 \,|\, U_0) + l_{\mathrm{Sh}}(U_2 \,|\, V_2) \leqslant t_{20}. \quad (10)$$

Используя неравенства (4) и (5), для t_{10} имеем

$$t_{10} = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{1k}}{a_{0k}} \right\rceil =$$
$$= \left\lceil \log \max_{1 \leqslant k \leqslant q} \max\left(\frac{a_{1k}}{a_{1k}}, \frac{a_{1k}}{a_{2k}}, \frac{a_{1k}}{a_{3k}}, \frac{a_{1k}}{2^{t_2 - \max(t_{21}, t_{23})}}\right) \right\rceil =$$

$$= \max(0, t_{12}, t_{13}, t_1 - (t_2 - \max(t_{21}, t_{23}))) = \max(t_{12}, t_{13}, 0) = t_{12}. \quad (11)$$

Далее, для t_{20} получаем

$$t_{20} = \left\lceil \log \max_{1\leqslant k\leqslant q} \frac{a_{2k}}{a_{0k}} \right\rceil =$$

$$= \left\lceil \log \max_{1\leqslant k\leqslant q} \max\left(\frac{a_{2k}}{a_{1k}}, \frac{a_{2k}}{a_{2k}}, \frac{a_{2k}}{a_{3k}}, \frac{a_{2k}}{2^{t_2-\max(t_{21},t_{23})}}\right)\right\rceil =$$

$$= \max(t_{21}, 0, t_{23}, t_2 - (t_2 - \max(t_{21}, t_{23}))) = \max(t_{21}, t_{23}, 0). \quad (12)$$

С помощью (8)–(12) приходим к оценке

$$\Sigma \leqslant (t_2 - \max(t_{21}, t_{23}, 0)) + t_{12} + \max(t_{21}, t_{23}, 0) = t_2 + t_{12}. \quad (13)$$

Осталось оценить величину $l_{\mathrm{Sh}}(U_3 \,|\, \{V_1, V_2\})$. Используя (4), (11), (12), а также неравенства $t_{10} \geqslant t_{13}$, $t_{20} \geqslant t_{23}$, для b_{1l}, b_{2l}, $l = 1, 2, \ldots, q$, имеем

$$b_{1l} = \min(a_{0l} \cdot 2^{t_{10}-\max(t_{13},0)}, a_{1l}, a_{3l}) =$$

$$= \min(a_{1l} \cdot 2^{t_{10}-\max(t_{13},0)}, a_{2l} \cdot 2^{t_{10}-\max(t_{13},0)}, a_{3l} \cdot 2^{t_{10}-\max(t_{13},0)},$$

$$2^{(t_2-\max(t_{21},t_{23}))+(t_{10}-\max(t_{13},0))}, a_{1l}, a_{3l}) =$$

$$= \min(a_{1l}, a_{2l} \cdot 2^{t_{10}-\max(t_{13},0)}, a_{3l}, 2^{(t_2-\max(t_{21},t_{23}))+(t_{10}-\max(t_{13},0))}) =$$

$$= \min(a_{1l}, a_{2l} \cdot 2^{t_{12}-\max(t_{13},0)}, a_{3l}, 2^{(t_2-\max(t_{21},t_{23}))+(t_{12}-\max(t_{13},0))}) =$$

$$= \min(a_{1l}, a_{2l} \cdot 2^{\max(t_{12},t_{13},0)-\max(t_{13},0)}, a_{3l},$$

$$2^{\max(t_1-\max(t_{12},t_{13}),t_2-\max(t_{21},t_{23}))+(\max(t_{12},t_{13},0)-\max(t_{13},0))}) =$$

$$= \min(a_{1l}, a_{2l} \cdot Y_{123}, a_{3l}, X_{123} \cdot Y_{123}) = d_{3l}^{12},$$

$$b_{2l} = \min(a_{0l} \cdot 2^{t_{20}-\max(t_{23},0)}, a_{2l}, a_{3l}) =$$

$$= \min(a_{1l} \cdot 2^{t_{20}-\max(t_{23},0)}, a_{2l} \cdot 2^{t_{20}-\max(t_{23},0)}, a_{3l} \cdot 2^{t_{20}-\max(t_{23},0)},$$

$$2^{(t_2-\max(t_{21},t_{23}))+(t_{20}-\max(t_{23},0))}, a_{2l}, a_{3l}) =$$

$$= \min(a_{1l} \cdot 2^{t_{20}-\max(t_{23},0)}, a_{2l}, a_{3l}, 2^{(t_2-\max(t_{21},t_{23}))+(t_{20}-\max(t_{23},0))}) =$$

$$= \min(a_{1l} \cdot 2^{\max(t_{21},t_{23},0)-\max(t_{23},0)}, a_{2l}, a_{3l},$$

$$2^{\max(t_1-\max(t_{12},t_{13}),t_2-\max(t_{21},t_{23}))+(\max(t_{21},t_{23},0)-\max(t_{23},0))}) =$$

$$= \min(a_{1l} \cdot Y_{213}, a_{2l}, a_{3l}, X_{213} \cdot Y_{213}) = d_{3l}^{21}.$$

Используя лемму 7, получаем

$$l_{\mathrm{Sh}}(U_3 \,|\, \{V_1, V_2\}) \leqslant \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{3l}}{\max(b_{1l}, b_{2l})} \right\rceil + 1 \leqslant$$

$$\leqslant \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{3l}}{\max\left(d_{3l}^{12}, d_{3l}^{21}\right)} \right\rceil + 1. \quad (14)$$

Наконец, подставляя (13) и (14) в (7), приходим к оценке

$$l_{\mathrm{Sh}}(A \,|\, x_1x_2\cdots x_q) \leqslant t_2 + t_{12} + \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{3l}}{\max\left(d_{3l}^{12}, d_{3l}^{21}\right)} \right\rceil + 1 \leqslant$$

$$\leqslant \max_{(i_1,i_2,i_3)\in S_3} \left(t_{i_1} + t_{i_2i_1} + \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2}, d_{i_3l}^{i_2i_1}\right)} \right\rceil \right) + 1,$$

что и требовалось. Лемма 8 доказана.

Следующая лемма устанавливает нижнюю оценку для той же самой величины.

Лемма 9. *Пусть $A = (a_{ij})$ — матрица размера $3 \times q$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A \,|\, x_1x_2\cdots x_q) \geqslant \max_{(i_1,i_2,i_3)\in S_3} \left(\max\left(t_{i_1} + t_{i_2i_1}, t_{i_2} + t_{i_1i_2}\right) + \right.$$

$$\left. + \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2}, d_{i_3l}^{i_2i_1}\right)} \right\rceil \right).$$

Доказательство. Пусть $\{U_1, U_2, U_3\}$ — соответствующая матрице A система мономов. Пусть S — произвольная схема, реализующая систему мономов $\{U_1, U_2, U_3\}$ над мономом $x_1x_2\cdots x_q$. Выделим из схемы S несколько подсхем в соответствии с табл. 1.

Таблица 1

$\langle 1\rangle$	$\langle 2\rangle$	$\langle 3\rangle$	$\langle 4\rangle$
$S_0 = \{U \in S \mid U \leqslant U_1,\ U \leqslant U_2,\ U \leqslant U_3\}$	S_0	$\{x_1x_2\cdots x_q\}$	$U \leqslant U_1,\ U \leqslant U_2,\ U \leqslant U_3,\ U \neq x_1x_2\cdots x_q$
$S_{13} = \{U \in S \mid U \leqslant U_1,\ U \leqslant U_3\}$	S_{13}	S_0	$U \leqslant U_1,\ U \not\leqslant U_2,\ U \leqslant U_3$
$S_{23} = \{U \in S \mid U \leqslant U_2,\ U \leqslant U_3\}$	S_{23}	S_0	$U \not\leqslant U_1,\ U \leqslant U_2,\ U \leqslant U_3$
$S_1 = \{U \in S \mid U \leqslant U_1\}$	$\{U_1\}$	S_{13}	$U \leqslant U_1,\ U \not\leqslant U_3$
$S_2 = \{U \in S \mid U \leqslant U_2\}$	$\{U_2\}$	S_{23}	$U \leqslant U_2,\ U \not\leqslant U_3$
$S_3 = \{U \in S \mid U \leqslant U_3\}$	$\{U_3\}$	$S_{13} \cup S_{23}$	$U \not\leqslant U_1,\ U \not\leqslant U_2,\ U \leqslant U_3$

Эту таблицу нужно читать построчно следующим образом. Выделим из схемы S подпоследовательность мономов $\langle 1\rangle$. Эта подпоследовательность мономов образует схему, реализующую систему мономов $\langle 2\rangle$ над системой мономов $\langle 3\rangle$. Любой существенный моном U этой схемы удовлетворяет условиям $\langle 4\rangle$. Напомним, что моном U называется существенным для схемы S над системой мономов M, если $U \in S \setminus M$, поэтому сложность схемы равна числу её существенных мономов.

Как нетрудно видеть из четвёртого столбца табл. 1, каждый существенный моном схемы S может быть существенным не более чем для одной из схем S_0, S_{13}, S_{23}, S_1, S_2, S_3. Отсюда

$$l_{\mathrm{Sh}}(S) \geqslant l_{\mathrm{Sh}}(S_0) + l_{\mathrm{Sh}}(S_{13}) + l_{\mathrm{Sh}}(S_{23}) + l_{\mathrm{Sh}}(S_1) + l_{\mathrm{Sh}}(S_2) + l_{\mathrm{Sh}}(S_3). \quad (15)$$

Положим

$$\begin{gathered} U_0 = x_1^{a_{01}} x_2^{a_{02}} \cdots x_q^{a_{0q}} = U_{\max}(S_0), \\ V_1 = x_1^{b_{11}} x_2^{b_{12}} \cdots x_q^{b_{1q}} = U_{\max}(S_{13}), \\ V_2 = x_1^{b_{21}} x_2^{b_{22}} \cdots x_q^{b_{2q}} = U_{\max}(S_{23}), \\ V_3 = U_{\max}(S_{13} \cup S_{23}) = U_{\max}(V_1, V_2). \end{gathered}$$

Используя леммы 3 и 6, получаем

$$\begin{gathered} l_{\mathrm{Sh}}(S_0) \geqslant l_{\mathrm{Sh}}(U_0 \,|\, x_1 x_2 \cdots x_q), \\ l_{\mathrm{Sh}}(S_{13}) \geqslant l_{\mathrm{Sh}}(V_1 \,|\, S_0) \geqslant l_{\mathrm{Sh}}(V_1 \,|\, U_0), \\ l_{\mathrm{Sh}}(S_{23}) \geqslant l_{\mathrm{Sh}}(V_2 \,|\, S_0) \geqslant l_{\mathrm{Sh}}(V_2 \,|\, U_0), \\ l_{\mathrm{Sh}}(S_1) = l_{\mathrm{Sh}}(U_1 \,|\, S_{13}) \geqslant l_{\mathrm{Sh}}(U_1 \,|\, V_1), \\ l_{\mathrm{Sh}}(S_2) = l_{\mathrm{Sh}}(U_2 \,|\, S_{23}) \geqslant l_{\mathrm{Sh}}(U_2 \,|\, V_2), \\ l_{\mathrm{Sh}}(S_3) = l_{\mathrm{Sh}}(U_3 \,|\, S_{13} \cup S_{23}) \geqslant l_{\mathrm{Sh}}(U_3 \,|\, V_3). \end{gathered}$$

Подставляя эти неравенства в (15), имеем

$$\begin{aligned} l_{\mathrm{Sh}}(S) \geqslant l_{\mathrm{Sh}}(U_0 \,|\, x_1 x_2 \cdots x_q) + l_{\mathrm{Sh}}(V_1 \,|\, U_0) + l_{\mathrm{Sh}}(V_2 \,|\, U_0) + \\ + l_{\mathrm{Sh}}(U_1 \,|\, V_1) + l_{\mathrm{Sh}}(U_2 \,|\, V_2) + l_{\mathrm{Sh}}(U_3 \,|\, V_3). \end{aligned}$$

Если считать мономы U_1, U_2 и U_3 зафиксированными, то правую часть последнего неравенства можно рассматривать как функцию, зависящую от мономов U_0, V_1 и V_2 (моном V_3 однозначно определяется через мономы V_1 и V_2). Обозначим эту функцию через Σ и будем искать её минимум. Для краткости записи введём обозначения:

$$\begin{gathered} s_0 = l_{\mathrm{Sh}}(U_0 \,|\, x_1 x_2 \cdots x_q), \quad s_{10} = l_{\mathrm{Sh}}(V_1 \,|\, U_0), \quad s_{20} = l_{\mathrm{Sh}}(V_2 \,|\, U_0), \\ s_1 = l_{\mathrm{Sh}}(U_1 \,|\, V_1), \quad s_2 = l_{\mathrm{Sh}}(U_2 \,|\, V_2), \quad s_3 = l_{\mathrm{Sh}}(U_3 \,|\, V_3). \end{gathered}$$

Тогда полученная оценка принимает вид

$$l_{\mathrm{Sh}}(S) \geqslant \Sigma = s_0 + s_{10} + s_{20} + s_1 + s_2 + s_3. \quad (16)$$

Опираясь на лемму 3, проведём следующее рассуждение. Если зафиксировать величины s_0, s_{10} и s_{20}, то для показателей степеней мономов U_0, V_1 и V_2 должны быть выполнены следующие неравенства:

$$\begin{gathered} a_{0k} \leqslant \min\left(2^{s_0}, a_{1k}, a_{2k}, a_{3k}\right), \\ b_{1k} \leqslant \min\left(a_{0k} \cdot 2^{s_{10}}, a_{1k}, a_{3k}\right), \end{gathered}$$

$$b_{2k} \leqslant \min(a_{0k} \cdot 2^{s_{20}}, a_{2k}, a_{3k}),$$
$$k = 1, 2, \ldots, q.$$

При увеличении этих показателей степеней без нарушения данных неравенств величины s_1, s_2 и s_3 не увеличатся. Таким образом, подход к поиску минимума функции Σ заключается в следующем: положим

$$a_{0k} = \min(2^{s_0}, a_{1k}, a_{2k}, a_{3k}),$$
$$b_{1k} = \min(a_{0k} \cdot 2^{s_{10}}, a_{1k}, a_{3k}),$$
$$b_{2k} = \min(a_{0k} \cdot 2^{s_{20}}, a_{2k}, a_{3k}),$$
$$k = 1, 2, \ldots, q,$$

и будем рассматривать Σ как функцию от переменных s_0, s_{10} и s_{20}. Значения s'_0, s'_{10} и s'_{20} назовём *оптимальными* для переменных s_0, s_{10} и s_{20} соответственно, если при их подстановке функция Σ принимает минимальное значение. Заметим, что оптимальных значений может быть несколько, но в рамках доказательства нам достаточно найти одну тройку таких значений. Для функции Σ и всех промежуточных функций, зависящих от переменных s_0, s_{10} и s_{20}, будем использовать штрих, чтобы обозначить подстановку в них оптимальных значений этих переменных. Таким образом, $\Sigma' = \Sigma(s'_0, s'_{10}, s'_{20}) = \min\limits_{s_0, s_{10}, s_{20}} \Sigma(s_0, s_{10}, s_{20})$.

Пусть значения переменных s_0 и s_{20} зафиксированы. При каком значении переменной s_{10} функция Σ будет принимать минимальное значение? Положим $s_{10} = 0$ и будем последовательно увеличивать её значение на единицу. При этом могут изменяться только слагаемые s_{10}, s_1 и s_3. Для слагаемого s_1 запишем следующее выражение:

$$\begin{aligned} s_1 = l_{\mathrm{Sh}}(U_1 \,|\, V_1) = \\ = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{1k}}{b_{1k}} \right\rceil = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{1k}}{\min(a_{0k} \cdot 2^{s_{10}}, a_{1k}, a_{3k})} \right\rceil = \\ = \left\lceil \log \max_{1 \leqslant k \leqslant q} \max\left(\frac{a_{1k}}{a_{0k} \cdot 2^{s_{10}}}, \frac{a_{1k}}{a_{1k}}, \frac{a_{1k}}{a_{3k}} \right) \right\rceil = \max(t_{10} - s_{10}, t_{13}, 0). \end{aligned}$$

Отсюда видно, что если $s_{10} < t_{10} - \max(t_{13}, 0)$, то при увеличении значения переменной s_{10} на 1 слагаемое s_1 уменьшится на 1, в противном случае $s_{10} \geqslant t_{10} - \max(t_{13}, 0)$, и при увеличении s_{10} на 1 слагаемое s_1 не изменится. Что касается слагаемого

$$s_3 = l_{\mathrm{Sh}}(U_3 \,|\, V_3) = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{3k}}{\max(b_{1k}, b_{2k})} \right\rceil,$$

то оно при увеличении значения переменной s_{10} на 1 либо не изменится, либо уменьшится на 1. Таким образом, при увеличении значения переменной s_{10} вплоть до $t_{10} - \max(t_{13}, 0)$ значение функции Σ не увеличивается, а при дальнейшем её увеличении — не уменьшается. Следовательно, формула

$$s'_{10} = t'_{10} - \max(t_{13}, 0)$$

будет доставлять оптимальное значение переменной s_{10} при любых фиксированных значениях остальных двух переменных. Отметим, что эта формула, строго говоря, представляет функцию, которая зависит от оптимального значения s'_0, так как через неё определяется величина t'_{10}. Также из полученных формул следует, что

$$s'_1 = \max(t_{13}, 0).$$

Далее, рассуждая аналогично, получаем формулы

$$s'_{20} = t_{20} - \max(t_{23}, 0), \quad s'_2 = \max(t_{23}, 0).$$

Подставляя полученное в функцию Σ, имеем

$$\begin{aligned}\Sigma' = s'_0 + (t'_{10} - \max(t_{13}, 0)) + (t'_{20} - \max(t_{23}, 0)) + \\ + \max(t_{13}, 0) + \max(t_{23}, 0) + s'_3 = s'_0 + t'_{10} + t'_{20} + s'_3 = \\ = \min_{s_0}(s_0 + t_{10}(s_0) + t_{20}(s_0) + s_3(s_0)). \quad (17)\end{aligned}$$

Определим оптимальное значение для переменной s_0. Положим $s_0 = 0$ и будем последовательно увеличивать её значение на единицу. Представим слагаемые суммы из (17) в более удобном для анализа виде:

$$\begin{aligned}t_{10} = \left\lceil \log \max \frac{a_{1k}}{a_{0k}} \right\rceil = \left\lceil \log \max_{1 \leqslant k \leqslant q} \max \left(\frac{a_{1k}}{2^{s_0}}, \frac{a_{1k}}{a_{1k}}, \frac{a_{1k}}{a_{2k}}, \frac{a_{1k}}{a_{3k}} \right) \right\rceil = \\ = \max(t_1 - s_0, t_{12}, t_{13}, 0),\end{aligned}$$

$$\begin{aligned}t_{20} = \left\lceil \log \max \frac{a_{2k}}{a_{0k}} \right\rceil = \left\lceil \log \max_{1 \leqslant k \leqslant q} \max \left(\frac{a_{2k}}{2^{s_0}}, \frac{a_{2k}}{a_{1k}}, \frac{a_{2k}}{a_{2k}}, \frac{a_{2k}}{a_{3k}} \right) \right\rceil = \\ = \max(t_2 - s_0, t_{21}, t_{23}, 0),\end{aligned}$$

$$\begin{aligned}s_3 = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{3k}}{\max(b_{1k}, b_{2k})} \right\rceil = \\ = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{3k}}{\max(\min(a_{0k} \cdot 2^{s_{10}}, a_{1k}, a_{3k}), \min(a_{0k} \cdot 2^{s_{20}}, a_{2k}, a_{3k}))} \right\rceil.\end{aligned}$$

Отсюда видно, что ключевыми для поиска оптимального значения являются величины $t_1 - \max(t_{12}, t_{13}, 0)$ и $t_2 - \max(t_{21}, t_{23}, 0)$. Обозначим их через r_1 и r_2 соответственно. Что произойдёт со слагаемыми при увеличении значения переменной s_0 на 1? Рассмотрим три случая. Если $s_0 < \min(r_1, r_2)$, то второе и третье слагаемое уменьшатся на 1. Если

$\min(r_1, r_2) \leqslant s_0 < \max(r_1, r_2)$, то одно из слагаемых t_{10} и t_{20} уменьшится на 1, а второе не изменится. Наконец, если $s_0 \geqslant \max(r_1, r_2)$, то оба этих слагаемых не изменятся. Первое слагаемое в любом случае, очевидно, увеличится на 1. Что касается последнего, четвёртого слагаемого, то нам достаточно заметить, что оно не может увеличиться и не может уменьшиться больше чем на 1. Отсюда следует, что значение

$$s'_0 = \max(r_1, r_2) = \\ = \max(t_1 - \max(t_{12}, t_{13}, 0), t_2 - \max(t_{21}, t_{23}, 0)) = \log X_{123}$$

будет оптимальным для переменной s_0. Подставляя его в формулы, получаем

$$t'_{10} = \max(t_1 - s'_0, t_{12}, t_{13}, 0) = \\ = \max(t_1 - \max(t_1 - \max(t_{12}, t_{13}, 0), t_2 - \max(t_{21}, t_{23}, 0)), t_{12}, t_{13}, 0) = \\ = \max(\min(\max(t_{12}, t_{13}, 0), t_1 - t_2 + \max(t_{21}, t_{23}, 0)), \max(t_{12}, t_{13}, 0)) = \\ = \max(t_{12}, t_{13}, 0),$$

аналогично

$$t'_{20} = \max(t_{21}, t_{23}, 0).$$

Отсюда

$$s'_0 + t'_{10} + t'_{20} = \max(t_1 + \max(t_{21}, t_{23}, 0), t_2 + \max(t_{12}, t_{13}, 0)). \tag{18}$$

Исходя из формул (16)–(18), получаем оценку

$$l_{\mathrm{Sh}}(S) \geqslant \max(t_1 + t_{21}, t_2 + t_{12}) + s'_3. \tag{19}$$

Теперь выразим величины b'_{1l} и b'_{2l}, $l = 1, \ldots, q$. Имеем

$$b'_{1l} = \min(a'_{0l} \cdot 2^{s'_{10}}, a_{1l}, a_{3l}) = \\ = \min(\min(2^{s'_0}, a_{1l}, a_{2l}, a_{3l}) \cdot 2^{s'_{10}}, a_{1l}, a_{3l}) = \\ = \min(2^{s'_0 + s'_{10}}, a_{1l} \cdot 2^{s'_{10}}, a_{2l} \cdot 2^{s'_{10}}, a_{3l} \cdot 2^{s'_{10}}, a_{1l}, a_{3l}) = \\ = \min(a_{1l}, a_{2l} \cdot 2^{s'_{10}}, a_{3l}, 2^{s'_0 + s'_{10}}) = \\ = \min(a_{1l}, a_{2l} \cdot 2^{t'_{10} - \max(t_{13}, 0)}, a_{3l}, 2^{s'_0 + t'_{10} - \max(t_{13}, 0)}) = \\ = \min(a_{1l}, a_{2l} \cdot 2^{\max(t_{12}, t_{13}, 0) - \max(t_{13}, 0)}, a_{3l}, 2^{s'_0 + \max(t_{12}, t_{13}, 0) - \max(t_{13}, 0)}) = \\ = \min(a_{1l}, a_{2l} \cdot Y_{123}, a_{3l}, X_{123} \cdot Y_{123}) = d^{12}_{3l}.$$

Равенство $b'_{2l} = d^{21}_{3l}$ доказывается аналогично. С учётом этого получаем

$$s'_3 = \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{3k}}{\max\left(d^{12}_{3k}, d^{21}_{3k}\right)} \right\rceil.$$

Наконец, подставляя последнее в (19), имеем

$$l_{\mathrm{Sh}}(S) \geqslant \max\left(t_1 + t_{21}, t_2 + t_{12}\right) + \left\lceil \log \max_{1 \leqslant k \leqslant q} \frac{a_{3k}}{\max\left(d_{3k}^{12}, d_{3k}^{21}\right)} \right\rceil.$$

Оценки, получаемые из этой перестановкой индексов, доказываются аналогично. Объединяя их все, приходим к неравенству

$$l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) \geqslant \max_{(i_1,i_2,i_3) \in S_3} \Bigg(\max\left(t_{i_1} + t_{i_2 i_1}, t_{i_2} + t_{i_1 i_2}\right) + \\ + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d_{i_3 l}^{i_1 i_2}, d_{i_3 l}^{i_2 i_1}\right)} \right\rceil \Bigg),$$

что и требовалось. Лемма 9 доказана.

Сформулируем и докажем основную теорему.

Теорема 1. *Пусть $A = (a_{ij})$ — матрица размера $3 \times q$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A) = \max_{(i_1,i_2,i_3) \in S_3} \left(t_{i_1} + t_{i_2 i_1} + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d_{i_3 l}^{i_1 i_2}, d_{i_3 l}^{i_2 i_1}\right)} \right\rceil \right) + \Delta(A),$$

где $\Delta(A)$ — некоторая функция от матрицы A, которая может принимать только целые неотрицательные значения не больше q.

Доказательство. Используя леммы 1, 4 и 8, получаем верхнюю оценку

$$l_{\mathrm{Sh}}(A) \leqslant l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) + l_{\mathrm{Sh}}(x_1 x_2 \cdots x_q) \leqslant \\ \leqslant \max_{(i_1,i_2,i_3) \in S_3} \left(t_{i_1} + t_{i_2 i_1} + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d_{i_3 l}^{i_1 i_2}, d_{i_3 l}^{i_2 i_1}\right)} \right\rceil \right) + q.$$

С помощью леммы 9 получаем нижнюю оценку

$$l_{\mathrm{Sh}}(A) \geqslant l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) \geqslant \max_{(i_1,i_2,i_3) \in S_3} \Bigg(\max\left(t_{i_1} + t_{i_2 i_1}, t_{i_2} + t_{i_1 i_2}\right) + \\ + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d_{i_3 l}^{i_1 i_2}, d_{i_3 l}^{i_2 i_1}\right)} \right\rceil \Bigg) \geqslant \max_{(i_1,i_2,i_3) \in S_3} \Bigg(t_{i_1} + t_{i_2 i_1} + \\ + \left\lceil \log \max_{1 \leqslant l \leqslant q} \frac{a_{i_3 l}}{\max\left(d_{i_3 l}^{i_1 i_2}, d_{i_3 l}^{i_2 i_1}\right)} \right\rceil \Bigg).$$

Теорема 1 доказана.

При доказательстве теоремы мы привели нижнюю оценку к виду верхней оценки. Можно действовать наоборот. В этом случае формула становится длиннее, но, во-первых, первое слагаемое будет равно сложности

реализации соответствующей подсистемы из двух мономов над мономом $x_1x_2\cdots x_q$, а во-вторых, выражение не будет изменяться при перестановке индексов i_1 и i_2, что позволяет брать максимум по трём величинам, а не по шести. Сформулируем и докажем соответствующий результат.

Утверждение 1. *Пусть $A=(a_{ij})$ — матрица размера $p\times 3$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A)=\max_{\substack{(i_1,i_2,i_3)\in S_3,\\ i_1<i_2}}\Bigg(\max\left(t_{i_1}+t_{i_2i_1},t_{i_2}+t_{i_1i_2}\right)+\\ +\left\lceil\log\max_{1\leqslant l\leqslant q}\frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2},d_{i_3l}^{i_2i_1}\right)}\right\rceil\Bigg)+\Delta(A),$$

где $\Delta(A)$ — некоторая функция от матрицы A, которая может принимать только целые неотрицательные значения не больше q.

Доказательство. Нижняя оценка сразу следует из леммы 9. Используя леммы 1, 4 и 8, получаем верхнюю оценку

$$\begin{aligned}l_{\mathrm{Sh}}(A)&\leqslant l_{\mathrm{Sh}}(A\,|\,x_1x_2\cdots x_q)+l_{\mathrm{Sh}}(x_1x_2\cdots x_q)\leqslant\\ &\leqslant\max_{(i_1,i_2,i_3)\in S_3}\left(t_{i_1}+t_{i_2i_1}+\left\lceil\log\max_{1\leqslant l\leqslant q}\frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2},d_{i_3l}^{i_2i_1}\right)}\right\rceil\right)+q\leqslant\\ &\leqslant\max_{\substack{(i_1,i_2,i_3)\in S_3,\\ i_1<i_2}}\Bigg(\max\left(t_{i_1}+t_{i_2i_1},t_{i_2}+t_{i_1i_2}\right)+\\ &\qquad+\left\lceil\log\max_{1\leqslant l\leqslant q}\frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2},d_{i_3l}^{i_2i_1}\right)}\right\rceil\Bigg)+q.\end{aligned}$$

Утверждение 1 доказано.

Отметим, что для величины $l_{\mathrm{Sh}}(A)$ мы использовали довольно грубую нижнюю оценку — фактически такую же, какая была получена в лемме 9 для величины $l_{\mathrm{Sh}}(A\,|\,x_1x_2\cdots x_q)$. Из-за этого «зазор» между верхней и нижней оценками увеличился. Если применять леммы 8 и 9 для оценки величины $l_{\mathrm{Sh}}(A\,|\,x_1x_2\cdots x_q)$, то получится следующий результат.

Утверждение 2. *Пусть $A=(a_{ij})$ — матрица размера $3\times q$ из натуральных чисел. Тогда*

$$\begin{aligned}&l_{\mathrm{Sh}}(A\,|\,x_1x_2\cdots x_q)=\\ &\quad=\max_{(i_1,i_2,i_3)\in S_3}\left(t_{i_1}+t_{i_2i_1}+\left\lceil\log\max_{1\leqslant l\leqslant q}\frac{a_{i_3l}}{\max\left(d_{i_3l}^{i_1i_2},d_{i_3l}^{i_2i_1}\right)}\right\rceil\right)+\delta(A),\end{aligned}$$

где $\delta(A)$ — некоторая функция от матрицы A, которая может принимать только значения 0 *и* 1.

Возможно ли сохранить такой же «зазор» при переходе к величине $l_{\mathrm{Sh}}(A)$, т. е. заменить $\Delta(A)$ на $\delta(A)$ в формулировке теоремы 1? Можно обобщить этот вопрос так: является ли реализация монома, состоящего из всех переменных, оптимальным первым шагом для реализации матрицы без нулей? Сформулируем гипотезу, соответствующую положительному ответу на этот вопрос.

Гипотеза 1. *Пусть $A = (a_{ij})$ — матрица размера $p \times q$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A) = l_{\mathrm{Sh}}(A \,|\, x_1 x_2 \cdots x_q) + q - 1.$$

Известно, что эта гипотеза верна для матриц размера $2 \times q$ (см. [6, утверждение 4]) и $p \times 2$ (см. [8, лемма 6]). Докажем, что она верна и для матриц размера $p \times 3$.

Лемма 10. *Пусть $A = (a_{ij})$ — матрица размера $p \times 3$ из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A) = l_{\mathrm{Sh}}(A \,|\, x_1 x_2 x_3) + 2.$$

Доказательство. Верхняя оценка сразу следует из леммы 1. Докажем нижнюю оценку.

Пусть M — система мономов, соответствующая матрице A, а S — минимальная схема, реализующая эту систему мономов. Рассмотрим множество мономов из схемы S, содержащих хотя бы две переменные. Для каждого $i = 1, 2, 3$ выберем из этого множества моном V_i, в котором степень при переменной x_i минимальна (если таких мономов несколько, то выберем первый из них) и обозначим эту степень через a_i. Положим $V_0 = x_1^{a_1} x_2^{a_2} x_3^{a_3}$ и заметим, что для любого монома U из системы M выполнено условие $V_0 \leqslant U$.

Выделим из схемы S подпоследовательность $S_0 = \{U \in S \mid U \leqslant V_0\}$. Легко видеть, что она образует схему, реализующую систему мономов $\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}$.

Заменим каждый моном U в схеме S мономом $U' = U_{\max}(U, V_0)$. Полученная при такой замене последовательность мономов S' образует схему, реализующую систему мономов M над мономом V_0. Действительно, легко видеть, что для любого монома U из системы M выполнено условие $U' = U$, откуда получаем $U \in S'$. Кроме того, из условий $U_1 \leqslant U$, $U_2 \leqslant U$ и $U \leqslant U_1 U_2$ следует выполнение условий $U_1' \leqslant U'$, $U_2' \leqslant U'$ и $U' \leqslant U_1' U_2'$, поэтому из леммы 2 вытекает, что для последовательности мономов S' найдётся набор правил вычисления. Отметим, что при переходе от схемы S к схеме S' число мономов не изменилось, но число существенных мономов увеличилось на два, поэтому $l_{\mathrm{Sh}}(S') = l_{\mathrm{Sh}}(S) + 2$.

Удалим из схемы S' все повторяющиеся мономы, оставив только первый экземпляр каждого из них. Получим схему S'', которая также реализует систему мономов M над мономом V_0.

Выделим из схемы S подпоследовательность мономов $S_0 = \{U \in S \mid U \leqslant V_0\}$. Заметим, что эта последовательность образует схему, реализующую систему мономов $\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}$, причём если $U \in S_0$, то $U' = V_0$, поэтому схема S' содержит хотя бы $|S_0|$ мономов V_0 (здесь и далее $|S_0|$ обозначает число мономов в схеме S_0). Так как $|S_0| = l_{\mathrm{Sh}}(S_0) + 3$, число мономов V_0, удалённых при переходе от схемы S' к схеме S'', будет не меньше $l_{\mathrm{Sh}}(S_0) + 2$.

Чтобы оценить величину $l_{\mathrm{Sh}}(S'')$, рассмотрим три случая.

Случай 1. Если схема S_0 содержит моном от трёх переменных или два монома от двух переменных, то

$$l_{\mathrm{Sh}}(S_0) \geqslant l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) + 2.$$

Как было показано выше, число мономов V_0, удалённых при переходе от схемы S' к схеме S'', будет не меньше $l_{\mathrm{Sh}}(S_0) + 2$, поэтому

$$l_{\mathrm{Sh}}(S'') \leqslant l_{\mathrm{Sh}}(S') - (l_{\mathrm{Sh}}(S_0) + 2) \leqslant l_{\mathrm{Sh}}(S') - l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) - 4.$$

Случай 2. Если схема S_0 содержит ровно один моном от двух переменных, то

$$l_{\mathrm{Sh}}(S_0) \geqslant l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) + 1.$$

Можно без ограничения общности считать, что это моном $x_1^{a_1}x_2^{a_2}$. Тогда моном V_3 не может содержаться в мономе V_0. Пусть его правило вычисления в схеме S имеет вид $V_3 = \big(x_3^{a_3}; W_3\big)_{R_3}$. Значит, схема S содержит мономы V_3 и W_3, которые не содержатся в схеме S_0, а при переходе к схеме S' будут заменены одним и тем же мономом $V_3' = W_3'$. Следовательно, при переходе к схеме S'' будут удалены не только мономы V_0, получающиеся из схемы S', но и хотя бы один моном V_3'. Отсюда

$$l_{\mathrm{Sh}}(S'') \leqslant l_{\mathrm{Sh}}(S') - (l_{\mathrm{Sh}}(S_0) + 2) - 1 \leqslant l_{\mathrm{Sh}}(S') - l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) - 4.$$

Случай 3. Если схема S_0 не содержит мономов от двух переменных, то будем использовать тривиальную оценку

$$l_{\mathrm{Sh}}(S_0) \geqslant l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big).$$

В этом случае все три монома V_1, V_2, V_3 различны и не содержатся в мономе V_0. Для каждого $i = 1, 2, 3$ проведём следующее рассуждение (аналогичное рассуждению из предыдущего случая). Пусть правило вычисления монома V_i в схеме S имеет вид $V_i = \big(x_3^{a_3}; W_i\big)_{R_i}$. Значит, схема S содержит мономы V_i и W_i, которые не содержатся в схеме S_0, а при переходе к схеме S' будут заменены одним и тем же мономом $V_i' = W_i'$. Следовательно, при переходе к схеме S'' будут удалены не только мономы V_0,

получающиеся из схемы S', но и хотя бы по одному моному V_1', V_2', V_3'. Отсюда

$$l_{\mathrm{Sh}}(S'') \leqslant l_{\mathrm{Sh}}(S') - (l_{\mathrm{Sh}}(S_0) + 2) - 3 \leqslant l_{\mathrm{Sh}}(S') - l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) - 5.$$

Итак, в любом случае справедливо неравенство

$$l_{\mathrm{Sh}}(S'') \leqslant l_{\mathrm{Sh}}(S') - l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) - 4,$$

откуда, используя лемму 1 и свойства рассматриваемых схем, получаем цепочку неравенств

$$\begin{aligned} l_{\mathrm{Sh}}(A) = l_{\mathrm{Sh}}(M) &\leqslant l_{\mathrm{Sh}}(V_0) + l_{\mathrm{Sh}}(M \,|\, V_0) \leqslant \\ &\leqslant \big(l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) + 2\big) + l_{\mathrm{Sh}}(S'') \leqslant \\ &\leqslant \big(l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) + 2\big) + \big(l_{\mathrm{Sh}}(S') - l_{\mathrm{Sh}}\big(\{x_1^{a_1}, x_2^{a_2}, x_3^{a_3}\}\big) - 4\big) = \\ &= l_{\mathrm{Sh}}(S') - 2 = l_{\mathrm{Sh}}(S) = l_{\mathrm{Sh}}(A). \end{aligned}$$

Отметим, что при использовании оценки из случая 3 правая часть станет меньше на единицу, что приведёт к противоречию. Значит, схема S_0 обязана содержать хотя бы один моном от двух переменных. Из полученной цепочки неравенств следует равенство

$$l_{\mathrm{Sh}}(A) = l_{\mathrm{Sh}}(V_0) + l_{\mathrm{Sh}}(M \,|\, V_0).$$

Наконец, с помощью лемм 1, 3 и 4 получаем

$$\begin{aligned} l_{\mathrm{Sh}}(A) &= l_{\mathrm{Sh}}(V_0) + l_{\mathrm{Sh}}(M \,|\, V_0) = \\ &= l_{\mathrm{Sh}}(V_0 \,|\, x_1x_2x_3) + 2 + l_{\mathrm{Sh}}(A \,|\, V_0) \geqslant l_{\mathrm{Sh}}(A \,|\, x_1x_2x_3) + 2, \end{aligned}$$

что и требовалось. Лемма 10 доказана.

Используя утверждение 2 и лемму 10, приходим к следующему результату.

Утверждение 3. *Пусть $A = (a_{ij})$ — матрица размера 3×3 из натуральных чисел. Тогда*

$$l_{\mathrm{Sh}}(A) = \max_{(i_1,i_2,i_3)\in S_3} \left(t_{i_1} + t_{i_2i_1} + \left\lceil \log \max_{1\leqslant l\leqslant q} \frac{a_{i_3l}}{\max\big(d_{i_3l}^{i_1i_2}, d_{i_3l}^{i_2i_1}\big)} \right\rceil\right) + 2 + \delta(A),$$

где $\delta(A)$ — некоторая функция от матрицы A, которая может принимать только значения 0 *и* 1.

Автор посвящает статью памяти Юрия Владимировича Мерекина (1935–2025). В работе [2] Ю. В. Мерекина впервые введено понятие схемы композиции, а полученный им результат о сложности реализации одного монома [2, теорема 2] лёг в основу исследований соответствующей вычислительной модели.

Финансирование работы

Исследование выполнено при финансовой поддержке Министерства науки и высшего образования России в рамках программы Московского центра фундаментальной и прикладной математики (соглашение № 075–15–2025–345). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Ширшов А. И.** Некоторые алгоритмические проблемы для алгебр Ли // Сиб. мат. журн. 1962. Т. 3, № 2. С. 292–296.
2. **Мерекин Ю. В.** О порождении слов с использованием операции композиции // Дискрет. анализ и исслед. операций. Сер. 1. 2003. Т. 10, № 4. С. 70–78.
3. **Merekin Yu. V.** Some bounds on the complexity of words // Southeast Asian Bull. Math. 2006. V. 30, No. 6. P. 1081–1121.
4. **Трусевич Е. Н.** О сложности реализации схемами композиции систем из двух мономов от двух переменных // Мат. VIII Молодёж. науч. шк. по дискретной математике и её приложениям. Ч. 2 (Москва, Россия, 24–29 окт. 2011 г.). М.: ИПМ РАН, 2011. С. 40–44.
5. **Трусевич Е. Н.** О сложности вычисления некоторых систем одночленов схемами композиции // Вестн. Моск. ун-та. Сер. 1. 2014. № 5. С. 18–22.
6. **Корнеев С. А.** О сложности реализации системы из двух мономов схемами композиции // Дискрет. математика. 2020. Т. 32, № 2. С. 15–31.
7. **Корнеев С. А.** Об асимптотическом поведении функций шенноновского типа, характеризующих сложность вычисления систем мономов // Учён. зап. Казан. ун-та. Сер. Физ.-мат. науки. 2020. Т. 162, № 3. С. 300–310.
8. **Корнеев С. А.** О сложности реализации системы мономов от двух переменных схемами композиции // Прикл. дискрет. математика. 2021. № 53. С. 103–119.
9. **Лупанов О. Б.** Асимптотические оценки сложности управляющих систем. М.: Изд-во Моск. ун-та, 2024. 124 с.
10. **Сэвидж Дж. Э.** Сложность вычислений. М.: Факториал, 1998. 368 с.
11. **Храпченко В. М.** Нижние оценки сложности схем из функциональных элементов // Кибернетический сборник. Новая сер. Вып. 21. М.: Мир, 1984. С. 3–54.
12. **Кочергин В. В.** Задачи Р. Беллмана и Д. Кнута и их обобщения: Сложность аддитивных вычислений. Saarbrücken: Palmarium Acad. Publ., 2012. 396 с.
13. **Кочергин В. В.** О задачах Беллмана и Кнута и их обобщениях // Фундамент. и прикл. математика. 2015. Т. 20, № 6. С. 159–188.

14. Кочергин В. В. Задачи Беллмана, Кнута, Лупанова, Пиппенджера и их вариации как обобщения задачи об аддитивных цепочках // Математические вопросы кибернетики. Вып. 20. М.: Физматлит, 2022. С. 119–256.

15. Кочергин В. В. О сложности вычисления систем одночленов от двух переменных // Тр. VII Междунар. конф. «Дискретные модели в теории управляющих систем» (Покровское, Россия, 4–6 мар. 2006 г.). М.: МАКС Пресс, 2006. С. 185–190.

16. Сидоренко А. Ф. Сложность аддитивных вычислений семейства целочисленных линейных форм // Зап. науч. сем. ЛОМИ. Т. 105. Теоретические применения методов математической логики. III. Л.: Наука, 1981. С. 53–61.

17. Knuth D. E., Papadimitriou C. H. Duality in addition chains // Bull. Eur. Assoc. Theor. Comput. Sci. 1981. V. 13. P. 2–4.

18. Olivos J. On vectorial addition chains // J. Algorithms. 1981. V. 2, No. 1. P. 13–21.

19. Кочергин В. В. О сложности совместного вычисления трёх одночленов от трёх переменных // Математические вопросы кибернетики. Вып. 15. М.: Физматлит, 2006. С. 79–154.

20. Кочергин В. В. Простое доказательство верхней оценки сложности вычисления трёх одночленов трёх переменных // Вестн. Моск. ун-та. Сер. 1. 2019. № 2. С. 3–8.

21. Кочергин В. В. Об одном соотношении двух мер сложности вычисления систем одночленов // Вестн. Моск. ун-та. Сер. 1. 2009. № 4. С. 8–13.

Корнеев Сергей Александрович

Статья поступила
18 июня 2025 г.

После доработки —
20 августа 2025 г.

Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 146–171

UDC 519.71 DOI: 10.33048/daio.2025.32.838

ON THE COMPLEXITY OF IMPLEMENTATION OF A SYSTEM OF THREE MONOMIALS BY COMPOSITION CIRCUITS

S. A. Korneev[1, 2]

[1] Lomonosov Moscow State University,
1 Leninskie Gory, 119991, Moscow, Russia

[2] Moscow Center of Fundamental and Applied Mathematics
1 Leninskie Gory, 119991, Moscow, Russia

E-mail: korneev.sa.42@gmail.com

Abstract. We study circuit complexity of monomial system computation. In the considered computational model, the complexity means the minimal number of composition operations sufficient to compute the monomial system. Multiple use of results in intermediate calculations is allowed. We consider a three-monomial system without zeros. The main results of this paper are as follows. A formula for asymptotic growth of circuit complexity of system computation is established. In case of three variables, a more accurate formula is obtained, which expresses circuit complexity of system computation with precision of one. Tab. 1, bibliogr. 21.

Keywords: composition circuit, circuit of functional elements, set of monomials, computational complexity, circuit complexity.

References

1. **A. I. Shirshov,** Some algorithmic problems for Lie algebras, *Sib. Mat. Zh.* **3** (2), 292–296 (1962) [Russian].
2. **Yu. V. Merekin,** On the generation of words using the composition operation, *Diskretn. Anal. Issled. Oper., Ser. 1*, **10** (4), 70–78 (2003) [Russian].
3. **Yu. V. Merekin,** Some bounds on the complexity of words, *Southeast Asian Bull. Math.* **30** (6), 1081–1121 (2006).
4. **E. N. Trusevich,** On the complexity of implementation of a system of two monomials in two variables by composition circuits, in *Proc. VIII Youth Sci. Sch. Discrete Mathematics and Its Applications*, Pt. 2 (Moscow, Russia, Oct. 24–29, 2011) (IPM RAN, Moscow, 2011), pp. 40–44 [Russian].

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

5. **E. N. Trusevich,** Complexity of certain systems of monomials in calculation by composition circuits, *Vestn. Mosk. Univ., Ser. 1*, No. 5, 18–22 (2014) [Russian] [*Mosc. Univ. Math. Bull.* **69** (5), 193–197 (2014), DOI: 10.3103/S0027132214050039].
6. **S. A. Korneev,** On the complexity of implementation of a system of two monomials by composition circuits, *Diskretn. Mat.* **32** (2), 15–31 (2020) [Russian] [*Discrete Math. Appl.* **31** (2), 113–125 (2021), DOI: 10.1515/dma-2021-0010].
7. **S. A. Korneev,** On the asymptotic behavior of Shannon-type functions characterizing the computing complexity of systems of monomials, *Uchyon. Zap. Kazan. Univ., Ser. Fiz.-Mat. Nauki* **162** (3) 300–310 (2020) [Russian].
8. **S. A. Korneev,** The complexity of implementation of a system of monomials in two variables by composition circuits, *Prikl. Diskretn. Mat.*, No. 53, 103–119 (2021) [Russian].
9. **O. B. Lupanov,** *Asymptotic Estimates for the Complexity of Control Systems* (Izd. Mosk. Univ., Moscow, 2024) [Russian].
10. **J. E. Savage,** *The Complexity of Computing* (Wiley, New York, 1976; Faktorial, Moscow, 1998 [Russian]).
11. **V. M. Khrapchenko,** Lower Bounds for the Complexity of Circuits of Functional Elements, in *Cybernetics Collection, New Series*, Vol. 21 (Mir, Moscow, 1984), pp. 3–54.
12. **V. V. Kochergin,** *The R. Bellman's and D. Knuth's Problems and Their Generalizations: The Complexity of Additive Computing* (Palmarium Acad. Publ., Saarbrücken, 2012).
13. **V. V. Kochergin,** On Bellman's and Knuth's problems and their generalizations, *Fundam. Prikl. Mat.* **20** (6), 159–188 (2015) [Russian] [*J. Math. Sci.* **233** (1), 103–124 (2018), DOI: 10.1007/s10958-018-3928-4].
14. **V. V. Kochergin,** The Bellman, Knuth, Lupanov, Pippenger problems and their variations as generalizations of the additive circuits problem, in *Mathematical Problems of Cybernetics*, Vol. 20 (Fizmatlit, Moscow, 2022), pp. 119–256 [Russian].
15. **V. V. Kochergin,** On the computation complexity of systems of monomials in two variables, in *Proc. VII Int. Conf. "Discrete Models in Control Systems Theory"* (Pokrovskoe, Russia, Mar. 4–6, 2006) (MAKS Press, Moscow, 2006), pp. 185–190 [Russian].
16. **A. F. Sidorenko,** Complexity of additive computations of systems of linear forms, in *Zap. Nauchn. Sem. LOMI*, Vol. 105. Theoretical Applications of Methods of Mathematical Logics, Pt. III (Nauka, Leningrad, 1981), pp. 53–61 [Russian] [*J. Sov. Math.* **22** (3), 1310–1315 (1983), DOI: 10.1007/BF01084394].
17. **D. E. Knuth** and **C. H. Papadimitriou,** Duality in addition chains, *Bull. Eur. Assoc. Theor. Comput. Sci.* **13**, 2–4 (1981).
18. **J. Olivos,** On vectorial addition chains, *J. Algorithms* **2** (1), 13–21 (1981).
19. **V. V. Kochergin,** On the complexity of joint computing of tree monomials it tree variables, in *Mathematical Problems of Cybernetics*, Vol. 15 (Fizmatlit, Moscow, 2006), pp. 79–154 [Russian].

20. V. V. Kochergin, A simple proof for the upper bound of the computational complexity of three monomials in three variables, *Vestn. Mosk. Univ., Ser. 1*, No. 2, 3–8 (2019) [Russian] [*Mosc. Univ. Math. Bull.* **74** (2), 43–48 (2019), DOI: 10.3103/S0027132219020013].

21. V. V. Kochergin, Relation between two measures of the computation complexity for systems of monomials, *Vestn. Mosk. Univ., Ser. 1*, No. 4, 8–13 (2009) [Russian] [*Mosc. Univ. Math. Bull.* **64** (4), 144–149 (2009), DOI: 10.3103/S0027132209040020].

Sergey A. Korneev

Received June 18, 2025
Revised August 20, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 172–190

УДК 519.8 DOI: 10.33048/daio.2025.32.834

О СЛОЖНОСТИ ДВУХ ЗАДАЧ ПОИСКА КЛАСТЕРОВ С БОЛЬШОЙ МОЩНОСТЬЮ

С. М. Нещадим[1, a], *В. И. Хандеев*[2, b]

[1] Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
[2] Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: [a] s.neshchadim@g.nsu.ru, [b] khandeev@math.nsc.ru

Аннотация. Для конечного множества точек евклидова пространства рассматриваются задачи поиска его непересекающихся подмножеств. В одной из задач мощность каждого из подмножеств должна быть не меньше заданного числа. В другой задаче мощность всех подмножеств одинакова, а их объединение должно совпадать с исходным множеством. В обеих задачах дополнительно требуется, чтобы для каждого подмножества сумма квадратов расстояний до центроида не превосходила заданной величины. Доказано, что задачи NP-полны в сильном смысле в случае, когда число кластеров равно двум, а размерность пространства является частью входа задачи. Кроме того, доказано, что задачи NP-полны в одномерном случае для любого фиксированного числа кластеров. Ил. 2, библиогр. 16.

Ключевые слова: кластеризация, наименьший размер кластера, ограниченный разброс, евклидово пространство, NP-полнота.

Введение

Предметом исследования настоящей статьи является задача поиска семейства непересекающихся подмножеств в конечном множестве точек евклидова пространства при ограничении снизу на мощность каждого подмножества и ограничении сверху на его разброс. Также рассматривается частный случай, в котором объединение искомого семейства совпадает со всем входным множеством. Целью исследования является установление сложностного статуса ранее не изученных подслучаев этих задач.

Одной из первых и наиболее известных задач кластеризации является задача k-means (k-средних), также именуемая MSSC (Minimum Sum of Squares Clustering) [1, 2]. Суть задачи состоит в том, чтобы разбить

множество точек в евклидовом пространстве на k кластеров таким образом, чтобы минимизировать сумму квадратов расстояний между каждой точкой и центром соответствующего ей кластера. Центром выступает центроид, т. е. среднее арифметическое точек внутри кластера. Для приближённого решения этой задачи часто применяют эвристический алгоритм k-means, который находит широкое применение в различных областях: от здравоохранения [3] до сегментации изображений [4] и анализа сетевого трафика [5]. При этом важно отметить, что для разбиения на два кластера задача NP-трудна [6], тогда как для одномерного случая существует алгоритм с полиномиальной сложностью [7]. Задача также разрешима за полиномиальное время при $k = 1$, поскольку в этом случае оптимальное решение представляет собой исходное множество точек целиком.

В работе рассматривается следующая похожая задача кластеризации.

Задача 1. *Даны множество* $\mathcal{Y} = \{y_1, \ldots, y_n\}$ *векторов из* $\mathbb{R}^d$ *и числа* $A \in \mathbb{R}_+$, $M \in \mathbb{N}$. *Существуют ли* k *попарно непересекающихся множеств* $\mathcal{C}_i \subset \mathcal{Y}$, $i = 1, \ldots, k$, *таких, что* $|\mathcal{C}_i| \geqslant M$ *и* $F(\mathcal{C}_i) \leqslant A$ *при* $i = 1, \ldots, k$, *где*

$$F(\mathcal{C}_i) = \sum_{y \in \mathcal{C}_i} \|y - \overline{y}(\mathcal{C}_i)\|^2,$$

$$\overline{y}(\mathcal{C}_i) = \frac{1}{|\mathcal{C}_i|} \sum_{y \in \mathcal{C}_i} y, \quad i = 1, \ldots, k?$$

Заметим, что в отличие от задачи MSSC задача 1 в оптимизационной форме может быть записана как задача поиска семейства непересекающихся кластеров с максимальной мощностью минимального (по числу элементов) кластера и ограничением на разброс каждого кластера. При этом поиск кластеров большего размера, удовлетворяющих определённым ограничениям, применяется в таких областях, как, например, задачи анализа социальных сетей [8], биоинформатика [9] и т. д. Если же в задаче 1 дополнительно потребовать, чтобы все подмножества имели одинаковую мощность и в объединении совпадали со всем множеством $\mathcal{Y}$, то получим следующую задачу.

Задача 2. *Даны множество* $\mathcal{Y} = \{y_1, \ldots, y_n\}$ *векторов из* $\mathbb{R}^d$, *где* $n = kM$, *и число* $A \in \mathbb{R}_+$. *Существует ли разбиение множества* $\mathcal{Y}$ *на* k *множеств* $\mathcal{C}_i$, $i = 1, \ldots, k$, *такое, что* $|\mathcal{C}_i| = M$ *и* $F(\mathcal{C}_i) \leqslant A$?

Задача 2 является частным случаем задачи 1, а значит, задача 1 NP-полна во всех случаях, когда NP-полна задача 2.

В [10] рассмотрена задача VS-2 (в оптимизационной формулировке также известная как M-Variance [11]), которая представляет собой частный случай задачи 1 при $k = 1$. В [10] показано, что такая задача NP-

полна в сильном смысле, если размерность является частью входа задачи. При этом в одномерном случае ($d = 1$) при $k = 1$ задача M-Variance полиномиально разрешима [12].

Как и задача MSSC, задача 2 при $k = 1$ полиномиально разрешима: достаточно проверить разброс единственного возможного решения, а именно всего входного множества $\mathcal{Y}$.

В [13] построено полиномиальное сведение задачи о сбалансированной k-раскраске однородного графа к задаче 1. Тем самым из того, что задача о сбалансированной k-раскраске NP-полна при $k \geqslant 3$, следует NP-полнота задачи 1 в случае, когда размерность пространства является частью входа задачи, а число кластеров k фиксировано и больше либо равно трём. Более того, поскольку в сведении, представленном в [13], векторы в задаче 1 содержат только числа 0, 1 и -1, все кластеры имеют одинаковую мощность, а их объединение совпадает со всем входным множеством точек $\mathcal{Y}$, из этого сведения следует более строгое утверждение — NP-полнота в сильном смысле задачи 2 (как и задачи 1) при тех же условиях.

Таким образом, оставался открытым вопрос о сложностном статусе задач 1 и 2 в случае фиксированного $k = 2$. При этом похожие задачи, в которых в формуле разброса кластера используется не геометрический центр, а фиксированная точка евклидова пространства либо произвольная точка входного множества, NP-полны даже в одномерном случае [14].

В данной работе докажем, что задача 2 (а следовательно, и задача 1) NP-полна в сильном смысле в случае, когда $k = 2$ и размерность пространства является частью входа задачи. Также докажем, что обе эти задачи NP-полны в обычном смысле в случае одномерного пространства при произвольном фиксированном $k \geqslant 2$.

1. Вспомогательные задачи

В работе будет использоваться NP-полная [15] задача Максимальный разрез.

Задача Максимальный разрез. *Даны граф $G = (V, E)$ и целое $L_1 > 0$. Верно ли, что существует разбиение множества V на два таких непересекающихся множества V_1 и V_2, что число рёбер, соединяющих вершины из множеств V_1 и V_2, не меньше L_1?*

Кроме того, нам понадобится следующая вспомогательная задача.

Задача Два разреженных подграфа. *Даны граф $G = (V, E)$ и целое $L_2 > 0$. Верно ли, что существует разбиение множества V на два непересекающихся множества V_1 и V_2 таких, что $|V_1| = |V_2|$, а число*

рёбер для каждого $i = 1, 2$, *соединяющих вершины из* V_i *между собой, не превосходит* L_2?

Покажем, что данная задача NP-полна.

Теорема 1. *Задача* Два разреженных подграфа NP-*полна.*

Доказательство. Построим полиномиальное сведение задачи Максимальный разрез к задаче Два разреженных подграфа. Рассмотрим произвольный пример задачи Максимальный разрез — граф $G = (V, E)$ и число $L_1 \in \mathbb{Z}_+$. По этому примеру можно за полиномиальное время построить следующий пример задачи Два разреженных подграфа — граф $G^* = (V^*, E^*)$, состоящий из графа G и его копии, которую будем обозначать $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, и число $L_2 = |E| - L_1$. Покажем, что исходный и построенный примеры одновременно либо имеют решения, либо нет.

Пусть для примера задачи Максимальный разрез существует решение — разбиение множества V вершин графа G на непересекающиеся подмножества V_1 и V_2 такие, что мощность множества

$$E_{12} = \{e \in E \mid \exists\, v_1 \in V_1,\, v_2 \in V_2 \colon e = (v_1, v_2)\}$$

рёбер, соединяющих вершины из V_1 и V_2, не меньше L_1. Рассмотрим следующую пару множеств вершин графа G^*:

$$V_1^* = V_1 \cup \widetilde{V}_2, \quad V_2^* = V_2 \cup \widetilde{V}_1,$$

где $\widetilde{V}_1$ и $\widetilde{V}_2$ — копии множеств V_1 и V_2, образующие вместе множество вершин графа $\widetilde{G}$. На рис. 1 показан пример с изображением графа G, его копии $\widetilde{G}$ и множеств V_1^*, V_2^*.

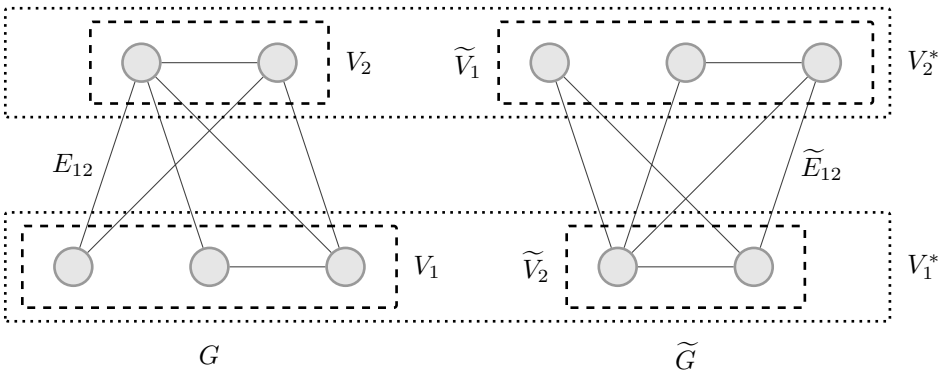


Рис. 1. Пример задачи Два разреженных подграфа

Покажем, что множества V_1^* и V_2^* являются решением построенного примера задачи Два разреженных подграфа. Множества V_i^* не пересекаются по построению; равенство мощностей $|V_1^*|$ и $|V_2^*|$ следует из того, что

$$|V_1| = |\widetilde{V}_1|, \quad |V_2| = |\widetilde{V}_2|.$$

Для каждого $i = 1, 2$ оценим число рёбер графа G^*, соединяющих вершины из V_i^* между собой. Обозначим множество всех таких рёбер через E_i^*. Множество E_1^* состоит из рёбер, соединяющих вершины из V_1, а также из рёбер, соединяющих вершины из $\widetilde{V}_2$. В свою очередь, множество E_2^* состоит из рёбер, соединяющих вершины из $\widetilde{V}_1$, а также из рёбер, соединяющих вершины из V_2. Следовательно,

$$|E_1^*| = |E_2^*| = |E_1| + |E_2|,$$

где E_i — множество рёбер графа G, соединяющих вершины из множества V_i между собой, $i = 1, 2$. Поскольку по предположению мощность множества E_{12} не меньше L_1, имеем

$$|E_1^*| = |E_2^*| = |E| - |E_{12}| \leqslant |E| - L_1 = L_2.$$

Таким образом, множества V_1^* и V_2^* являются решением построенного примера задачи Два разреженных подграфа.

Теперь предположим, что в задаче Два разреженных подграфа существует решение — требуемые множества V_1^* и V_2^* вершин. Пусть

$$V_1 = V_1^* \cap V, \quad V_2 = V_2^* \cap V, \quad \widetilde{V}_1 = V_2^* \cap \widetilde{V}, \quad \widetilde{V}_2 = V_1^* \cap \widetilde{V}.$$

Другими словами, разбиение вершин графа G^* на V_1^* и V_2^* порождает разбиение вершин графа G на V_1 и V_2, а вершин графа $\widetilde{G}$ — на $\widetilde{V}_2$ и $\widetilde{V}_1$. Множества E_1, E_2, E_{12} ($\widetilde{E}_1$, $\widetilde{E}_2$, $\widetilde{E}_{12}$) рёбер определим так же, как в первой части доказательства, используя множества вершин V_1, V_2 ($\widetilde{V}_1$, $\widetilde{V}_2$). Тогда по предположению $|E_1| + |\widetilde{E}_2| \leqslant L_2$, $|E_2| + |\widetilde{E}_1| \leqslant L_2$. Покажем, что как минимум одна из пар — V_1, V_2 или $\widetilde{V}_1, \widetilde{V}_2$ — является решением исходного примера задачи Максимальный разрез. Для этого оценим сумму мощностей множеств E_{12} и $\widetilde{E}_{12}$:

$$\begin{aligned} |E_{12}| + |\widetilde{E}_{12}| &= (|E| - |E_1| - |E_2|) + (|\widetilde{E}| - |\widetilde{E}_1| - |\widetilde{E}_2|) = \\ &= (|E| - |E_1| - |\widetilde{E}_2|) + (|E| - |E_2| - |\widetilde{E}_1|) \geqslant 2(|E| - L_2) = 2L_1. \end{aligned}$$

Из этой оценки следует, что хотя бы одна из мощностей $|E_{12}|$, $|\widetilde{E}_{12}|$ больше либо равна L_1, а значит, соответствующая пара множеств вершин (V_1, V_2 или $\widetilde{V}_1, \widetilde{V}_2$) будет решением исходной задачи Максимальный разрез.

Таким образом, произвольный пример задачи Максимальный разрез и построенный за полиномиальное время пример задачи Два разреженных подграфа одновременно либо имеют решения, либо не имеют, из чего следует, что задача Два разреженных подграфа NP-полна. Теорема 1 доказана.

Заметим, что задача Два разреженных подграфа эквивалентна следующей задаче на дополнении исходного графа.

Задача Два плотных подграфа. *Даны граф* $G = (V, E)$ *и целое* $L_3 > 0$. *Верно ли, что существует разбиение множества* V *на два непересекающихся множества* V_1 *и* V_2 *таких, что* $|V_1| = |V_2|$, *а число рёбер для каждого* $i = 1, 2$, *соединяющих вершины из* V_i *между собой, не меньше* L_3?

Действительно, поскольку в задаче Два разреженных подграфа $|V_1| = |V_2| = |V|/2$, существование разбиения на два равных по мощности подмножества вершин не более чем с L_2 рёбрами между элементами каждого из них эквивалентно существованию в дополнении графа разбиения на два равных по мощности подмножества вершин не менее чем с $\frac{1}{2} \cdot \frac{|V|}{2} \cdot \left(\frac{|V|}{2} - 1\right) - L_2$ рёбрами между элементами каждого из них. Таким образом, из теоремы 1 получаем

Следствие 1. *Задача* Два плотных подграфа NP-*полна.*

Наконец, в работе будет использоваться NP-полная [15] задача Разбиение. Заметим, что в этой задаче, добавив к входному множеству столько нулей, сколько элементов в этом множестве, можно дополнительно потребовать, чтобы искомое подмножество содержало ровно половину элементов входного множества. После этого можно считать, что все элементы ненулевые (в противном случае можно увеличить каждый элемент на 1). Таким образом, будет использоваться следующая NP-полная задача.

Задача Разбиение. *Дано множество* $\{n_1, \ldots, n_{2S}\} \subset \mathbb{N} \setminus \{0\}$ *натуральных чисел. Верно ли, что существует подмножество индексов* $\mathcal{I} \subset \{1, \ldots, 2S\}$, $|\mathcal{I}| = S$, *таких, что* $\sum\limits_{i \in \mathcal{I}} n_i = T$, *где* $2T = \sum\limits_{i=1}^{2S} n_i$?

2. Случай двух кластеров при произвольной размерности пространства

Далее будем рассматривать задачу 2, в которой число кластеров фиксировано (не является частью входа задачи). В этом разделе рассмотрим случай двух кластеров и произвольной размерности пространства

и покажем, что в этом случае задача 2 (а значит, и задача 1) NP-полна в сильном смысле, используя задачу Два плотных подграфа.

Теорема 2. *Задача 2 при $k = 2$ NP-полна в сильном смысле.*

Доказательство. Построим полиномиальное сведение задачи Два плотных подграфа к задаче 2 при $k = 2$.

Рассмотрим произвольный пример задачи Два плотных подграфа — граф $G = (V, E)$ с чётным числом вершин $|V| = 2S$, $S \in \mathbb{N}$, и положительное целое число L_3.

Используя граф G, построим пример задачи 2. Для мощности кластера и ограничения на разброс положим

$$M = S, \quad A = (S-1)(2S-1) - \frac{2}{S}L_3.$$

Определим множество $\mathcal{Y}$. Пусть вершины V и рёбра E занумерованы так, что $V = \{v_1, \ldots, v_{2S}\}$, а $E = \{e_1, \ldots, e_{|E|}\}$. Тогда каждой вершине v_i поставим в соответствие элемент множества $\mathcal{Y}$ — точку $x_i \in \mathbb{R}^{|E|+D}$,

$$D = \sum_{v \in V} \overline{\deg}\, v,$$

где $\overline{\deg}\, v = |V| - 1 - \deg v$ — степень вершины $v \in V$ в дополнении графа G, а $\deg v$ — степень вершины v в самом графе G. Определим x_i следующим образом:

$$x_i = (x_i^{(1)}, \ldots, x_i^{(|E|)}, x_i^{(|E|+1)}, \ldots, x_i^{(|E|+D)}),$$

где при $j = 1, \ldots, |E|$ выполнено $x_i^{(j)} = 1$, если i-я вершина v_i инцидентна j-му ребру e_j, и $x_i^{(j)} = 0$ иначе; при $j = |E|+1, \ldots, |E|+D$ выполнено $x_i^{(j)} = 1$, если

$$\sum_{s=1}^{i-1} \overline{\deg}\, v_s < j - |E| \leqslant \sum_{s=1}^{i} \overline{\deg}\, v_s,$$

и $x_i^{(j)} = 0$ иначе.

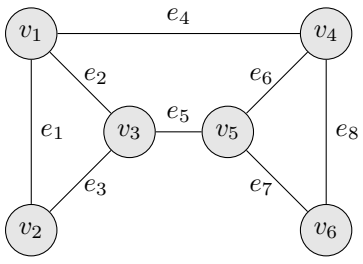


Рис. 2. Пример задачи Два плотных подграфа

Например, вершинам графа на 6 вершинах, изображённого на рис. 2, будут соответствовать точки

$$\begin{aligned}
x_1 &= (\overbrace{1101\,0000}^{|E|=8} \mid \overbrace{1100\,0000\,0000\,00}^{D=14}),\\
x_2 &= (1010\,0000 \mid 0011\,1000\,0000\,00),\\
x_3 &= (0110\,1000 \mid 0000\,0110\,0000\,00),\\
x_4 &= (0001\,0101 \mid 0000\,0001\,1000\,00),\\
x_5 &= (0000\,1110 \mid 0000\,0000\,0110\,00),\\
x_6 &= (0000\,0011 \mid 0000\,0000\,0001\,11).
\end{aligned}$$

Пусть для рассматриваемого примера задачи Два плотных подграфа существует решение, т. е. существует пара множеств $V_1, V_2 \subset V$ таких, что $V_1 \cap V_2 = \varnothing$, $|V_i| = S$, а каждое из сужений G_1, G_2 исходного графа G на эти множества содержит не менее L_3 рёбер.

Используя эти множества V_1, V_2, построим допустимое решение для примера задачи 2, а именно покажем, что множества

$$\mathcal{C}_i = \{x_j,\ j = 1, \ldots, 2S \mid v_j \in V_i\}, \quad i = 1, 2,$$

являются допустимым решением. Для этого достаточно показать, что выполняется ограничение на разбросы множеств $\mathcal{C}_i$, так как условие на мощности выполнено по построению. Рассмотрим произвольное S-элементное подмножество $\mathcal{X} \subset \mathcal{Y}$ и покажем, как его разброс выражается через вершины графа G, соответствующие его элементам, а затем применим получившуюся формулу к множествам $\mathcal{C}_i$. Для вычисления разброса $F(\mathcal{X})$ будем использовать следующую формулу (см., например, равенство (9) в работе [10]):

$$\sum_{y \in \mathcal{X}} \|y - \overline{y}(\mathcal{X})\|^2 = \frac{1}{2|\mathcal{X}|} \sum_{y,z \in \mathcal{X}} \|y - z\|^2. \tag{1}$$

Найдём, чему равна величина $\|y - z\|^2$ для пары различных точек $y, z \in \mathcal{X}$. Прежде всего, для этой величины справедливо равенство

$$\|y - z\|^2 = \sum_{j=1}^{|E|} (y^{(j)} - z^{(j)})^2 + \sum_{j=|E|+1}^{|E|+D} (y^{(j)} - z^{(j)})^2, \tag{2}$$

где $y^{(j)}$ и $z^{(j)}$ — j-е координаты точек y и z соответственно. При этом первые $|E|$ координат каждой точки имеют столько единиц, какова степень соответствующей вершины в графе. Кроме того, у двух точек среди этих координат может быть общая единица, только если между соответствующими вершинами есть ребро. Таким образом, если точкам y, z

соответствуют вершины v_y, v_z, то первая сумма в правой части (2) равна

$$\sum_{j=1}^{|E|}(y^{(j)} - z^{(j)})^2 = \deg v_y + \deg v_z - 2I(v_y v_z \in E), \tag{3}$$

где $I(v_y v_z \in E) = 1$, если вершины v_y и v_z смежны, и $I(v_y v_z \in E) = 0$ иначе. При этом вторая сумма в правой части (2) равна

$$\sum_{j=|E|+1}^{|E|+D}(y^{(j)} - z^{(j)})^2 = \overline{\deg}\, v_y + \overline{\deg}\, v_z, \tag{4}$$

так как в последних D координатах точек y и z единицы стоят на разных местах. Объединяя (2)–(4), получаем

$$\|y - z\|^2 = \deg v_y + \deg v_z - 2I(v_y v_z \in E) + \overline{\deg}\, v_y + \overline{\deg}\, v_z = \\ = 2(2S - 1 - I(v_y v_z \in E)).$$

Используя предыдущую формулу, а также (1), получаем, что разброс множества $\mathcal{X}$ может быть записан в следующем виде:

$$F(\mathcal{X}) = \frac{1}{2S}\sum_{\substack{y,z\in\mathcal{X},\\ y\neq z}}\|y - z\|^2 = (S-1)(2S-1) - \frac{1}{S}\sum_{\substack{y,z\in\mathcal{X},\\ y\neq z}} I(v_y v_z \in E). \tag{5}$$

Используя (5), можем переписать ограничение на разброс $F(\mathcal{X}) \leqslant A$ для произвольного S-элементного подмножества $\mathcal{X}$ в следующем эквивалентном виде:

$$\sum_{\substack{y,z\in\mathcal{X},\\ y\neq z}} I(v_y v_z \in E) \geqslant S((S-1)(2S-1) - A) = 2L_3. \tag{6}$$

Так как в правой части (6) каждая пара вершин из подмножества $\mathcal{X}$ учитывается дважды, а соответствующие множествам $\mathcal{C}_1$, $\mathcal{C}_2$ графы G_1, G_2 содержат не менее L_3 рёбер, множества $\mathcal{C}_1, \mathcal{C}_2$ удовлетворяют требуемым ограничениям на разброс, а значит, образуют решение построенного примера задачи 2.

Предположим, что у построенного примера задачи 2 существует решение — множества $\mathcal{C}_1$ и $\mathcal{C}_2$. Поскольку $|\mathcal{Y}| = 2S$ и каждое из множеств должно содержать не менее S элементов, то $|\mathcal{C}_1| = |\mathcal{C}_2| = S$. Рассмотрим множества вершин

$$V_i = \{v_j,\, j = 1, \ldots, 2S \mid x_j \in \mathcal{C}_i\}, \quad i = 1, 2.$$

Так как множества $\mathcal{C}_i$, $i = 1, 2$, удовлетворяют ограничению на разброс, из (6) следует, что графы G_1, G_2, индуцированные множествами V_1, V_2, содержат как минимум L_3 рёбер и оба состоят из S вершин.

Другими словами, V_1, V_2 являются решением задачи Два плотных подграфа. Тем самым задача 2 при $k = 2$ NP-полна.

Сильная NP-полнота следует из того факта, что частный случай задачи 2, индуцированный построенными примерами, не является задачей с числовыми параметрами, поскольку все входные значения либо равны единице или нулю (координаты точек входного множества $\mathcal{Y}$), либо ограничены полиномиальной функцией по длине входа (ограничение на разброс A кластеров). Теорема 2 доказана.

3. Одномерный случай

В этом разделе, в отличие от предыдущего, дополнительно предположим, что размерность d пространства равна единице — в предыдущем разделе величина d была частью входа задачи. Таким образом, будем рассматривать задачу 2 в случае, когда требуется найти два кластера, а все элементы входного множества $\mathcal{Y}$ имеют только одну вещественную координату.

Теорема 3. *Задача* 2 *при* $k = 2$ NP-*полна в случае фиксированной размерности пространства* $d = 1$.

Доказательство. Рассмотрим произвольный пример задачи Разбиение — $\{n_1, \ldots, n_{2S}\} \subset \mathbb{N} \setminus \{0\}$, $\sum\limits_{i=1}^{2S} n_i = 2T$. Без ограничения общности будем считать, что $S \geqslant 6$.

Построим следующий пример задачи 2 при $k = 2$ и $d = 1$: множество

$$\mathcal{Y} = \{B, B, n_1, \ldots, n_{2S}\},$$

где

$$B = -\max\left\{4T,\, N\left\lceil\sqrt{2S(S+1)}\,\right\rceil,\, \frac{1}{2}(2S+1)SN^2\right\} - 1, \quad N = \max_{i=1,\ldots,2S} n_i,$$

мощность кластеров $M = S + 1$, ограничение на разброс

$$A = \frac{S}{S+1}B^2 - \frac{2B}{S+1}T + SN^2.$$

Пусть у рассматриваемого примера задачи Разбиение имеется решение, т. е. существует множество индексов $\mathcal{I}$, $|\mathcal{I}| = S$, такое, что $\sum\limits_{i\in\mathcal{I}} n_i = T$. Рассмотрим пару множеств

$$\mathcal{C}_i = \{B\} \cup \{n_j \mid j \in \mathcal{I}_i\}, \quad i = 1, 2, \tag{7}$$

где $\mathcal{I}_1 = \mathcal{I}$, а $\mathcal{I}_2 = \{1, \ldots, 2S\} \setminus \mathcal{I}$.

Оценим разбросы $F(C_i)$, $i = 1, 2$, этих множеств. Для произвольного $\mathcal{C} \subset \mathbb{R}^d$ справедлива формула (см., например, цепочку (7) равенств в работе [10])

$$\sum_{y \in \mathcal{C}} \|y - \overline{y}(C)\|^2 = \sum_{y \in \mathcal{C}} \|y\|^2 - |\mathcal{C}| \cdot \|\overline{y}(\mathcal{C})\|^2. \tag{8}$$

Подставляя в (8) $\mathcal{C} = \mathcal{C}_i$, получаем, что для произвольного множества $\mathcal{C}_i$, имеющего структуру (7), выполнено

$$F(\mathcal{C}_i) = \sum_{y \in \mathcal{C}_i} |y - \overline{y}(\mathcal{C}_i)|^2 = \sum_{y \in \mathcal{C}_i} y^2 - |\mathcal{C}_i| \cdot (\overline{y}(\mathcal{C}_i))^2 =$$

$$= B^2 + \sum_{j \in \mathcal{I}_i} n_j^2 - \frac{1}{S+1}\Big(\sum_{j \in \mathcal{I}_i} n_j + B\Big)^2.$$

После раскрытия скобок и приведения подобных слагаемых получаем

$$F(\mathcal{C}_i) = \frac{S}{S+1}B^2 - \frac{2B}{S+1}\sum_{j \in \mathcal{I}_i} n_j + \sum_{j \in \mathcal{I}_i} n_j^2 - \frac{1}{S+1}\Big(\sum_{j \in \mathcal{I}_i} n_j\Big)^2. \tag{9}$$

Чтобы оценить величину $F(\mathcal{C}_i)$ сверху, применим равенство $\sum\limits_{j \in \mathcal{I}} n_j = T$, а также неравенства $n_j^2 \leqslant N^2$, $j = 1, \ldots, 2S$. Тогда из (9) следует оценка

$$F(\mathcal{C}_i) \leqslant \frac{S}{S+1}B^2 - \frac{2B}{S+1}T + SN^2 = A,$$

что означает, что множества $\mathcal{C}_i$, $i = 1, 2$, являются решением построенного примера задачи 2.

Пусть построенный пример задачи 2 имеет решение — множества $\mathcal{C}_i$, $i = 1, 2$.

Сначала покажем, что оба элемента B из входного $\mathcal{Y}$ не могут лежать в одном подмножестве. Для этого предположим обратное — что одно из подмножеств, например $\mathcal{C}_1$, содержит оба элемента B, т. е. имеет вид

$$\mathcal{C}_1 = \{B, B\} \cup \{n_j \mid j \in \mathcal{I}_1\}, \quad |\mathcal{I}_1| = S - 1.$$

Чтобы прийти к противоречию, покажем, что множество $\mathcal{C}_1$ не удовлетворяет ограничению на разброс. Для этого понадобится вспомогательное неравенство: предварительно покажем, что разброс A удовлетворяет неравенству $A < B^2$. Действительно, используя определение A, это неравенство можно переформулировать следующим образом:

$$\frac{S}{S+1}B^2 - \frac{2B}{S+1}T + SN^2 < B^2,$$

или эквивалентно

$$-\frac{2T}{S+1}B + SN^2 < \frac{1}{S+1}B^2. \tag{10}$$

Из определения B следует, что выполнены следующие два неравенства:

$$-\frac{2B}{S+1}T < \frac{1}{2(S+1)}B^2,$$

$$SN^2 < \frac{1}{2(S+1)}B^2,$$

после сложения которых получим, что неравенство (10) выполняется, а значит, $A < B^2$.

Оценим разброс $\mathcal{C}_1$ снизу:

$$F(\mathcal{C}_1) = \sum_{y\in\mathcal{C}_1} |y - \overline{y}(\mathcal{C}_1)|^2 \geqslant 2|B - \overline{y}(\mathcal{C}_1)|^2. \tag{11}$$

Заметим, что

$$\overline{y}(\mathcal{C}_1) = \frac{1}{S+1}\Big(2B + \sum_{j\in\mathcal{I}_1} n_j\Big) > \frac{2B}{S+1}.$$

Так как $B < \frac{2B}{S+1}$ и $S \geqslant 6$, оценку (11) можно продолжить:

$$F(\mathcal{C}_1) \geqslant 2|B - \overline{y}(\mathcal{C}_1)|^2 > 2\Big(B - \frac{2B}{S+1}\Big)^2 = 2B^2\Big(\frac{S-1}{S+1}\Big)^2 \geqslant B^2,$$

что противоречит неравенству $A < B^2$. Значит, оба элемента B не могут лежать в одном множестве, т. е. оба множества имеют вид (7).

Предположим, что сумма элементов d_i, $i \in \mathcal{I}_1$, не равна T. Без ограничения общности можно считать, что $\sum\limits_{i\in\mathcal{I}_1} d_i \geqslant T+1$, так как в противном случае можно поменять $\mathcal{I}_1$ и $\mathcal{I}_2$ местами.

Оценим разброс множества $\mathcal{C}_1$ снизу, используя равенство (9):

$$F(\mathcal{C}_1) = \frac{S}{S+1}B^2 - \frac{2B}{S+1}\sum_{j\in\mathcal{I}_1} n_j + \sum_{j\in\mathcal{I}_1} n_j^2 - \frac{1}{S+1}\Big(\sum_{j\in\mathcal{I}_1} n_j\Big)^2 \geqslant$$

$$\geqslant \frac{S}{S+1}B^2 - \frac{2B}{S+1}(T+1) + 0 - \frac{1}{S+1}(SN)^2 =$$

$$= A - \frac{2B}{S+1} - SN^2 - \frac{1}{S+1}(SN)^2 = A - \frac{1}{S+1}(2B + (2S+1)SN^2).$$

Так как для выбранного B выполнено $-2B > (2S+1)SN^2$, имеем $F(\mathcal{C}_1) > A$. Таким образом, наше предположение приводит к противоречию с допустимостью множеств $\mathcal{C}_1$, $\mathcal{C}_2$. Значит, сумма элементов n_j, $j \in \mathcal{I}_1$, равна T.

Следовательно, исходный пример задачи Разбиение и построенный пример задачи 2 одновременно либо имеют решения, либо не имеют. Значит, задача 2 при $k = 2$ NP-трудна в одномерном случае. Теорема 3 доказана.

Обобщим полученный в теореме 3 результат на случай произвольного фиксированного числа кластеров $k \geqslant 2$, используя индукцию по k.

Теорема 4. *Для произвольного $k \geqslant 2$ задача 2 с фиксированным числом кластеров k* NP*-полна даже в одномерном случае.*

Доказательство. Докажем утверждение индукцией по числу кластеров k. NP-полнота в случае $k = 2$ доказана в теореме 3.

Допустим, что задача 2 для k кластеров NP-полна в одномерном случае, и покажем, что задача 2 для $k+1$ кластеров также NP-полна в этом случае. Для этого построим полиномиальное сведение случая k кластеров к $k + 1$.

Рассмотрим произвольный одномерный пример задачи 2 для k кластеров — множество $\mathcal{Y}$, $|\mathcal{Y}| = kM$, ограничение на разброс A. Построим следующий пример задачи 2 для $k + 1$ кластеров: ограничение на разброс A остаётся таким же, входное множество $\widetilde{\mathcal{Y}}$ равно

$$\widetilde{\mathcal{Y}} = \mathcal{Y} \cup \{B_1, \ldots, B_M\}, \quad B_i = B,\ i = 1, \ldots, M,$$

где $B = \left\lceil \sqrt{2M(A+1)} \right\rceil + N$, а $N = \max\limits_{y \in \mathcal{Y}} |y|$ — максимальное абсолютное значение элементов из $\mathcal{Y}$.

Покажем, что оба примера имеют решения либо не имеют одновременно. Пусть существует решение задачи 2 для k кластеров — $\mathcal{C}_i$, $i = 1, \ldots, k$. Рассмотрим $k + 1$ множеств

$$\widetilde{\mathcal{C}}_i = \mathcal{C}_i,\ i = 1, \ldots, k, \quad \widetilde{\mathcal{C}}_{k+1} = \{B_1, \ldots, B_M\}.$$

Очевидно, что мощности всех этих множеств равны M и они попарно не пересекаются. Разбросы множеств $\widetilde{\mathcal{C}}_i$ не превосходят порога A по предположению, а разброс $\widetilde{\mathcal{C}}_{k+1}$ равен 0, так как все его элементы одинаковы. Следовательно, множества $\widetilde{\mathcal{C}}_i$, $i = 1, \ldots, k + 1$, являются решением построенного примера задачи 2 для $k + 1$ кластеров.

Допустим, что построенный пример задачи 2 с $k+1$ кластерами имеет решение — множества $\widetilde{\mathcal{C}}_i$, $|\widetilde{\mathcal{C}}_i| = M$, $i = 1, \ldots, k + 1$. Покажем, что все элементы B_i лежат в одном множестве. Для этого рассмотрим множество, содержащее хотя бы один элемент B_i (такое множество существует, так как по предположению $\widetilde{\mathcal{C}}_i$, $i = 1, \ldots, k + 1$, — это разбиение $\widetilde{\mathcal{Y}}$). Без ограничения общности можно считать, что это множество $\widetilde{\mathcal{C}}_{k+1}$. Допустим, что в $\widetilde{\mathcal{C}}_{k+1}$ есть элемент не из B_i. Тогда разброс множества $\widetilde{\mathcal{C}}_{k+1}$ можно

оценить снизу, используя формулу (1):

$$F(\widetilde{\mathcal{C}}_{k+1}) = \frac{1}{2M} \sum_{x,y \in \widetilde{\mathcal{C}}_{k+1}} |x-y|^2 \geqslant \frac{1}{2M} |x^* - B|^2,$$

где x^* — некоторый элемент из $\mathcal{Y}$. Учитывая определение B и то, что $|x^*| \leqslant N$, последнее неравенство можно продолжить:

$$F(\widetilde{\mathcal{C}}_{k+1}) \geqslant \frac{1}{2M}(B-N)^2 \geqslant A+1 > A.$$

Таким образом, разброс $F(\widetilde{\mathcal{C}}_{k+1})$ строго больше A, что противоречит допустимости решения $\widetilde{\mathcal{C}}_i$, $i = 1, \ldots, k+1$. Стало быть, все элементы B_i содержатся в множестве $\widetilde{\mathcal{C}}_{k+1}$. Тогда множества $\mathcal{C}_i = \widetilde{\mathcal{C}}_i$, $i = 1, \ldots, k$, являются решением задачи 2 для k кластеров.

Таким образом, задачи 2 для k и $k+1$ кластеров одновременно либо имеют решения, либо не имеют. Следовательно, доказаны база и шаг индукции, что означает, что задача 2 с фиксированным числом кластеров k NP-полна в одномерном случае для произвольного $k \geqslant 2$. Теорема 4 доказана.

Таким образом, для произвольного числа кластеров $k \geqslant 2$ задачи 1 и 2 для k кластеров NP-полны даже в одномерном случае. Заметим, что для этих задач существует [16] псевдополиномиальный алгоритм, поэтому они не будут NP-полными в сильном смысле.

Заключение

В работе рассмотрены задачи поиска непересекающихся подмножеств большой мощности при ограничении на их разброс. Доказано, что задачи NP-полны в сильном смысле в случае поиска двух кластеров и размерности пространства, являющейся частью входа задачи, а также что задачи NP-полны в одномерном случае.

Интересным направлением дальнейших исследований является выяснение сложности задач в случае фиксированной мощности искомых кластеров.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0015). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Pérez-Ortega J., Almanza-Ortega N. N., Vega-Villalobos A.** [et al.]. The K-means algorithm evolution // Introduction to data science and machine learning. Rijeka: IntechOpen, 2019. 22 p. DOI: 10.5772/intechopen.85447.
2. **Ikotun A. M., Ezugwu A. E., Abualigah L.** [et al.]. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data // Inf. Sci. 2023. V. 622. P. 178–210. DOI: 10.1016/j.ins.2022.11.139.
3. **Grant R. W., McCloskey J., Hatfield M.** [et al.]. Use of latent class analysis and k-means clustering to identify complex patient profiles // JAMA Netw. Open. 2020. V. 3, No. 12. Article ID e2029068. 13 p. DOI: 10.1001/jamanetworkopen.2020.29068.
4. **Dhanachandra N., Manglem K., Chanu Y. J.** Image segmentation using k-means clustering algorithm and subtractive clustering algorithm // Proc. Comput. Sci. 2015. V. 54. P. 764–771. DOI: 10.1016/j.procs.2015.06.090.
5. **Kumari R., Sheetanshu, Singh M. K., Jha R.** Anomaly detection in network traffic using K-mean clustering // 2016 3rd Int. Conf. Recent Advances in Information Technology (Dhanbad, India, Mar. 3–5, 2016). Piscataway: IEEE, 2016. P. 387–393. DOI: 10.1109/RAIT.2016.7507933.
6. **Aloise D., Deshpande A., Hansen P.** [et al.]. NP-hardness of Euclidean sum-of-squares clustering // Mach. Learn. 2009. V. 75, No. 2. P. 245–248. DOI: 10.1007/s10994-009-5103-0.
7. **Jørgensen A. G., Larsen K. G., Mathiasen A.** [et al.]. Fast exact k-means, k-medians and Bregman divergence clustering in 1D. Ithaca, NY, 2017. 16 p. (e-Print Archive / Cornell Univ.; arXiv:1701.07204). DOI: 10.48550/arXiv.1701.07204.
8. **Khodadadi A., Saeidi S.** Discovering the maximum k-clique on social networks using bat optimization algorithm // Comput. Soc. Netw. 2021. V. 8, No. 1. Article ID 6. 15 p. DOI: 10.1186/s40649-021-00087-y.
9. **Tomita E., Akutsu T., Matsunaga T.** Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics // Biomedical engineering, trends in electronics, communications and software. Rijeka: Intech Open, 2011. P. 625–640. DOI: 10.5772/13245.
10. **Кельманов А. В., Пяткин А. В.** NP-полнота некоторых задач выбора подмножества векторов // Дискрет. анализ и исслед. операций. 2010. Т. 17, № 5. С. 37–45.
11. **Aggarwal A., Imai H., Katoh N., Suri S.** Finding k points with minimum diameter and related problems // J. Algorithms. 1991. V. 12, No. 1. P. 38–56. DOI: 10.1016/0196-6774(91)90022-Q.
12. **Kel'manov A. V., Ruzankin P. S.** An accelerated exact algorithm for the one-dimensional M-variance problem // Pattern Recognit. Image Anal. 2019. V. 29, No. 4. P. 573–576. DOI: 10.1134/S1054661819040072.
13. **Пяткин А. В.** О сложности задачи выбора кластеров большого размера // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 2. С. 136–143.

14. Кельманов А. В., Пяткин А. В., Хандеев В. И. О сложности некоторых максиминных задач кластеризации // Тр. Ин-та математики и механики. 2018. Т. 24, № 4. С. 189–198.

15. Garey M. R., Johnson D. S. Computers and intractability: A guide to the theory of NP-completeness. San Francisco: Freeman, 1979. 338 p.

16. Khandeev V. I., Neshchadim S. M. Pseudo-polynomial algorithms for some problems of searching for the largest subsets // Mathematical optimization theory and operations research: Recent trends. Rev. Sel. Pap. 23th Int. Conf. (Omsk, Russia, June 30 – July 6, 2024). Cham: Springer, 2024. P. 319–333. (Commun. Comput. Inf. Sci.; V. 2239). DOI: 10.1007/978-3-031-73365-9_22.

Нещадим Сергей Михайлович
Хандеев Владимир Ильич

Статья поступила
5 мая 2025 г.
После доработки —
25 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 172–190

UDC 519.8 DOI: 10.33048/daio.2025.32.834

ON THE COMPLEXITY OF TWO PROBLEMS OF FINDING CLUSTERS OF LARGE CARDINALITY

S. M. Neshchadim[1, a] and *V. I. Khandeev*[2, b]

[1] Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
[2] Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: [a] s.neshchadim@g.nsu.ru, [b] khandeev@math.nsc.ru

Abstract. Clustering problems for a finite point set in Euclidean space are considered. The first problem requires each subset to have cardinality no smaller than a given threshold (not necessarily covering the entire set), while the second one requires all subsets to have the same cardinality and form a partition of the given set. In both problems for each subset, it is additionally required that the sum of squared distances to the centroid does not exceed a given value. Both problems are proven to be strongly NP-complete when the number of clusters is two and the space dimension is part of the input. Furthermore, NP-completeness is established for the one-dimensional case with an arbitrary fixed number of clusters. Illustr. 2, bibliogr. 16.

Keywords: clustering, bounded scatter, minimum cluster size, Euclidean space, NP-completeness.

References

1. **J. Pérez-Ortega, N. N. Almanza-Ortega, A. Vega-Villalobos,** [et al.], The K-means algorithm evolution, in *Introduction to Data Science and Machine Learning* (IntechOpen, Rijeka, 2019), DOI: 10.5772/intechopen.85447.
2. **A. M. Ikotun, A. E. Ezugwu, L. Abualigah,** [et al.], K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, *Inf. Sci.* **622**, 178–210 (2023), DOI: 10.1016/j.ins.2022.11.139.

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

3. **R. W. Grant, J. McCloskey, M. Hatfield,** [et al.], Use of latent class analysis and k-means clustering to identify complex patient profiles, *JAMA Netw. Open.* **3** (12), ID e2029068 (2020), DOI: 10.1001/jamanetworkopen.2020.29068.
4. **N. Dhanachandra, K. Manglem,** and **Y. J. Chanu,** Image segmentation using k-means clustering algorithm and subtractive clustering algorithm, *Proc. Comput. Sci.* **54**, 764–771 (2015), DOI: 10.1016/j.procs.2015.06.090.
5. **R. Kumari, M. K. Sheetanshu, Singh,** and **R. Jha,** Anomaly detection in network traffic using K-mean clustering, in *2016 3rd Int. Conf. Recent Advances in Information Technology* (Dhanbad, India, Mar. 3–5, 2016) (IEEE, Piscataway, 2016), pp. 387–393, DOI: 10.1109/RAIT.2016.7507933.
6. **D. Aloise, A. Deshpande, P. Hansen,** [et al.], NP-hardness of Euclidean sum-of-squares clustering, *Mach. Learn.* **75** (2), 245–248 (2009), DOI: 10.1007/s10994-009-5103-0.
7. **A. G. Jørgensen, K. G. Larsen, A. Mathiasen,** [et al.], Fast exact k-means, k-medians and Bregman divergence clustering in 1D (Ithaca, NY, 2017) (e-Print Archive / Cornell Univ., arXiv:1701.07204), DOI: 10.48550/arXiv.1701.07204.
8. **A. Khodadadi** and **S. Saeidi,** Discovering the maximum k-clique on social networks using bat optimization algorithm, *Comput. Soc. Netw.* **8** (1), ID 6 (2021), DOI: 10.1186/s40649-021-00087-y.
9. **E. Tomita, T. Akutsu,** and **T. Matsunaga,** Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics, in *Biomedical Engineering, Trends in Electronics, Communications and Software* (Intech Open, Rijeka, 2011), pp. 625–640, DOI: 10.5772/13245.
10. **A. V. Kel'manov** and **A. V. Pyatkin,** NP-completeness of some problems of choosing a vector subset, *Diskretn. Anal. Issled. Oper.* **17** (5), 37–45 (2010) [Russian] [*J. Appl. Ind. Math.* **5** (3), 352–357 (2011) DOI: 10.1134/S1990478911030069].
11. **A. Aggarwal, H. Imai, N. Katoh,** and **S. Suri,** Finding k points with minimum diameter and related problems, *J. Algorithms* **12** (1), 38–56 (1991), DOI: 10.1016/0196-6774(91)90022-Q.
12. **A. V. Kel'manov** and **P. S. Ruzankin,** An accelerated exact algorithm for the one-dimensional M-variance problem, *Pattern Recognit. Image Anal.* **29** (4), 573–576 (2019), DOI: 10.1134/S1054661819040072.
13. **A. V. Pyatkin,** On the complexity of the problem of choice of large clusters, *Diskretn. Anal. Issled. Oper.* **31** (2), 136–143 (2024) [Russian] [*J. Appl. Ind. Math.* **18** (2), 312–315 (2024), DOI: 10.1134/S1990478924020121].
14. **A. V. Kel'manov, A. V. Pyatkin,** and **V. I. Khandeev,** On the complexity of some max–min clustering problems, *Tr. Inst. Mat. Mekh.* **24** (4), 189–198 (2018) [Russian] [*Proc. Steklov Inst. Math.* **309** (Suppl. 1), S65–S73 (2020) DOI: 10.1134/S0081543820040082].
15. **M. R. Garey** and **D. S. Johnson,** *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).

16. V. I. Khandeev and **S. M. Neshchadim,** Pseudo-polynomial algorithms for some problems of searching for the largest subsets, in *Mathematical Optimization Theory and Operations Research: Recent Trends*, Rev. Sel. Pap. 23th Int. Conf. (Omsk, Russia, June 30 – July 6, 2024) (Springer, Cham, 2024), pp. 319–333 (Commun. Comput. Inf. Sci., V. 2239), DOI: 10.1007/978-3-031-73365-9_22.

Sergey M. Neshchadim
Vladimir I. Khandeev

Received May 5, 2025
Revised August 25, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 191–212

УДК 519.178 DOI: 10.33048/daio.2025.32.839

ПОЛИНОМИАЛЬНАЯ РАЗРЕШИМОСТЬ ЗАДАЧИ О НЕЗАВИСИМОМ МНОЖЕСТВЕ ДЛЯ НЕКОТОРЫХ НАСЛЕДСТВЕННЫХ КЛАССОВ ГРАФОВ С ЛОГАРИФМИЧЕСКИМИ И КВАЗИЛОГАРИФМИЧЕСКИМИ ОГРАНИЧЕНИЯМИ НА СТЕПЕНИ ИЛИ АНТИСТЕПЕНИ ВЕРШИН

С. В. Сорочан

Нижегородский гос. университет им. Н. И. Лобачевского,
пр. Гагарина, 23, 603950 Нижний Новгород, Россия

E-mail: sergey.sorochan@itmm.unn.ru

Аннотация. Триод — это дерево, содержащее не более трёх листьев и не более одной вершины степени 3. Задача о наибольшем независимом множестве (ННМ) разрешима за полиномиальное время для графов, не содержащих в качестве подграфа граф, у которого каждая компонента является триодом с любым фиксированным числом вершин. Если же запрещается k-вершинный порождённый подграф с триодными компонентами, то при $k > 5$ вопрос о сложности решения этой задачи остаётся открытым. Пусть F — граф, у которого каждая компонента является триодом, т. е. триодный граф. Известно, что если задача о ННМ полиномиально разрешима в классе всех графов, не содержащих F в качестве порождённого подграфа, то она полиномиально разрешима и в классе всех графов, свободных от графа $F + O_s$, получаемого из F добавлением s изолированных вершин. Если же запретить порождённый подграф вида $F + P_2$, где P_2 — цепь с двумя вершинами, то полиномиальная разрешимость задачи о ННМ в классе всех графов без порождённых $F + P_2$ доказана только для очень небольшого числа триодных графов F, а для подавляющего большинства таких графов F сложностной статус задачи о ННМ в классе графов, свободных от $F + P_2$, остаётся неизвестным.

В этой статье рассматриваются так называемые графы с логарифмическими ограничениями. Для каждой вершины такого графа выполнено хотя бы одно из следующих трёх условий:

1) степень вершины не превосходит (с точностью до мультипликативной константы) логарифма от общего числа вершин в графе;

2) относительная степень вершины (т. е. максимальное число смежных с ней вершин таких, что все эти смежные вершины несмежны с одной из них) не превосходит (с точностью до мультипликативной константы) логарифма от общего числа вершин в графе;

3) антистепень вершины (т. е. число несмежных с ней вершин) не превосходит (с точностью до мультипликативной константы) логарифма от общего числа вершин в графе.

Доказано, что для любого триодного графа F такого, что задача о ННМ полиномиально разрешима в классе графов, свободных от F, она остаётся полиномиально разрешимой и для всех графов с логарифмическими ограничениями, свободных от $F + P_2$. Также рассмотрено одно нетривиальное расширение этого множества графов, названное множеством графов с квазилогарифмическими ограничениями, и доказан аналогичный результат для всех таких графов, свободных от $F + P_2$. Предложены полиномиальные алгоритмы для решения задачи о ННМ для графов с логарифмическими и квазилогарифмическими ограничениями, свободных от $F + P_2$, и найдены верхние оценки сложности этих алгоритмов. Библиогр. 19.

Ключевые слова: независимое множество, монотонный класс, наследственный класс, НМ-простой класс, НМ-сложный класс, запрещённый подграф, триод, граф с логарифмическими ограничениями, граф с квазилогарифмическими ограничениями, полиномиальный алгоритм.

Введение

Под *классом графов* понимается множество графов, замкнутое относительно переименования вершин. Класс графов называется *наследственным*, если он замкнут относительно удаления вершин, и *монотонным*, если он замкнут относительно удаления вершин и рёбер. Любой наследственный класс $\mathcal{X}$ может быть задан множеством *запрещённых подграфов* $\mathcal{M}$: $\mathcal{X}$ состоит из всех графов, не содержащих порождённых подграфов, изоморфных графам из $\mathcal{M}$. В этом случае используется обозначение $\mathcal{X} = \mathrm{Free}(\mathcal{M})$, а графы из $\mathcal{X}$ называют $\mathcal{M}$-*свободными*. Если $\mathcal{M}$ — конечное множество, то класс $\mathrm{Free}(\mathcal{M})$ называется *конечно определённым*. Всякий монотонный класс наследственный, поэтому он тоже может быть задан множеством запрещённых (порождённых) подграфов.

Независимое множество в графе — это множество несмежных между собой вершин. *Задача о независимом множестве* состоит в том, чтобы в заданном графе найти независимое множество наибольшей мощности. Размер наибольшего независимого множества (ННМ) графа G называется его *числом независимости* и обозначается через $\alpha(G)$. Задача о независимом множестве NP-трудна в множестве всех графов и остаётся такой для многих классов графов; называем такие классы НМ-*сложными*.

Известно также немало классов графов, для которых эта задача может быть решена за полиномиальное время, их называем НМ-*простыми*.

Кроме того, есть информация общего характера, относящаяся не к отдельным классам, а к семействам классов. Так, в [1] установлено, что конечно определённый монотонный класс $\mathcal{X}$ будет НМ-сложным, если $\mathcal{T} \subseteq X$, и НМ-простым в противном случае. Здесь $\mathcal{T}$ — класс всех графов, у которых каждая компонента связности является деревом не более чем с тремя листьями. Такое дерево называется *триодом*.

Обозначенную дихотомию пока не удалось распространить на наследственные классы, не являющиеся монотонными. В [2] доказано, что любой конечно определённый наследственный класс, включающий класс $\mathcal{T}$, будет НМ-сложным. Движение в обратном направлении связано с разработкой полиномиальных алгоритмов для наследственных классов, определяемых запрещёнными подграфами из $\mathcal{T}$. Если говорить только о классах, определяемых одним запрещённым подграфом из $\mathcal{T}$, то наибольшие продвижения здесь состоят в установлении НМ-простоты следующих классов:

- Free(mP_2) при любом натуральном m [3], где P_2 — цепь на двух вершинах;
- Free($K_{1,3}$) [4], где $K_{1,3}$ — четырёхвершинный триод с тремя листьями, названный *клешнёй*;
- Free($T_{1,1,2}$) [5], где $T_{1,1,2}$ — триод на пяти вершинах, названный *вилкой*, т. е. граф, получаемый из графа $K_{1,3}$ подразбиением одного ребра;
- Free($P_2 + K_{1,3}$) [6], где $P_2 + K_{1,3}$ — дизъюнктное объединение графов P_2 и $K_{1,3}$;
- Free(P_5) [7];
- Free(P_6) [8].

Также имеется ряд результатов о НМ-простоте некоторых подмножеств классов, определяемых запрещёнными подграфами из $\mathcal{T}$ (см., например, [9–13]). Кроме того, отметим работы [14, 15], содержащие идеи, которые используются в [10].

Как видно, в отличие от монотонных классов, для конечно определённых наследственных классов графов, не являющихся монотонными, имеется значительный разрыв в наших знаниях о НМ-простых и НМ-сложных классах. По-видимому, трудно рассчитывать на ликвидацию этого разрыва в ближайшем будущем, и имеет смысл испытать другие подходы к проблеме разделения НМ-простых и НМ-сложных наследственных классов. Одним из направлений может быть рассмотрение семейств классов графов, промежуточных между семействами монотонных и наследственных классов. Для этих промежуточных семейств можно надеяться получить результаты дихотомического характера типа упомянутого выше, либо хотя бы приблизиться к этому.

Первая попытка получения новых результатов в данном направлении предпринята в [16]. Если наследственный класс определяется одним запрещённым подграфом F, то множество запрещённых подграфов, состоящее из всех остовных надграфов графа F, определяет монотонный класс. Если ограничить добавление рёбер какими-либо правилами, то получится класс, заключённый между этими двумя. Вводя ограничения на множество добавляемых рёбер, получаем возможность определять семейства классов графов, промежуточные между семействами монотонных и наследственных классов. В [16] рассмотрено три типа ограничений на добавление рёбер к цепи P_k (с k вершинами) и доказана НМ-простота следующих наследственных классов:

• множество запрещённых подграфов состоит из всех остовных надграфов цепи P_k с минимальной степенью вершин меньше $k/2$;

• множество запрещённых подграфов состоит из всех остовных надграфов цепи P_k, в дополнительных графах которых меньше $k/2$ рёбер;

• множество запрещённых подграфов состоит из всех остовных надграфов цепи P_k, из которых с помощью операции пересечения графов можно получить P_k.

Исследования, начатые в [16], получили продолжение и развитие в статье [17]. Так как всякая простая цепь содержится в качестве порождённого подграфа в некотором триоде, следующим естественным шагом в заданном направлении стало рассмотрение аналогичных типов ограничений по отношению к триодам.

Пусть $T_{i,j,k}$ — это триод, у которого из единственной вершины степени 3 выходят простые цепи длины i, j, k соответственно, и пусть $s = i + j + k$. В продолжение результатов работы [16] в [17] рассмотрено три типа ограничений на добавление рёбер к триоду $T_{i,j,k}$, которые задают более хитро устроенные классы, промежуточные между монотонными и наследственными, а именно:

• множество запрещённых подграфов состоит из всех остовных надграфов триода $T_{i,j,k}$, у которых минимальная степень вершины меньше $\frac{1}{2}(s + i + 1)$;

• множество запрещённых подграфов состоит из всех остовных надграфов триода $T_{i,j,k}$, у которых дополнительный граф имеет меньше $s/2 - 1$ рёбер;

• множество запрещённых подграфов состоит из всех надграфов триода $T_{i,j,k}$, из которых с помощью операции пересечения графов можно получить $T_{i,j,k}$.

Для первых двух из этих классов и некоторого нетривиального подкласса третьего класса в [17] доказана НМ-простота: представлены полиномиальные алгоритмы решения задачи о ННМ в этих классах, и приведены верхние оценки сложности решающих алгоритмов.

В этой статье будут рассмотрены типы ограничений, похожие на исследованные в работах [16, 17], но привязанные к размеру входного графа, для которого будет решаться задача о ННМ. А именно, мы будем рассматривать такие графы, каждая вершина которых удовлетворяет хотя бы одному из следующих ограничений:

• степень вершины не превосходит (с точностью до мультипликативной константы) логарифма от общего числа вершин в графе;

• относительная степень вершины (максимальное число её соседей таких, что все они несмежны с одним из выбранных соседей) не превосходит логарифма от общего числа вершин в графе (с точностью до мультипликативной константы);

• антистепень вершины (число несмежных с ней вершин) не превосходит логарифма от общего числа вершин в графе (с точностью до мультипликативной константы); в дальнейшем это ограничение будет немного ослаблено повышением верхней границы для значения антистепени.

Графы, удовлетворяющие этим свойствам, для краткости будем называть *графами с логарифмическими ограничениями*, а те из них, для которых ограничение на антистепени ослабляется за счёт небольшого повышения верхней границы для их значений, будем называть *графами с квазилогарифмическими ограничениями*.

Мотивацией для рассмотрения таких ограничений являются следующие известные результаты. Пусть F — фиксированный граф, у которого каждая компонента связности является некоторым триодом. Известно, что если задача о ННМ полиномиально разрешима в классе графов $\mathrm{Free}(F)$, то она полиномиально разрешима и в классе графов $\mathrm{Free}(F+O_s)$ (граф $F+O_s$ получается из F добавлением фиксированного числа s изолированных вершин). Если же запретить порождённый подграф вида $F+P_2$, где P_2 — цепь с двумя вершинами, то полиномиальная разрешимость задачи о ННМ в классе $\mathrm{Free}(F+P_2)$ доказана только для очень небольшого числа триодных графов F (из перечисленных выше работ такие результаты получены только в [3, 6]), а для подавляющего большинства графов F, состоящих из триодных компонент, сложностной статус задачи о ННМ в классе графов $\mathrm{Free}(F+P_2)$ остаётся неизвестным.

Мы докажем, что для любого фиксированного триодного графа F такого, что задача о ННМ полиномиально разрешима в классе $\mathrm{Free}(F)$, она остаётся полиномиально разрешимой и для всех графов с логарифмическими ограничениями, принадлежащих классу $\mathrm{Free}(F+P_2)$. Затем этот результат будет распространён и на все графы с квазилогарифмическими ограничениями из класса $\mathrm{Free}(F+P_2)$. Будут предложены полиномиальные алгоритмы для решения задачи о ННМ для графов с логарифмическими и квазилогарифмическими ограничениями, и найдены верхние оценки их сложности.

В статье применяются следующие обозначения:

- $G_1 + G_2$ — дизъюнктное объединение графов G_1 и G_2;
- $\alpha(G)$ — число независимости графа G (размер его ННМ);
- $N(a)$ — множество вершин графа, смежных с вершиной a (*окрестность* вершины a);
- $\overline{N}(a)$ — множество вершин графа, несмежных с вершиной a (*антиокрестность* вершины a).

Для произвольного подмножества X множества вершин графа G введём следующие обозначения:

- $G[X]$ — подграф графа G, порождённый множеством X;
- $G - X$ — подграф, полученный удалением из графа G всех вершин множества X;
- $N(X)$ — множество всех вершин графа, смежных с вершинами из X (*окрестность* множества X);
- $\overline{N}(X)$ — множество всех вершин графа, несмежных с вершинами из X (*антиокрестность* множества X);
- $\deg_X a$ — число вершин в множестве X, смежных с вершиной a (*степень* вершины a в множестве X);
- $\overline{\deg}_X a$ — число вершин в множестве X, несмежных с вершиной a (*антистепень* вершины a в множестве X).

Если X совпадает с множеством всех вершин графа, то в двух последних обозначениях опускаем индекс X.

1. Класс графов без порождённого подграфа $F + O_s$

Этот раздел предварительный. В нём для любого фиксированного графа F, у которого каждая компонента связности является некоторым триодом, при любом фиксированном натуральном s мы рассмотрим класс $\mathrm{Free}(F + O_s)$ всех графов, не содержащих граф $F + O_s$ в качестве порождённого подграфа.

Во многих работах по тематике задачи о ННМ (наверное, первая из таких работ — это статья [2]) имеется упоминание довольно просто доказываемого результата, говорящего о том, что если задача о ННМ полиномиально разрешима в классе графов $\mathrm{Free}(F)$, то она полиномиально разрешима и в классе графов $\mathrm{Free}(F+O_s)$ при любом фиксированном натуральном s. Однако полиномиального алгоритма для нахождения ННМ в графах из класса $\mathrm{Free}(F + O_s)$ в явном виде ни в [2], ни в других работах представлено не было, как и не было выполнено оценки его трудоёмкости в формальном виде. Довольно простая идея этого алгоритма, краткое описание которой можно найти, например, в учебниках [18, 19], будет использована и несколько усовершенствована при выводе основных результатов данной статьи, поэтому здесь приведём доказательство НМ-

простоты для класса $\mathrm{Free}(F + O_s)$ с выводом полиномиальной оценки трудоёмкости решающего алгоритма.

Пусть $\mathcal{X}$ — произвольный наследственный класс графов, а G — n-вершинный граф из $\mathcal{X}$. Через $t_n(G)$ обозначим сложность решения задачи о ННМ для графа G, а через $t_n(\mathcal{X})$ — сложность её решения в множестве всех n-вершинных графов из класса $\mathcal{X}$. Если $\mathcal{X}_n$ — это множество всех n-вершинных графов из $\mathcal{X}$, то очевидно, что $t_n(\mathcal{X}) = \max\limits_{G \in \mathcal{X}_n} t_n(G)$.

Теорема 1. *Пусть F — произвольный граф с триодными компонентами такой, что задача о* ННМ *в классе* $\mathrm{Free}(F)$ *полиномиально разрешима и сложность решения в этом классе равна $t_n(\mathrm{Free}(F)) = O(n^k)$, где k — некоторая натуральная константа. Тогда для любой фиксированной натуральной константы s задача о* ННМ *в классе* $\mathrm{Free}(F + O_s)$ *тоже полиномиально разрешима, причём $t_n(\mathrm{Free}(F + O_s)) = O(n^{k+s})$.*

Доказательство. Вывод оценки сложности решения задачи о ННМ в классе $\mathrm{Free}(F + O_s)$ использует простую идею самого общего (а потому экспоненциального в общем случае) алгоритма поиска ННМ в произвольном графе [18, 19].

Наибольшее независимое множество любого графа есть объединение наибольших независимых множеств его компонент связности. Тем самым, не ограничивая общности, можно считать, что входной граф G связный. Его ННМ можно найти следующим способом. Возьмём произвольную вершину a графа G, не являющуюся изолированной вершиной. Если она не принадлежит никакому ННМ графа G, то $\alpha(G) = \alpha(G - a)$, а если принадлежит хотя бы одному ННМ, то ясно, что $\alpha(G) = \alpha(G - N(a)) = \alpha(G[\overline{N}(a)]) + 1$. Следовательно, для решения задачи о ННМ в графе G достаточно решить её по отдельности для каждого из графов $G - a$ и $G - N(a)$, а затем выбрать из двух независимых множеств то, у которого мощность больше: $\alpha(G) = \max\{\alpha(G - a), \alpha(G[\overline{N}(a)]) + 1\}$.

В каждом из графов $G - a$ и $G[\overline{N}(a)]$ число вершин меньше, чем в G. Это приводит к рекурсивному алгоритму нахождения ННМ графа G и его числа независимости $\alpha(G)$. Для произвольного графа G верхняя оценка сложности такого алгоритма может быть экспоненциальной, но если $G \in \mathrm{Free}(F + O_s)$, то легко установить наличие полиномиальной верхней оценки.

Действительно, для каждого связного графа G и любой его неизолированной вершины a имеем

$$t_n(G) \leqslant t_{n-1}(G - a) + t_{n-1-\deg a}(G[\overline{N}(a)]).$$

Пусть сначала $s = 1$. Используем условие $G \in \mathrm{Free}(F + O_1)$. В силу свойства наследственности $G - a \in \mathrm{Free}(F + O_1)$, а граф $G[\overline{N}(a)]$

удовлетворяет более сильному требованию: $G[\overline{N}(a)] \in \mathrm{Free}(F)$. В самом деле, если предположить, что в графе $G[\overline{N}(a)]$ имеется порождённый подграф, изоморфный F, то вместе с вершиной a он бы давал порождённый подграф $F + O_1$, что невозможно. Следовательно,

$$t_n(G) \leqslant t_{n-1}(\mathrm{Free}(F + O_1)) + t_{n-2}(\mathrm{Free}(F)) \leqslant t_{n-1}(\mathrm{Free}(F + O_1)) + O(n^k).$$

Итерируя это неравенство по n, получаем $t_n(\mathrm{Free}(F + O_1)) = O(n^{k+1})$.

Далее доказательство легко завершить, примененяя индукцию по s. Таким образом, $t_n(\mathrm{Free}(F + O_s)) = O(n^{k+s})$. Теорема 1 доказана.

2. Графы с логарифмическими ограничениями из класса $\mathrm{Free}(F + P_2)$

В этом и следующем разделах для фиксированного триодного графа F рассмотрим наследственный класс графов, свободных от $F + P_2$. Если предполагать, что класс графов $\mathrm{Free}(F)$ HM-простой, то в идеале из этого предположения хотелось бы вывести результат о HM-простоте класса $\mathrm{Free}(F+P_2)$ подобно тому, как это доказано в теореме 1 для класса $\mathrm{Free}(F + O_s)$. Однако, к сожалению, очевидного обобщения доказательства теоремы 1 на класс графов $\mathrm{Free}(F + P_2)$ получить пока не удалось, ввиду чего высказанное предположение остаётся под вопросом и пока является в общем случае нерешённой проблемой. В попытке получить желаемый результат хотя бы для какого-нибудь нетривиально устроенного подмножества графов из класса $\mathrm{Free}(F+P_2)$ введём дополнительные ограничения на степени или антистепени вершин входного графа, причём эти ограничения будут логарифмическими функциями от размера графа.

Помимо общеизвестных понятий степени и антистепени вершины нам понадобится здесь ещё одно новое понятие — относительной степени вершины. *Относительной степенью* вершины a в графе G, которую обозначим через $\deg(a \,|\, N(a))$, назовём максимальное число смежных с a вершин этого графа таких, что все выбранные вершины несмежны с одной из них:

$$\deg(a \,|\, N(a)) = \max_{x \in N(a)} |N(a) \cap \overline{N}(x)|.$$

Будет доказано, что подмножество графов из класса $\mathrm{Free}(F+P_2)$, у которых каждая вершина имеет либо степень, либо относительную степень, либо антистепень, ограниченную с точностью до мультипликативной константы логарифмом от числа вершин графа, образует HM-простой класс.

Выпишем формальное определение графа с логарифмическими ограничениями. *Графом с логарифмическими ограничениями* назовём граф, для каждой вершины a которого выполняется хотя бы одно из следующих трёх условий:

(a) $|N(a)| = \deg a \leqslant p \log_2 n$;
(b) $\deg(a \mid N(a)) \leqslant (p-1)\log_2 n$;
(c) $|\overline{N}(a)| = \overline{\deg}\, a \leqslant (k+p)\log_2 n$.

Здесь p и k — фиксированные натуральные константы.

Теорема 2. *Пусть F — произвольный граф с триодными компонентами такой, что задача о* НИМ *в классе* $\mathrm{Free}(F)$ *полиномиально разрешима и сложность её решения в этом классе равна $t_n(\mathrm{Free}(F)) = O(n^k)$, где k — некоторая натуральная константа. Пусть $G \in \mathrm{Free}(F + P_2)$ — n-вершинный граф с логарифмическими ограничениями для k и некоторого натурального p. Тогда задача о* НИМ *для графа G полиномиально разрешима:* НИМ *для графа G можно найти за время $t_n(G) = O(n^{k+p+2})$.*

Доказательство. Как и при доказательстве теоремы 1, без ограничения общности считаем, что граф G связен, поэтому снова для любой его неизолированной вершины a воспользуемся доказанным в теореме 1 неравенством

$$t_n(G) \leqslant t_{n-1}(G-a) + t_{n-1-\deg a}(G[\overline{N}(a)]).$$

Произвольным образом выберем вершину a графа G. В силу того, что для графа выполняются логарифмические ограничения, имеет место хотя бы один из трёх рассматриваемых ниже случаев.

Случай 1: $|\overline{N}(a)| = \overline{\deg}\, a \leqslant (k+p)\log_2 n$. Так как антистепень вершины a ограничена сверху логарифмом от числа вершин графа G, НИМ графа $G[\overline{N}(a)]$ можно найти простым перебором всех подмножеств его вершин. Среди таких подмножеств рассмотрим только независимые и выберем наибольшее из них. Сложность такого перебора и проверок независимости подмножеств не превосходит $2^{\overline{\deg} a}(\overline{\deg} a)^2$, откуда получаем

$$t_n(G) \leqslant t_{n-1}(G-a) + t_{\overline{\deg} a}(G[\overline{N}(a)]) \leqslant t_{n-1}(G-a) + \\ + 2^{(k+p)\log_2 n}((k+p)\log_2 n)^2 \leqslant t_{n-1}(G-a) + n^{k+p+1}.$$

Случай 2: $|N(a)| = \deg a \leqslant p \log_2 n$. Так как степень вершины a ограничена сверху логарифмом от числа вершин графа G, НИМ графа $G-a$ можно найти, сочетая идею переборного метода с полиномиальным алгоритмом для некоторых порождённых подграфов графа $G[\overline{N}(a)]$, принадлежащих классу $\mathrm{Free}(F)$.

Действительно, переберём все непустые подмножества множества вершин $N(a)$ (их число не превосходит $2^{\deg a}$). Далее проверим, будет ли каждое из них независимым (число операций для проверки независимости каждого такого множества не превосходит $\deg^2(a)$). Затем каждое такое независимое множество M дополним наибольшим независимым множеством в графе $G[\overline{N}(M) \cap \overline{N}(a)]$. После этого среди всех полученных

независимых множеств выберем наибольшее, оно и будет ННМ в графе $G - a$ при условии, что $\alpha(G - a) > \alpha(G[\overline{N}(a)])$. В противном случае $\alpha(G - a) = \alpha(G[\overline{N}(a)])$, и существует ННМ графа $G - a$, совпадающее с ННМ графа $G[\overline{N}(a)]$.

При этом важно то, что для каждого из указанных независимых множеств M граф $G[\overline{N}(M) \cap \overline{N}(a)]$ обязательно принадлежит классу $\mathrm{Free}(F)$. В самом деле, если предположить, что для какого-нибудь M в графе $G[\overline{N}(M) \cap \overline{N}(a)]$ есть порождённый подграф, изоморфный графу F, то вместе с вершиной a и любой одной вершиной из M в графе G существовал бы порождённый подграф $F + P_2$; противоречие. Отсюда

$$\begin{aligned} t_n(G) &\leqslant t_{n-1}(G - a) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant \\ &\leqslant 2^{\deg a} \deg^2(a) t_{\overline{\deg a}}(\mathrm{Free}(F)) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant \\ &\leqslant 2^{p \log_2 n} (p \log_2 n)^2 O(n^k) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant \\ &\leqslant t_{n-1-\deg a}(G[\overline{N}(a)]) + O(n^{k+p+1}). \end{aligned}$$

Случай 3: $\deg(a \,|\, N(a)) \leqslant (p-1) \log_2 n$. Так как относительная степень вершины a ограничена сверху логарифмом от числа вершин графа G, как и в случае 2, ННМ графа $G - a$ можно найти, сочетая идею переборного метода с полиномиальным алгоритмом для некоторых порождённых подграфов графа $G[\overline{N}(a)]$, принадлежащих классу $\mathrm{Free}(F)$. Только при этом незначительно изменится обоснование полиномиальной оценки сложности отыскания ННМ в графе $G - a$.

В самом деле, в случае 3 для отыскания ННМ графа $G - a$ можно перебрать вершины $x \in N(a)$, для каждой из которых найти все независимые множества $M \subseteq N(a) \cap \overline{N}(x)$ (по условию число таких множеств не превосходит $2^{\deg(a \,|\, N(a))}$). Каждое найденное независимое множество $M \cup \{x\}$ нужно дополнить наибольшим независимым множеством в графе $G[\overline{N}(M \cup \{x\}) \cap \overline{N}(a)]$. После этого остаётся среди всех полученных независимых множеств выбрать наибольшее, оно и будет ННМ в графе $G - a$ при условии, что $\alpha(G - a) > \alpha(G[\overline{N}(a)])$. В противном случае $\alpha(G - a) = \alpha(G[\overline{N}(a)])$, и существует ННМ графа $G - a$, совпадающее с ННМ графа $G[\overline{N}(a)]$.

Обоснование того, что для каждого из перебираемых независимых множеств $M \cup \{x\}$ граф $G[\overline{N}(M \cup \{x\}) \cap \overline{N}(a)]$ обязательно принадлежит классу $\mathrm{Free}(F)$, остаётся таким же, как аналогичное обоснование в случае 2. Следовательно, в случае 3 получаем

$$\begin{aligned} t_n(G) &\leqslant t_{n-1}(G - a) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant \\ &\leqslant n 2^{\deg(a \,|\, N(a))} \deg^2(a \,|\, N(a)) t_{\overline{\deg a}}(\mathrm{Free}(F)) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant \end{aligned}$$

$$\leqslant n2^{(p-1)\log_2 n}((p-1)\log_2 n)^2 O(n^k) + t_{n-1-\deg a}(G[\overline{N}(a)]) \leqslant$$
$$\leqslant t_{n-1-\deg a}(G[\overline{N}(a)]) + O(n^{k+p+1}).$$

Таким образом, в каждом из рассмотренных трёх случаев получен результат следующего вида: сложность решения задачи о НИМ в графе $G \in \mathrm{Free}(F + P_2)$ с логарифмическими ограничениями удовлетворяет неравенству

$$t_n(G) \leqslant t_{n-1}(G') + O(n^{k+p+1}).$$

Здесь G' — некоторый порождённый подграф графа G с числом вершин, не превосходящим $n-1$. Ясно, что для каждой вершины графа G' выполняется хотя бы одно из логарифмических ограничений (a), (b) или (c), где логарифм в каждом из ограничений для вершин из G' берётся от числа вершин n исходного графа G.

Итерирование полученного неравенства по n приводит к требуемой верхней оценке сложности $t_n(G) = O(n^{k+p+2})$. Теорема 2 доказана.

3. Графы с квазилогарифмическими ограничениями из класса $\mathrm{Free}(F + P_2)$

По-видимому, утверждение, доказанное в теореме 2, можно считать отправной точкой в предполагаемой серии результатов о нетривиальных НМ-простых подклассах класса графов $\mathrm{Free}(F + P_2)$. В этом разделе рассмотрим нетривиальное расширение множества всех графов с логарифмическими ограничениями из класса $\mathrm{Free}(F + P_2)$ и докажем НМ-простоту для такого расширения.

Графом с квазилогарифмическими ограничениями назовём граф, для каждой вершины a которого выполняется хотя бы одно из следующих трёх условий:

(a) $|N(a)| = \deg a \leqslant p\log_2 n$;
(b) $\deg(a \,|\, N(a)) \leqslant (p-1)\log_2 n$;
(d) $|\overline{N}(a)| = \overline{\deg}\, a \leqslant (k+p+1)(\ln n)(p\log_2\log_2 n)^{1-\varepsilon}$.

Здесь p и k — фиксированные натуральные константы, а $\varepsilon > 0$ — сколь угодно малая константа.

Очевидно, что множество графов с квазилогарифмическими ограничениями является расширением множества всех графов с логарифмическими ограничениями и получается за счёт замены условия (c) более слабым ограничением (d) на антистепени вершин. Докажем НМ-простоту множества всех графов с квазилогарифмическими ограничениями из класса $\mathrm{Free}(F + P_2)$.

Введём одно вспомогательное понятие, связанное с числовыми последовательностями определённого вида. Бесконечную монотонно неубывающую неотрицательную числовую последовательность $\{l_n\}_{n\geqslant 0}$ назовём

сублинейной последовательностью, если $l_n = o(n)$, т. е. если она растёт медленнее последовательности натуральных чисел. Сублинейную последовательность назовём *типичной*, если при любом натуральном n и при всех таких i, что $n - l_n \leqslant i \leqslant n$, выполняются соотношения $l_i = l_{n+1} - o(1)$. Смысл понятия типичной сублинейной последовательности заключается в том, что для всех номеров i из диапазона $n - l_n \leqslant i \leqslant n + 1$ разница между любыми двумя элементами данной последовательности с номерами из этого диапазона ничтожно мала (стремится к нулю при n, стремящемся к бесконечности) и ей фактически можно пренебречь, хотя при этом сама последовательность может быть расходящейся.

Стоит отметить, что многие хорошо известные примеры сублинейных числовых последовательностей, которые построены на основе соответствующих классических сублинейных функций, являются примерами типичных сублинейных последовательностей. Так, к числу типичных относятся стандартная логарифмическая последовательность $l_n = p \log_2 n$ при любом $p > 0$ и стандартная степенная последовательность $l_n = n^\varepsilon$ при любом ε таком, что $0 < \varepsilon < \frac{1}{2}$. Действительно, для логарифмической последовательности $l_n = p \log_2 n$ при $n \to \infty$ справедливо

$$l_{n+1} - l_{n-l_n} = p \log_2(n+1) - p \log_2(n - p \log_2 n) = p \log_2 \frac{n+1}{n - p \log_2 n} \to 0.$$

Аналогично для степенной последовательности $l_n = n^\varepsilon$ выполняется

$$\begin{aligned} l_{n+1} - l_{n-l_n} &= (n+1)^\varepsilon - (n - n^\varepsilon)^\varepsilon = \\ &= n^\varepsilon \left(1 + \frac{1}{n}\right)^\varepsilon - n^\varepsilon \left(1 - \frac{1}{n^{1-\varepsilon}}\right)^\varepsilon = \frac{(1+n^{-1})^\varepsilon - (1 - n^{\varepsilon-1})^\varepsilon}{n^{-\varepsilon}} \sim \\ &\sim \frac{\varepsilon\,(1+n^{-1})^{\varepsilon-1} \cdot (-n^{-2}) - \varepsilon(1-n^{\varepsilon-1})^{\varepsilon-1} \cdot (1-\varepsilon) n^{\varepsilon-2}}{(-\varepsilon) \cdot n^{-\varepsilon-1}} = \\ &= \left(1 + \frac{1}{n}\right)^{\varepsilon-1} \cdot \frac{1}{n^{1-\varepsilon}} + (1-\varepsilon) \cdot \left(1 - \frac{1}{n^{1-\varepsilon}}\right)^{\varepsilon-1} \cdot \frac{1}{n^{1-2\varepsilon}} \to 0 \end{aligned}$$

при $n \to \infty$ и любом $\varepsilon \in \left(0, \frac{1}{2}\right)$. Во втором примере при переходе к эквивалентной последовательности мы воспользовались правилом Лопиталя.

Из второго примера видно, что не все сублинейные числовые последовательности типичные: ясно, что простейшими из контрпримеров являются степенные последовательности $l_n = n^\varepsilon$ при любом $\varepsilon \in \left[\frac{1}{2}, 1\right)$. Также можно привести контрпримеры, имеющие довольно искусственный характер построения, что, однако, не умаляет факта их существования. Так, примером нетипичной сублинейной последовательности является последовательность вида

$$l_n = 2^k \quad \text{при } n = 10^k, 10^k + 1, \ldots, 10^{k+1} - 1,\ k \in \mathbb{N}_0.$$

В самом деле, нетрудно видеть, что при любых k и $n = 10^k - 1$ условие близости нарушается даже для некоторых соседних элементов, а именно при $n \to \infty$ имеем

$$l_{n+1} - l_n = l_{10^k} - l_{10^k-1} = 2^k - 2^{k-1} = 2^{k-1} = \\ = 2^{\lg(n+1)-1} = \frac{1}{2}(n+1)^{\lg 2} \not\to 0.$$

Далее нас будут интересовать исключительно типичные сублинейные последовательности. Более того, конкретное применение при выводе основного результата этого раздела найдёт только последовательность вида $l_n = p \log_2 n$. Сначала докажем две вспомогательные леммы.

Лемма 1. *Пусть имеется бесконечная монотонно неубывающая неотрицательная числовая последовательность $\{f_n\}_{n\geqslant 0}$, каждый элемент которой удовлетворяет рекуррентному неравенству*

$$f_{n+1} \leqslant f_n + f_{n-l_n} \quad n \geqslant l_n,$$

где $\{l_n\}_{n\geqslant 0}$ — любая типичная монотонно неубывающая неотрицательная сублинейная числовая последовательность. Тогда при любом сколь угодно малом $\varepsilon \in \left(0, \frac{1}{4}\right]$ и любом $l_n \geqslant 2^{1/\varepsilon^2}$ имеет место верхняя оценка

$$f_n = O\big(\big(1 + l_n^{\varepsilon-1}\big)^n\big), \quad n \in \mathbb{N}_0.$$

Доказательство. Индукция по n:

$$f_{n+1} \leqslant f_n + f_{n-l_n} = O\big(\big(1 + l_n^{\varepsilon-1}\big)^n + \big(1 + l_{n-l_n}^{\varepsilon-1}\big)^{n-l_n}\big).$$

По условию последовательность l_n является типичной сублинейной последовательностью, поэтому, во-первых, $l_n = o(n)$, а во-вторых, при любом натуральном n и при всех i таких, что $n - l_n \leqslant i \leqslant n$, справедливы равенства $l_i = l_{n+1} - o(1)$. Отсюда, опуская для удобства индекс $n+1$ в обозначении l_{n+1} (т. е. вместо l_{n+1} будем записывать только l), при каждом натуральном n получаем

$$\big(1 + l_n^{\varepsilon-1}\big)^n + \big(1 + l_{n-l_n}^{\varepsilon-1}\big)^{n-l_n} = O\big(\big(1 + l^{\varepsilon-1}\big)^n + \big(1 + l^{\varepsilon-1}\big)^{n-l}\big).$$

Далее нетрудно проверить, что справедлива цепочка соотношений

$$(1 + l^{\varepsilon-1})^n + (1 + l^{\varepsilon-1})^{n-l} = \\ = (1 + l^{\varepsilon-1})^{n+1} - (1 + l^{\varepsilon-1})^{n-l}(l^{\varepsilon-1}((1 + l^{\varepsilon-1})^{l^{1-\varepsilon}})^{l^\varepsilon} - 1) \leqslant \\ \leqslant (1 + l^{\varepsilon-1})^{n+1} - (1 + l^{\varepsilon-1})^{n-l}(l^{\varepsilon-1}\, 2^{l^\varepsilon} - 1) \leqslant (1 + l^{\varepsilon-1})^{n+1},$$

где использовано известное неравенство $(1 + 1/x)^x \geqslant 2$ при $x = l^{1-\varepsilon} \geqslant 1$ и неравенство $l^{\varepsilon-1}\, 2^{l^\varepsilon} \geqslant 1$, которое, как нетрудно проверить, выполняется при всех $l \geqslant 2^{1/\varepsilon^2}$, если $0 < \varepsilon \leqslant \frac{1}{4}$. Таким образом, из предположения,

что $f_n = O\big(\big(1 + l_n^{\varepsilon-1}\big)^n\big)$, получаем $f_{n+1} = O\big(\big(1 + l_{n+1}^{\varepsilon-1}\big)^{n+1}\big)$. Лемма 1 доказана.

Следствие 1. *В условиях леммы* 1 *имеет место верхняя оценка* $f_n = O(e^{n\, l_n^{\varepsilon-1}})$.

Доказательство вытекает непосредственно из леммы 1 и известного неравенства $(1 + 1/x)^x \leqslant e$ при $x = l_n^{1-\varepsilon}$. Следствие 1 доказано.

Лемма 2. *Пусть* p *и* k *— натуральные константы, а* F *— граф с трио́дными компонентами, для которого задача о* НИМ *в классе* $\mathrm{Free}(F)$ НМ-*проста и* $t_n(\mathrm{Free}(F)) = O(n^k)$. *Тогда при любом сколь угодно малом* $\varepsilon \in \big(0, \frac{1}{4}\big]$ *для любого графа* $H \in \mathrm{Free}(F + P_2)$, *имеющего* $m + 1$ *вершин, при всех* $m \geqslant 2^{(1/p)\cdot 2^{1/\varepsilon^2}}$ *верна следующая верхняя оценка сложности решения задачи о* НИМ*:*

$$t_{m+1}(H) = O(e^{m(p\log_2 m)^{\varepsilon-1}}).$$

Доказательство. Если хотя бы для одной вершины a графа H выполняется хотя бы одно из логарифмических ограничений (a), (b) или (c), то, поступая так же, как при доказательстве теоремы 2, полиномиально сведём задачу о НИМ графа H к аналогичной задаче для некоторого его порождённого подграфа $H' \in \mathrm{Free}(F + P_2)$ с меньшим числом вершин. Продолжим выполнять такие действия до тех пор, пока на некотором шаге сведе́ния не окажется, что все вершины графа, являющегося результатом сведе́ния, перестанут удовлетворять каждому из трёх логарифмических ограничений (a), (b) и (c). Другими словами, на некотором шаге сведе́ния может оказаться, что у некоторого порождённого подграфа H' графа H значения степеней, относительных степеней и антистепеней всех вершин ограничены снизу логарифмом от числа вершин $m + 1$ исходного графа H (с точностью до мультипликативных констант). Если такого шага не будет, то теорема 2 гарантирует полиномиальную оценку сложности $t_{m+1}(H)$ решения задачи о НИМ в графе H.

Рассмотрим случай, когда такой шаг всё-таки есть. Не ограничивая общности, можно считать, что это уже самый первый шаг, иначе выполним полиномиальное сведе́ние к графу с меньшим числом вершин. Итак, пусть каждая вершина a графа H не удовлетворяет ни одному из логарифмических (относительно числа вершин $m + 1$) ограничений (a), (b) и (c). В силу невыполнения ограничения (a) для каждой вершины a графа H имеет место неравенство $\deg a > p\log_2 m$, из которого следует, что антистепень вершины a ограничена сверху: $\overline{\deg}\, a \leqslant m - p\log_2 m$. Это означает, что сложность $t_{m+1}(H)$ решения задачи о НИМ в $(m + 1)$-вершинном графе $H \in \mathrm{Free}(F + P_2)$ удовлетворяет неравенству

$$t_{m+1}(H) \leqslant t_m(H-a) + t_{\overline{\deg a}}(H[\overline{N}(a)]) \leqslant \\ \leqslant t_m(\mathrm{Free}(F+P_2)) + t_{m-p\log_2 m}(\mathrm{Free}(F+P_2)).$$

Положим $f_m = t_m(\mathrm{Free}(F+P_2))$ и применим лемму 1 при $n=m$ и при $l_m = p\log_2 m \geqslant 2^{1/\varepsilon^2}$. Заметим, что это заведомо можно сделать при любом сколь угодно малом $\varepsilon \in \left(0, \frac{1}{4}\right]$ и любом $m \geqslant 2^{(1/p)\cdot 2^{1/\varepsilon^2}}$ (нижняя оценка на m является фиксированной константой). По следствию 1 при $l_m = p\log_2 m$ получаем, что

$$t_{m+1}(H) = O(e^{m(p\log_2 m)^{\varepsilon-1}})$$

при любом сколь угодно малом $\varepsilon \in \left(0, \frac{1}{4}\right]$ и любом $m \geqslant 2^{(1/p)\cdot 2^{1/\varepsilon^2}}$. Лемма 2 доказана.

Следствие 2. *Пусть p и k — натуральные константы, а F — граф с триодными компонентами, для которого задача о* НИМ *в классе* $\mathrm{Free}(F)$ НМ-*проста и* $t_n(\mathrm{Free}(F)) = O(n^k)$. *Пусть G — любой n-вершинный граф из класса* $\mathrm{Free}(F+P_2)$, *а H — любой его $(m+1)$-вершинный порождённый подграф, для числа вершин которого выполняются неравенства*

$$2^{(1/p)\cdot 2^{1/\varepsilon^2}} \leqslant m \leqslant (k+p+1)(p\log_2\log_2 n)^{1-\varepsilon}\ln n,$$

где $\varepsilon \in \left(0, \frac{1}{4}\right]$ — сколь угодно малое число. Тогда сложность $t_{m+1}(H)$ решения задачи о НИМ *в графе H ограничена сверху полиномом от числа вершин n графа G, а именно $t_{m+1}(H) = O(n^{k+p+1})$.*

Доказательство. В справедливости следствия 2 нетрудно убедиться непосредственной подстановкой максимально возможного (в условиях этого следствия) значения $m = (k+p+1)(p\log_2\log_2 n)^{1-\varepsilon}\ln n$ в правую часть равенства

$$t_{m+1}(H) = O(e^{m(p\log_2 m)^{\varepsilon-1}}),$$

доказанного в лемме 2. Следствие 2 доказано.

Основным результатом этого раздела служит

Теорема 3. *Пусть F — произвольный граф с триодными компонентами такой, что задача о* НИМ *в классе* $\mathrm{Free}(F)$ *полиномиально разрешима, и сложность решения в этом классе равна* $t_n(\mathrm{Free}(F)) = O(n^k)$, *где k — некоторая натуральная константа. Пусть $G \in \mathrm{Free}(F+P_2)$ — n-вершинный граф с квазилогарифмическими ограничениями для k, некоторого натурального p и сколь угодно малого $\varepsilon \in \left(0, \frac{1}{4}\right]$. Тогда задача о* НИМ *для графа G полиномиально разрешима:* НИМ *для графа G можно найти за время* $t_n(G) = O(n^{k+p+2})$.

Доказательство. Воспользуемся той же идеей, что и при доказательстве теоремы 2. Отталкиваясь от справедливого для любого графа G неравенства

$$t_n(G) \leqslant t_{n-1}(G-a) + t_{\overline{\deg a}}(G[\overline{N}(a)]),$$

рассмотрим три случая, соответствующих квазилогарифмическим ограничениям на степень, относительную степень и антистепень произвольно выбранной в G неизолированной вершины a.

Обоснования полиномиальных оценок сложности в случаях 2 и 3 остаются такими же, как и в доказательстве теоремы 2, поскольку эти случаи соответствуют логарифмическим ограничениям (a) и (b).

Осталось рассмотреть модификацию случая 1, когда логарифмическое ограничение (c) на антистепень вершины a заменяется квазилогарифмическим ограничением (d). Однако в этом случае число вершин графа $G[\overline{N}(a)]$ удовлетворяет всем ограничениям, указанным в лемме 2 и следствии 2. Тем самым в силу следствия 2 задача о ННМ в графе $G[\overline{N}(a)]$ полиномиально разрешима:

$$t_{\overline{\deg a}}(G[\overline{N}(a)]) = O(n^{k+p+1}).$$

Отсюда в модифицированном случае 1 (т. е. для квазилогарифмического ограничения на антистепень вершины a) имеем неравенство

$$t_n(G) \leqslant t_{n-1}(G-a) + O(n^{k+p+1}),$$

аналогичное полученным в случаях 2 и 3 при доказательстве теоремы 2.

В завершение доказательства теоремы 3 объединяем все три случая и замечаем, что сложность решения задачи о ННМ для любого входного графа $G \in \mathrm{Free}(F + P_2)$ с квазилогарифмическими ограничениями удовлетворяет неравенству

$$t_n(G) \leqslant t_{n-1}(G') + O(n^{k+p+1}),$$

в котором G' — некоторый порождённый подграф графа G с числом вершин, не превосходящим $n-1$, для каждой вершины которого, очевидно, выполняется хотя бы одно из квазилогарифмических ограничений (a), (b) или (d), где логарифм в каждом из ограничений для вершин из G' берётся от числа вершин n исходного графа G.

Итерирование полученного неравенства по n приводит к требуемой верхней оценке сложности $t_n(G) = O(n^{k+p+2})$. Теорема 3 доказана.

Заключение

В статье введены понятия графов с логарифмическими ограничениями и графов с квазилогарифмическими ограничениями. Доказано, что

для каждого графа F с триодными компонентами из НМ-простоты класса графов Free(F) следует НМ-простота множества графов с логарифмическими ограничениями из класса Free($F+P_2$) и НМ-простота множества графов с квазилогарифмическими ограничениями из класса Free($F+P_2$).

По всей видимости, результаты, представленные в данной статье, можно считать отправной точкой исследований, связанных с обнаружением нетривиальных НМ-простых множеств графов, являющихся подмножествами класса графов Free($F+P_2$) в случае НМ-простого класса Free(F).

Перспективы дальнейших исследований такого типа можно связывать с надеждой на получение аналогичных результатов для подмножеств графов из класса Free($F+P_2$) с более слабыми (по сравнению с логарифмической функцией) типами ограничений либо на степени, либо на относительные степени, либо на антистепени вершин. Конечно, с этой точки зрения вслед за логарифмической функцией наиболее пристальное внимание привлекают, в первую очередь, степенные функции вида n^{ε}. Подобно рассмотренной в этой статье логарифмической функции такие степенные функции также являются примерами типичных сублинейных функций при любом ε таком, что $0 < \varepsilon < \frac{1}{2}$, а потому их вполне можно рассматривать в качестве верхних границ значений указанных трёх численных характеристик вершин графа, пусть даже при очень малых значениях положительной действительной константы $\varepsilon < \frac{1}{2}$.

К сожалению, в настоящий момент непонятно, каким образом можно получить обобщения результатов, доказанных для множеств графов с логарифмическими и квазилогарифмическими ограничениями из класса Free($F+P_2$), на множества графов со степенными ограничениями вида n^{ε} даже при сколь угодно малом значении положительной константы $\varepsilon < \frac{1}{2}$. В самом деле, из результатов, доказанных в статье, для случая степенных ограничений верно только утверждение леммы 1, но из него очевидным образом не выводится полиномиальная оценка сложности решения задачи о ННМ для графов из класса Free($F+P_2$) со степенными ограничениями. Говоря точнее, оценка сложности для графов со степенными ограничениями, которая напрямую выводится для них из леммы 1, оказывается уже экспоненциальной в отличие от полиномиальной оценки, полученной в следствии 2 для графов с квазилогарифмическими ограничениями.

Тем не менее, можно надеяться на то, что более детальное изучение случаев других типичных сублинейных ограничений на численные характеристики вершин графов из класса Free($F+P_2$), по всей видимости, может способствовать расширению горизонтов и перспектив для дальнейших исследований.

Финансирование работы

Исследование выполнено за счёт бюджета Нижегородского гос. университета им. Н. И. Лобачевского. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Алексеев В. Е., Коробицын Д. В.** О сложности некоторых задач на наследственных классах графов // Дискрет. математика. 1992. Т. 4, № 4. С. 34–40.
2. **Алексеев В. Е.** О влиянии локальных ограничений на сложность определения числа независимости графа // Комбинаторно-алгебраические методы в прикладной математике. Горький: Изд-во Горьк. ун-та, 1982. С. 3–13.
3. **Алексеев В. Е.** О числе тупиковых независимых множеств в графах из наследственных классов // Комбинаторно-алгебраические методы в дискретной оптимизации. Н. Новгород: Изд-во ННГУ, 1991. С. 5–8.
4. **Minty G.** On maximal independent sets of vertices in claw-free graphs // J. Comb. Theory, Ser. B. 1980. V. 28, No. 3. P. 284–304.
5. **Алексеев В. Е.** Полиномиальный алгоритм для нахождения наибольших независимых множеств в графах без вилок // Дискрет. анализ и исслед. операций. Сер. 1. 1999. Т. 6, № 4. С. 3–19.
6. **Lozin V. V., Mosca R.** Independent sets in extension of $2K_2$-free graphs // Discrete Appl. Math. 2005. V. 146. P. 74–80.
7. **Lokshantov D., Vatshelle M., Villanger Y.** Independent set in P_5-free graphs in polynomial time // Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms (Portland, OR, USA, Jan. 5–7, 2014). Philadelphia, PA: SIAM, 2014. P. 570–581.
8. **Grzesik A., Klimošová T., Pilipczuk Mar., Pilipczuk Mic.** Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. Ithaca, NY, 2017. (e-Print Archive / Cornell Univ.; arXiv:1707.05491).
9. **Karthick T., Maffray F.** Weighted independent sets in classes of P_6-free graphs // Discrete Appl. Math. 2016. V. 209. P. 217–226.
10. **Lozin V. V., Monnot J., Ries B.** On the maximum independent set problem in subclasses of subcubic graphs // J. Discrete Algorithms. 2015. V. 31. P. 104–112.
11. **Lozin V. V., Rautenbach D.** Some results on graphs without long induced paths // Inf. Process. Lett. 2003. V. 88. P. 167–171.
12. **Малышев Д. С., Сироткин Д. В.** Полиномиальная разрешимость задачи о независимом множестве в одном классе субкубических планарных графов // Дискрет. анализ и исслед. операций. 2017. Т. 24, № 3. С. 35–60.

13. **Abrishami T., Chudnovsky M., Dibek C., Rzążewski P.** Polynomial-time algorithm for maximum independent set unbounded-degree graphs with no long induced claws // Proc. 2022 Annu. ACM-SIAM Symp. Discrete Algorithms (Alexandria, VA, USA, Jan. 9–12, 2022). Philadelphia, PA: SIAM, 2022. P. 1448–1470. DOI: 10.1137/1.9781611977073.61.
14. **Alekseev V. E., Lozin V. V., Malyshev D. S., Milanič M.** The maximum independent set problem in planar graphs // Mathematical foundations of computer science 2008. Proc. 33rd Int. Symp. (Toruń, Poland, Aug. 25–29, 2008). Heidelberg: Springer, 2008. P. 96–107. (Lect. Notes Comput. Sci.; V. 5162).
15. **Малышев Д. С.** Классы субкубических планарных графов, для которых задача о независимом множестве полиномиально разрешима // Дискрет. анализ и исслед. операций. 2013. V. 20, No. 3. P. 26–44.
16. **Алексеев В. Е., Сорочан С. В.** Новые случаи полиномиальной разрешимости задачи о независимом множестве для графов с запрещёнными путями // Дискрет. анализ и исслед. операций. 2018. Т. 25, № 2. С. 5–18.
17. **Сорочан С. В.** Новые случаи полиномиальной разрешимости задачи о независимом множестве для графов с запрещёнными триодами // Дискрет. анализ и исслед. операций. 2023. Т. 30, № 1. С. 85–109.
18. **Алексеев В. Е., Захарова Д. В.** Теория графов. Н. Новгород: Изд-во Нижегор. гос. ун-та, 2018. 118 с.
19. **Алексеев В. Е., Таланов В. А.** Графы. Модели вычислений. Алгоритмы. Н. Новгород: Изд-во Нижегор. гос. ун-та, 2005. 308 с.

Сорочан Сергей Владимирович

Статья поступила
17 июля 2025 г.
После доработки —
30 июля 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 191–212

UDC 519.178 DOI: 10.33048/daio.2025.32.839

POLYNOMIAL SOLVABILITY OF THE INDEPENDENT SET PROBLEM FOR SOME HEREDITARY CLASSES OF GRAPHS WITH LOGARITHMIC AND QUASI-LOGARITHMIC CONSTRAINTS ON VERTEX DEGREES OR ANTI-DEGREES

S. V. Sorochan

Lobachevsky State University of Nizhny Novgorod,
23 Gagarin Avenue, 603950 Nizhny Novgorod, Russia
E-mail: sergey.sorochan@itmm.unn.ru

Abstract. A triode is a tree with at most three leaves and at most one vertex of degree 3. The maximum independent set (MIS) problem is solvable in polynomial time for graphs that do not contain as a subgraph a graph whose each component is a triode with any fixed number of vertices. If a k-vertex induced subgraph with triode components is forbidden, then, for $k > 5$, the question about the solution complexity for this problem remains open. Let F be a graph whose each component is a triode, that is a triode graph. It is known that, if the MIS problem is polynomially solvable in the class of all graphs not containing F as an induced subgraph, then the problem is also polynomially solvable in the class of all graphs free of $F + O_s$. The graph $F + O_s$ is obtained from F by adding s isolated vertices. If we forbid $F + P_2$ as an induced subgraph and P_2 is a path with two vertices, then the polynomial solvability of the MIS problem in the class of all graphs free of $F + P_2$ is proved only for a very small number of triode graphs F while, for the vast majority of such graphs F, the complexity status of the MIS problem in the class of $(F + P_2)$-free graphs remains unknown.

In this article so-called logarithmic constraint graphs are considered. Each vertex in such a graph satisfies at least one of the following three conditions:

1) the degree of a vertex does not exceed (up to a multiplicative constant) the logarithm of the total number of vertices in the graph;

2) the relative degree of a vertex (that is the maximum number of vertices adjacent to it such that all these adjacent vertices are not adjacent to one of them) does not exceed (up to a multiplicative constant) the logarithm of the total number of vertices in the graph;

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

3) the antidegree of a vertex (that is the number of vertices that are not adjacent to it) does not exceed (up to a multiplicative constant) the logarithm of the total number of vertices in the graph.

It is proved that, for any triode graph F such that the MIS problem is polynomially solvable in the class of F-free graphs, it remains polynomially solvable for all $(F + P_2)$-free graphs with logarithmic constraints. We also consider one non-trivial extension of the latter graph set that consists of graphs with quasi-logarithmic constraints. A similar result is proved for all such $(F + P_2)$-free graphs. We propose polynomial algorithms to solve the MIS problem for graphs with logarithmic and quasi-logarithmic constraints which are free of $F + P_2$, and upper bounds for the algorithm complexity are found. Bibliogr. 19.

Keywords: independent set, monotonic class, hereditary class, IS-easy class, IS-hard class, forbidden subgraph, triode, logarithmic constraint graphs, quasi-logarithmic constraint graphs, polynomial algorithm.

References

1. **V. E. Alekseev** and **D. V. Korobitsyn,** On the complexity of some problems on hereditary classes of graphs, *Diskretn. Mat.* **4** (4), 34–40 (1992) [Russian].
2. **V. E. Alekseev,** On the influence of local constraints on the complexity of determining the independence number of a graph, in *Combinatorial and Algebraic Methods in Applied Mathematics* (Izd. Gork. Univ., Gorky, 1982), pp. 3–13 [Russian].
3. **V. E. Alekseev,** On the number of maximum independent sets in graphs of hereditary classes, in *Combinatorial and Algebraic Methods in Discrete Optimization* (Izd. NNGU, Nizh. Novgorod, 1991), pp. 5–8 [Russian].
4. **G. Minty,** On maximal independent sets of vertices in claw-free graphs, *J. Comb. Theory, Ser. B*, **28** (3), 284–304 (1980).
5. **V. E. Alekseev,** A polynomial algorithm for finding largest independent sets in fork-free graphs, *Diskretn. Anal. Issled. Oper., Ser. 1*, **6** (4), 3–19 (1999) [Russian] [*Discrete Appl. Math.* **135** (1–3), 3–16 (2004)].
6. **V. V. Lozin** and **R. Mosca,** Independent sets in extension of $2K_2$-free graphs, *Discrete Appl. Math.* **146**, 74–80 (2005).
7. **D. Lokshantov, M. Vatshelle,** and **Y. Villanger,** Independent set in P_5-free graphs in polynomial time, in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms* (Portland, OR, USA, Jan. 5–7, 2014) (SIAM, Philadelphia, PA, 2014), pp. 570–581.
8. **A. Grzesik, T. Klimošová, Mar. Pilipczuk,** and **Mic. Pilipczuk,** Polynomial-time algorithm for maximum weight independent set on P_6-free graphs (Ithaca, NY, 2017) (e-Print Archive / Cornell Univ., arXiv:1707.05491).
9. **T. Karthick** and **F. Maffray,** Weighted independent sets in classes of P_6-free graphs, *Discrete Appl. Math.* **209**, 217–226 (2016).

10. **V. V. Lozin, J. Monnot,** and **B. Ries,** On the maximum independent set problem in subclasses of subcubic graphs, *J. Discrete Algorithms* **31**, 104–112 (2015).
11. **V. V. Lozin** and **D. Rautenbach,** Some results on graphs without long induced paths, *Inf. Process. Lett.* **88**, 167–171 (2003).
12. **D. S. Malyshev** and **D. V. Sirotkin,** Polynomial-time solvability of the independent set problem in a certain class of subcubic planar graphs, *Diskretn. Anal. Issled. Oper.* **24** (3), 35–60 (2017) [Russian] [*J. Appl. Ind. Math.* **11** (3), 400–414 (2017), DOI:].
13. **T. Abrishami, M. Chudnovsky, C. Dibek,** and **P. Rzążewski,** Polynomial-time algorithm for maximum independent set unbounded-degree graphs with no long induced claws, in *Proc. 2022 Annu. ACM-SIAM Symp. Discrete Algorithms* (Alexandria, VA, USA, Jan. 9–12, 2022) (SIAM, Philadelphia, PA, 2022), pp. 1448–1470, DOI: 10.1137/1.9781611977073.61.
14. **V. E. Alekseev, V. V. Lozin, D. S. Malyshev,** and **M. Milanič,** The maximum independent set problem in planar graphs, in *Mathematical Foundations of Computer Science 2008*, Proc. 33rd Int. Symp. (Toruń, Poland, Aug. 25–29, 2008) (Springer, Heidelberg, 2008), pp. 96–107 (Lect. Notes Comput. Sci., V. 5162).
15. **D. S. Malyshev,** Classes of subcubic planar graphs for which the independent set problem is polynomially solvable, *Diskretn. Anal. Issled. Oper.* **20** (3), 26–44 (2013) [Russian] [*J. Appl. Ind. Math.* **7** (4), 537–548 (2013)].
16. **V. E. Alekseev** and **S. V. Sorochan,** New cases of the polynomial solvability of the independent set problem for graphs with forbidden paths, *Diskretn. Anal. Issled. Oper.* **25** (2), 5–18 (2018) [Russian] [*J. Appl. Ind. Math.* **12** (2) 213–219 (2018)].
17. **S. V. Sorochan,** New cases of polynomial solvability of the independent set problem for graphs with forbidden triodes, *Diskretn. Anal. Issled. Oper.* **30** (1), 85–109 (2023) [Russian] [*J. Appl. Ind. Math.* **17** (1) 185–198 (2023)].
18. **V. E. Alekseev** and **D. V. Zakharova,** *Graph Theory* (Izd. Nizhegor. Gos. Univ., Nizh. Novgorod, 2018) [Russian].
19. **V. E. Alekseev** and **V. A. Talanov,** *Graphs. Computing Models. Algorithms* (Izd. Nizhegor. Gos. Univ., Nizh. Novgorod, 2005) [Russian].

Sergey V. Sorochan

Received July 17, 2025
Revised July 30, 2025
Accepted September 22, 2025

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 213–230

УДК 519.713 DOI: 10.33048/daio.2025.32.822

ЗАДАЧИ БЕСКОНЕЧНОЙ РЕГУЛЯРНОЙ РЕАЛИЗУЕМОСТИ

И. Н. Шиманогов[1, a], *М. Н. Вялый*[1, 2, 3, b]

[1] Московский физико-технический институт
(национальный исследовательский университет)
ул. Керченская, 1А, корп. 1, 117303 Москва, Россия

[2] Федеральный исследовательский центр «Информатика и управление» РАН
ул. Вавилова, 44, корп. 2, 119333 Москва, Россия

[3] Национальный исследовательский университет «Высшая школа экономики»,
Покровский б-р, 11, 109028 Москва, Россия

E-mail: [a] shimanogov.in@phystech.edu, [b] vyalyi@gmail.com

Аннотация. Хорошо изученным классом алгоритмических задач являются задачи регулярной реализуемости — проверки непустоты пересечения регулярного языка с заданным языком. Такая задача имеет естественную алгебраическую интерпретацию — проверку принадлежности элемента булевой алгебры ядру определённого гомоморфизма. Это мотивирует рассмотрение аналогичной задачи бесконечной регулярной реализуемости — проверки бесконечности пересечения регулярного языка с заданным. В работе рассматриваются задачи регулярной реализуемости для разрешимых языков и приводится сравнительный анализ сложности задач бесконечной регулярной реализуемости и задач регулярной реализуемости. Библиогр. 22.

Ключевые слова: регулярный язык, задача регулярной реализуемости, арифметическая иерархия.

Введение

Одним из основных объектов изучения в теории формальных языков являются регулярные языки. Хорошо изученным классом задач являются задачи проверки выполнимости некоторого регулярного условия для слов фиксированного языка. Рассмотрим некоторый язык L. Входом задачи регулярной реализуемости является регулярный язык R, заданный, например, некоторым автоматом. Необходимо проверить непустоту пересечения L и R. Интенсивно изучался случай, когда L является

контекстно-свободным языком (см. [1–6]). Задачи регулярной реализуемости для контекстно-свободных языков оказываются важными для теории тонких сводимостей (fine-grained reductions) (см. [3, 6]). В [7] рассмотрены задачи регулярной реализуемости, связанные с комбинаторикой слов. Разрешимость задач регулярной реализуемости для широкого круга теоретико-графовых задач в подходящей кодировке графов установлена в работах [8–10]. С использованием аналогичной кодировки в [11] доказана разрешимость задач регулярной реализуемости, связанных с целочисленным программированием.

Общая задача регулярной реализуемости введена М. Вялым в 2009 г. Основной мотивацией была надежда найти ограничения на возможные уровни алгоритмической сложности по аналогии со знаменитой гипотезой CSP (для задач CSP выполняется дихотомия: либо задача принадлежит P, либо NP-полна; точную исходную формулировку см. в [12]). Позднее гипотеза CSP была доказана в 2017 г. независимо А. Булатовым [13] и Д. Жуком [14], подробное изложение этиих доказательств см. в [15, 16]. Однако для задач регулярной реализуемости ситуация принципиально иная. В 2013 г. М. Вялый доказал универсальность задач регулярной реализуемости относительно алгоритмической сложности с точностью до некоторых достаточно слабых сводимостей. Позднее была доказана универсальность нескольких ограниченных классов задач регулярной реализуемости (см. [17]). Там же можно найти обзор более ранних результатов про универсальность задач регулярной реализуемости. Заметим, что самый популярный случай задач регулярной реализуемости для контекстно-свободных языков остаётся открытым, причём результаты [2] показывают, что и в этом случае разнообразие возможных уровней алгоритмической сложности может быть весьма велико.

Каждому языку L соответствует булева алгебра относительно регулярных языков: данная алгебра образована всеми возможными пересечениями регулярных языков с L с операциями пересечения, объединения и дополнения до L. При этом существует естественный гомоморфизм из алгебры регулярных языков на относительно регулярную алгебру L: $\varphi(R) = R \cap L$. Таким образом, задача регулярной реализуемости является задачей проверки того, что образ данного элемента не равен нулю.

В данной работе рассматриваем следующую, схожую задачу — алгоритмическую проверку того, что образ данного элемента не лежит в идеале Фреше, т. е. не является конечным объединением атомов. Так как это равносильно бесконечности пересечения L и R, данные задачи названы задачами бесконечной регулярной реализуемости. В заключительной части работы более подробно обсуждаются интерпретации доказанных утверждений с точки зрения булевых алгебр.

В работе рассматриваются задачи регулярной реализуемости для разрешимых языков и приводится анализ алгоритмической сложности задачи бесконечной регулярной реализуемости. Доказывается существование разрешимых языков таких, что задача регулярной реализуемости разрешима, а бесконечной регулярной реализуемости — нет, и наоборот. Так же приводится пример языка, когда данные задачи полны в Σ_1 и Π_2 соответственно.

1. Кодировка машин Тьюринга

Так как множество машин Тьюринга (МТ) счётно, каждой из них возможно сопоставить некоторое натуральное число. Стандартная кодировка строится следующим образом: через разделитель записывается количество состояний в машине, размер её алфавита, номера особых состояний, особых символов и все переходы в виде кортежей из номеров элементов упомянутых множеств. Далее получившееся слово приводится, например, к алфавиту $\{0, 1\}$ и трактуется как натуральное число.

Нумерацией назовём функцию, сопоставляющую каждому натуральному числу некоторую МТ. В дальнейшем нам понадобится особая нумерация машин Тьюринга со следующим свойством: для любой бесконечной арифметической прогрессии в этой нумерации для любой машины Тьюринга в выбранной прогрессии имеется бесконечное множество чисел, которым сопоставлена указанная МТ. При этом необходимо, чтобы стандартная кодировка была вычислима по данному особому номеру.

Последовательно будем строить частично определённые функции f_i с конечными областями определения, сопоставляющие натуральным числам номера машин Тьюринга в стандартной кодировке.

Пронумеруем все бесконечные арифметические прогрессии вычислимым образом. Пусть отображение на функциях $A_i^j(f)$ определено следующим образом: выделим среди элементов, на которых не определена f, наименьший, принадлежащий прогрессии под номером i, и зададим на нём значение j. Через B_i^j обозначим следующую композицию отображений: $A_i^j \circ A_{i-1}^j \circ \cdots \circ A_1^j$. Таким образом, применение к функции f отображения B_i^j гарантирует, что для любой из первых i арифметических прогрессий найдётся число n из этой прогрессии такое, что $B_i^j[f](n) = j$.

Рассмотрим композицию $C_i = B_i^i \circ B_{i-1}^{i-1} \circ \cdots \circ B_1^1$. Применение к функции f отображения C_i гарантирует, что для любой из первых i арифметических прогрессий найдутся числа $n_1, \ldots, n_i$ из этой прогрессии такие, что $C_i[f](n_i) = i$. Будем считать, что f_0 нигде не определена. Положим $f_i = C_i(f_{i-1})$ и $f = \bigcup f_i$.

Заметим, что такая функция f вычислима и всюду определена. Действительно, $f(n)$ определяется за конечное число шагов C_i, поскольку A_i^j

всегда увеличивает область определения функции таким образом, что она остаётся начальным отрезком натуральных чисел. Так как A_i^j всегда только доопределяет функцию в точке, где она ещё не определена, определение f как функции корректно.

Теперь рассмотрим следующую нумерацию МТ: номер n соответствует МТ со стандартным номером $f(n)$. При этом для каждой арифметической прогрессии i и для каждого номера j операция A_i^j входит во все C_k, где $k = \max(i, j)$. Таким образом, в любой арифметической прогрессии для любой машины встречается бесконечно много чисел, нумерующих данную машину.

Всюду далее используется построенная кодировка.

2. Арифметическая иерархия

Будем рассматривать предикаты как функции в множество $\{0, 1\}$.

Определение 1. Язык L будем называть *разрешимым*, если предикат принадлежности данному языку вычислим. Обозначим множество разрешимых языков через $\Delta_0 = \Sigma_0 = \Pi_0$.

Определение 2. Для $n \geqslant 1$ будем говорить, что

- L принадлежит Σ_n, если

$$x \in L \Leftrightarrow (\exists y_1)(\forall y_2)(\exists y_3) \dots (Qy_n)\, R(x, y_1, y_2, \dots, y_n),$$

где Q — это $\exists$ для нечётного n и $\forall$ для чётного, а R — вычислимый предикат;

- L принадлежит Π_n, если

$$x \in L \Leftrightarrow (\forall y_1)(\exists y_2)(\forall y_3) \dots (Qy_n)\, R(x, y_1, y_2, \dots, y_n),$$

где Q — это $\forall$ для нечётного n и $\exists$ для чётного, а R — вычислимый предикат;

- L принадлежит Δ_n, если $L \in \Pi_n$ и $L \in \Sigma_n$, т. е. $\Delta_n = \Pi_n \cap \Sigma_n$.

Теорема 1 [18]. *Для классов Σ_n, Π_n, Δ_n выполняются следующие соотношения:*

1) $A \in \Sigma_n \Leftrightarrow \overline{A} \in \Pi_n$;
2) $A \in \Sigma_n \cup \Pi_n \Rightarrow (\forall m > n)\, A \in \Sigma_m \cap \Pi_m$;
3) $A, B \in \Sigma_n \Rightarrow A \cup B, A \cap B \in \Sigma_n$;
4) $A, B \in \Pi_n \Rightarrow A \cup B, A \cap B \in \Pi_n$;
5) $R \in \Sigma_n \wedge A = \{x \mid \exists y\colon (x, y) \in R\} \Rightarrow A \in \Sigma_n$.

Теорема 2 (теорема Поста (см. [18])).

$$\Delta_1 = \Delta_0.$$

Определение 3. Будем говорить, что язык A *сводится* к языку B, и используем обозначение $A \leqslant B$, если существует вычислимая функция f такая, что $x \in A$ тогда и только тогда, когда $f(x) \in B$.

Определение 4. Будем говорить, что язык A *труден в классе* Γ, если для любого $B \in \Gamma$ имеет место $B \leqslant A$. Если, кроме того, $A \in \Gamma$, то будем говорить, что A *полон в* Γ.

Лемма 1 (см. [18]). *Пусть* $A \leqslant B$. *Тогда*
1) *если* $B \in \Sigma_n$, *то* $A \in \Sigma_n$;
2) *если* $B \in \Pi_n$, *то* $A \in \Pi_n$;
3) *если* A *труден в* Σ_n, *то* B *труден в* Σ_n;
4) *если* A *труден в* Π_n, *то* B *труден в* Π_n.

Лемма 2 (см. [18]). *Имеют место следующие утверждения:*
1) *если* A *полон в* Σ_n, *то* $\overline{A}$ *полон в* Π_n;
2) *если* A *полон в* Π_n, *то* $\overline{A}$ *полон в* Σ_n.

Лемма 3. *Следующий язык разрешим:*

$$\texttt{Time} = \{(x, y, t) \mid \text{МТ } \textit{с номером } x \textit{ на слове } y \textit{ останавливается на шаге } t\}.$$

Доказательство. Достаточно воспользоваться конструкцией универсальной машины Тьюринга из [18] и на отдельной ленте хранить число сделанных машиной x шагов, после чего, как только оно станет равно t, дать ответ. Лемма 3 доказана.

Лемма 4 (см. [18]). *Следующий язык полон в* Σ_1:

$$\texttt{Halt} = \{(x, y) \mid \text{МТ } \textit{с номером } x \textit{ останавливается на слове } y\}.$$

Лемма 5 (см. [18]). *Следующий язык полон в* Σ_1:

$$\texttt{Halt}_\varepsilon = \{x \mid \text{МТ } \textit{с номером } x \textit{ останавливается на пустом слове } \varepsilon\}.$$

Лемма 6 (см. [18]). *Следующий язык полон в* Π_2:

$$\texttt{Inf} = \{x \mid \text{МТ } \textit{с номером } x \textit{ останавливается на бесконечном множестве слов}\}.$$

3. Регулярные языки

Будем считать, что регулярные языки и автоматы определены в соответствии с [19]. Детерминированный и недетерминированный конечные автоматы кратко именуем ДКА и НКА соответственно.

Пусть дан автомат A. Будем обозначать распознаваемый им язык через $\mathcal{L}(A)$. Приведём теорему, дающую характеризацию регулярных языков в унарном алфавите.

Теорема 3 (см. [20]). *Унарный язык регулярен тогда и только тогда, когда множество длин его слов является конечным объединением арифметических прогрессий.*

Выделим важное свойство построенной кодировки машин Тьюринга.

Лемма 7. *В любом бесконечном унарном регулярном языке R есть бесконечно много номеров любой машины Тьюринга.*

Доказательство следует из теоремы 3 и свойств построенной нумерации машин. Лемма 7 доказана.

Сформулируем также несколько вспомогательных утверждений о вычислимости некоторых операций над языками. Будем считать, что ДКА на вход алгоритму подаётся в виде стандартного описания.

Лемма 8. *Функция, принимающая на вход два* ДКА *и возвращающая* ДКА *для пересечения языков входных автоматов, вычислима.*

Доказательство. Автомат для пересечения языков строится с использованием конструкции автоматов, описанной например в [19]. Лемма 8 доказана.

Лемма 9. *Функция, принимающая на вход регулярное выражение и возвращающая эквивалентный* ДКА*, вычислима.*

Доказательство. Алгоритм строится непосредственно из доказательства теоремы Клини об эквивалентности классов языков, задаваемых регулярными выражениями и ДКА (см. [19]). Лемма 9 доказана.

Лемма 10. *Функция, принимающая на вход* ДКА M *и возвращающая автомат, распознающий все префиксы слов языка* $\mathcal{L}(M)$*, вычислима.*

Доказательство. Действительно, достаточно сделать финальными все состояния автомата M, из которых достижимо какое-либо финальное состояние M. Лемма 10 доказана.

Лемма 11. *Язык* ДКА*, распознающих конечные языки, разрешим.*

Доказательство. Отметим, что задача достижимости на графе автомата разрешима, например при помощи обхода данного графа в глубину. Для каждого состояния, из которого достижимо финальное и которое достижимо из стартового, проверим, достижимо ли оно само из себя. Язык бесконечен тогда и только тогда, когда в нём есть слова сколь угодно большой длины, что равносильно тому, что существует такой путь, на котором есть состояние, достижимое из самого себя. Таким образом, язык бесконечен тогда и только тогда, когда состояние с указанным свойством найдено. Лемма 11 доказана.

Лемма 12. *Язык* ДКА*, распознающих пустой язык, разрешим.*

Доказательство. Проверим, существует ли такое состояние, которое достижимо из стартового и из которого достижимо финальное состояние. Язык непуст тогда и только тогда, когда такое состояние существует. Лемма 12 доказана.

Лемма 13. *Если* A — ДКА *и* $\mathcal{L}(A) \subset 0^*1^*$*, то вычислима функция, отображающая* A *в конечное множество* ДКА $\{Q_1, \ldots, Q_n, P_1, \ldots, P_n\}$*, где*

1) $\mathcal{L}(Q_i) \subset 0^*$;
2) $\mathcal{L}(P_i) \subset 1^*$;
3) $\mathcal{L}(A) = \bigcup\limits_{i=1}^{n} \mathcal{L}(Q_i)\mathcal{L}(P_i)$.

Доказательство. Для произвольного состояния i автомата A построим автомат Q_i для всех путей из стартового состояния в i, состоящих только из символа 0, и автомат P_i для всех путей из i в финальное состояние, состоящих только из символа 1. Оставим только такие пары автоматов Q_i, P_i, языки которых непусты. Тогда действительно $\mathcal{L}(A) = \bigcup\limits_{i=1}^{n} \mathcal{L}(Q_i)\mathcal{L}(P_i)$. Лемма 13 доказана.

4. Задачи регулярной реализуемости

Зачастую (см., например, [17]) при рассмотрении задач регулярной реализуемости входом считается некоторый НКА. В дальнейшем будем считать, что на вход подаётся ДКА. Так как построение эквивалентного ДКА — вычислимая операция (см. [19]), это не меняет доказанных свойств.

Определение 5. *Задачей регулярной реализуемости* для языка L назовём язык

$$\mathtt{DRR}_L = \{A \mid |L \cap \mathcal{L}(A)| \neq \varnothing\}.$$

Лемма 14. *Если язык* L *разрешим, то* $\mathtt{DRR}_L \in \Sigma_1$.

Доказательство. Пересечение L с входным языком непусто тогда и только тогда, когда существует слово, принадлежащее им обоим, иначе говоря,

$$A \in \mathtt{DRR}_L \Leftrightarrow \exists w\colon A(w) \wedge (w \in L).$$

Если L разрешим, то под кванторами находится разрешимый предикат, следовательно, $\mathtt{DRR}_L \in \Sigma_1$. Лемма 14 доказана.

5. Задачи бесконечной регулярной реализуемости

Определение 6. *Задачей бесконечной регулярной реализуемости* для языка L назовём язык

$$\mathtt{DRR}_L^\infty = \{A \mid |L \cap \mathcal{L}(A)| = \infty\}.$$

Лемма 15. *Если язык L разрешим, то $\mathtt{DRR}_L^\infty \in \Pi_2$.*

Доказательство. Пересечение L с входным языком бесконечно тогда и только тогда, когда в нём есть слово произвольной длины, иначе говоря,

$$A \in \mathtt{DRR}_L^\infty \Leftrightarrow \forall v\, \exists w\colon A(w) \wedge (w \in L) \wedge (|w| > |v|).$$

Если L разрешим, то под кванторами находится разрешимый предикат, следовательно, $\mathtt{DRR}_L^\infty \in \Pi_2$. Лемма 15 доказана.

Лемма 16. *Для любого языка L задача $\mathtt{DRR}_{0L}^\infty$ сводится к $\mathtt{DRR}_L^\infty$.*

Доказательство. Рассмотрим сводящую функцию f, которая по автомату A строит автомат для языка $\{x \mid 0x \in \mathcal{L}(A)\}$. Такая функция вычислима, так как для построения соответствующего автомата достаточно определить, какие состояния достижимы из стартового по 0. Тогда имеет место цепочка эквивалентностей

$$A \in \mathtt{DRR}_{0L}^\infty \Leftrightarrow |0L \cap \mathcal{L}(A)| = \infty \Leftrightarrow |L \cap \mathcal{L}(f(A))| = \infty \Leftrightarrow f(A) \in \mathtt{DRR}_L^\infty.$$

Лемма 16 доказана.

6. Основные результаты

Лемма 17. *Если языки L и $\mathtt{DRR}_L$ разрешимы, то $\mathtt{DRR}_L^\infty \in \Pi_1$.*

Доказательство. Рассмотрим ДКА A и положим

$$M_{i+} = \{w \in \mathcal{L}(A) \mid |w| \geqslant i\}.$$

Построим автомат A_{i+} для языка M_{i+} как пересечение автомата A и автомата для языка $\Sigma^i\Sigma^*$.

Обозначим через $t(A, i)$ функцию, отображающую автомат A в автомат A_{i+}. Эта функция вычислима в силу лемм 8 и 9. Пересечение произвольного языка с L бесконечно тогда и только тогда, когда для любого i непусто пересечение подмножества его слов длины не меньше i с языком L. Другими словами,

$$A \in \mathtt{DRR}_L^\infty \Leftrightarrow \forall v\, t(A, |v|) \in \mathtt{DRR}_L.$$

Если задача регулярной реализуемости для языка L разрешима, то под кванторами находится разрешимый предикат, следовательно, $\mathtt{DRR}_L^\infty \in \Pi_1$.

Заметим, что требование разрешимости L избыточно, так как оно следует из разрешимости $\mathtt{DRR}_L$. Действительно, вычислимая функция f, сопоставляющая слову w автомат с языком $\{w\}$, является сводимостью L к $\mathtt{DRR}_L$. Лемма 17 доказана.

Теорема 4. *Существует разрешимый язык L такой, что $\mathtt{DRR}_L$ разрешим, а $\mathtt{DRR}_L^\infty$ полон в Π_1.*

Доказательство. Рассмотрим язык

$$L = \{0^x 1^t \mid \text{МТ с номером } x \text{ не останавливается на } \varepsilon \text{ за } t \text{ шагов}\}.$$

Язык L сводится к $\overline{\mathtt{Time}}$ с помощью функции $f(0^x1^t) = (x, \varepsilon, t)$, поэтому разрешим по леммам 1 и 3 и теореме 1.

Сначала покажем разрешимость $\mathtt{DRR}_L$. Пусть на вход подан автомат A, положим $R = \mathcal{L}(A)$. Построим автомат для языка $R' = R \cap 0^*1^*$. Эта операция вычислима в силу лемм 8 и 9. Заметим, что $R \cap L = R' \cap L$, поэтому если $R' = \varnothing$, то $R \cap L = \varnothing$, а значит, $A \notin \mathtt{DRR}_L$. Проверка R' на пустоту является вычислимой операцией в силу леммы 12.

Пусть R' непуст. Построим по нему множество автоматов $\{Q_1, \ldots, Q_n, P_1, \ldots, P_n\}$, как описано в лемме 13. Язык R' пересекается с L тогда и только тогда, когда какой-то $\mathcal{L}(Q_i)\mathcal{L}(P_i)$ пересекается с L. Рассмотрим каждую пару автоматов Q_i, P_i в отдельности. Возможны несколько случаев в зависимости от конечности их языков.

Если $\mathcal{L}(Q_i)$ бесконечен, то имеем бесконечный унарный язык и по лемме 7 в нём встречается слово, длина которого соответствует номеру МТ, не останавливающейся за любое число шагов, а значит, пересечение $L \cap \mathcal{L}(Q_i)\mathcal{L}(P_i) \neq \varnothing$ и $R \in \mathtt{DRR}_L$.

Если язык $\mathcal{L}(Q_i)$ конечен, то обратим внимание на автомат P_i. В случае конечного $\mathcal{L}(P_i)$ проверим каждое слово $0^q1^p \in \mathcal{L}(Q_i)\mathcal{L}(P_i)$. Пересечение $L \cap \mathcal{L}(Q_i)\mathcal{L}(P_i)$ непусто тогда и только тогда, когда МТ с номером q не останавливается за p шагов на пустом слове, а это вычислимая операция, так как разрешим язык $\mathtt{Time}$.

В случае бесконечного $\mathcal{L}(P_i)$ (и конечного $\mathcal{L}(Q_i)$) рассмотрим каждое слово $0^q \in \mathcal{L}(Q_i)$. Выберем наименьшее по длине слово $1^p \in \mathcal{L}(P_i)$ и проверим, останавливается ли МТ с номером q на пустом слове за p шагов. Если останавливается, то пересечение $L \in \mathcal{L}(Q_i)\mathcal{L}(P_i)$ пусто, иначе нет.

Теперь покажем, что $\mathtt{DRR}_L^\infty$ полон в Π_1. Построим сводимость $\overline{\mathtt{Halt}}_\varepsilon$ к L и воспользуемся леммами 1, 2 и 5. Заметим, что функция $f(x) = A$, где $\mathcal{L}(A) = (0^x1^*)$, вычислима в силу леммы 9. Действительно, 0^x1^* имеет бесконечное пересечение с L тогда и только тогда, когда МТ с номером x не остановится на пустом слове. Тогда в силу лемм 1, 2 и 5 верно, что $\mathtt{DRR}_L^\infty$ полон в Π_1. Теорема 4 доказана.

Теорема 5. *Существует разрешимый язык L такой, что $\mathtt{DRR}_L^\infty$ разрешим, а $\mathtt{DRR}_L$ полон в Σ_1.*

Доказательство. Рассмотрим язык

$$L = \{0^x 1^t \mid \text{МТ с номером } x \text{ останавливается на } \varepsilon \text{ на шаге } t\}.$$

Язык L сводится к $\mathtt{Time}$ с помощью функции $f(0^x 1^t) = (x, \varepsilon, t)$, поэтому разрешим по леммам 1 и 3.

Сначала покажем разрешимость $\mathtt{DRR}_L^\infty$. Пусть на вход подан автомат A, обозначим $R = \mathcal{L}(A)$. Построим автомат для языка $R' = R \cap 0^*1^*$. Эта операция вычислима в силу лемм 8 и 9. Заметим, что $R \cap L = R' \cap L$, поэтому если R' конечен, то конечно и $R \cap L$, а значит, $A \notin \mathtt{DRR}_L^\infty$. Проверка R' на конечность является вычислимой операцией в силу леммы 11.

Пусть R' бесконечен. Рассмотрим тогда R_0 — язык, получаемый пересечением языка префиксов R' с 0^* (построение автомата для такого языка вычислимо в силу лемм 8–10). Если R_0 конечен, то конечно и пересечение R' и L, так как в L для каждого i лежит не более одного слова из 0^i1^*.

Пусть R_0 бесконечен. Тогда существует состояние q автомата для языка R' такое, что из стартового в q ведёт бесконечно много слов из 0^* и есть слово 1^z, которое ведёт из q в финальное состояние. Однако в унарном регулярном языке слов, которые ведут из стартового состояния в q, бесконечно много раз встречается номер любой МТ по лемме 7. В том числе и тех МТ, которые останавливаются за z шагов. Следовательно, пересечение $R \cap L$ бесконечно, и $A \in \mathtt{DRR}_L^\infty$.

Теперь покажем, что $\mathtt{DRR}_L$ полно в Σ_1. Построим сводимость $\mathtt{Halt}_\varepsilon$ к $\mathtt{DRR}_L$ с помощью функции $f(x) = A$, где $\mathcal{L}(A) = 0^x1^*$, которая вычислима в силу леммы 9. Действительно, 0^x1^* пересекается с L тогда и только тогда, когда МТ с номером x остановится на пустом слове. Тогда в силу лемм 1 и 5 верно, что $\mathtt{DRR}_L$ полон в Σ_1. Теорема 5 доказана.

Теорема 6. *Существует разрешимый язык L такой, что $\mathtt{DRR}_L$ полон в Σ_1, а $\mathtt{DRR}_L^\infty$ полон в Π_2.*

Доказательство. Зададим на словах нумерацию в алфавитном порядке (пусть слово w_y имеет номер y) и рассмотрим язык

$$L = \{0^x 1^y 2^t \mid \text{МТ с номером } x \text{ останавливается на слове } w_y \text{ на шаге } t\}.$$

Язык L сводится к $\mathtt{Time}$ с помощью функции $f(0^x1^y2^t) = (x, w_y, t)$, поэтому разрешим по леммам 1 и 3. В силу разрешимости L из лемм 14 и 15 следует, что $\mathtt{DRR}_L \in \Sigma_1$, а $\mathtt{DRR}_L^\infty \in \Pi_2$.

Для начала докажем, что $\mathtt{DRR}_L$ труден в Σ_1, построив сводимость $\mathtt{Halt}$ к $\mathtt{DRR}_L$. Рассмотрим вычислимую (по лемме 9) функцию $f\colon (x, w_y) \to A$, каждой паре из номера МТ и слова сопоставляющую ДКА A такой, что

$\mathcal{L}(A) = 0^x1^y2^*$. Если данная МТ останавливалась на данном слове (иначе говоря, $(x, w_y) \in \mathtt{Halt}$), то существует шаг p, на котором она останавливается. Тогда $0^x1^y2^*$ пересекается с L, так как в них обоих лежит слово $0^x1^y2^p$, а значит, $f(x, w_y) \in \mathtt{DRR}_L$. Обратно, если МТ x не останавливается на слове y, то никакое слово из $0^x1^y2^*$ не лежит в L, следовательно, $f(x, y) \notin \mathtt{DRR}_L$.

Докажем, что $\mathtt{DRR}_L^\infty$ труден в Π_2, построим сводимость $\mathtt{Inf}$ к $\mathtt{DRR}_L^\infty$. Рассмотрим вычислимую (по лемме 9) функцию $f\colon x \to A$, которая каждому номеру МТ сопоставляет ДКА A такой, что $\mathcal{L}(A) = 0^x1^*2^*$. Данная МТ останавливается на конкретном слове w_y тогда и только тогда, когда существует шаг p, на котором она останавливается. Тогда $0^x1^y2^*$ пересекается с L, так как в них обоих лежит слово $0^x1^y2^p$, при этом мощность данного пересечения равна 1. Значит, бесконечность множества слов, на которых останавливается МТ с номером x, равносильна бесконечности пересечения L с $0^x1^*2^*$. Теорема 6 доказана.

Теорема 7. *Существует разрешимый язык L такой, что $\mathtt{DRR}_L$ полон в Σ_1, а $\mathtt{DRR}_L^\infty$ полон в Π_1.*

Доказательство. Обозначим язык из теоремы 5 через L_0, а язык из теоремы 4 через L_1. Рассмотрим язык $L = 0L_0 \cup 1L_1$. Он разрешим, так как разрешимы L_0 и L_1.

Из леммы 14 следует, что $\mathtt{DRR}_L \in \Sigma_1$. Покажем, что $\mathtt{DRR}_L^\infty \in \Pi_1$. По определению $A \in \mathtt{DRR}_L^\infty$ равносильно тому, что $|0L_0 \cap \mathcal{L}(A)| = \infty$ или $|1L_1 \cap \mathcal{L}(A)| = \infty$. Следовательно, $\mathtt{DRR}_L^\infty = \mathtt{DRR}_{0L_0}^\infty \cup \mathtt{DRR}_{1L_1}^\infty$. По теореме 1 и леммам 1, 16 получаем, что $\mathtt{DRR}_L^\infty \in \Pi_1$.

Покажем, что $\mathtt{DRR}_L$ труден в Σ_1. Для этого построим функцию f, сводящую $\mathtt{DRR}_L$ к $\mathtt{DRR}_{L_0}$. Пусть f отображает A в автомат для языка $0\mathcal{L}(A)$. Так как это конкатенация регулярных языков, f вычислима, при этом

$$\begin{aligned} A \in \mathtt{DRR}_L &\Leftrightarrow |L \cap \mathcal{L}(A)| \neq \varnothing \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(A)| \neq \varnothing \vee |1L_1 \cap \mathcal{L}(A)| \neq \varnothing \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(f(A))| \neq \varnothing \Leftrightarrow f(A) \in \mathtt{DRR}_{L_0}. \end{aligned}$$

Осталось показать, что $\mathtt{DRR}_L^\infty$ труден в Π_1. Для этого построим функцию f, сводящую $\mathtt{DRR}_L^\infty$ к $\mathtt{DRR}_{L_1}^\infty$. Пусть f отображает A в автомат для языка $1\mathcal{L}(A)$. Так как это конкатенация регулярных языков, f вычислима, при этом

$$\begin{aligned} A \in \mathtt{DRR}_L^\infty &\Leftrightarrow |L \cap \mathcal{L}(A)| = \infty \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(A)| = \infty \vee |1L_1 \cap \mathcal{L}(A)| = \infty \Leftrightarrow \\ &\Leftrightarrow |1L_1 \cap \mathcal{L}(f(A))| = \infty \Leftrightarrow f(A) \in \mathtt{DRR}_{L_1}^\infty. \end{aligned}$$

Теорема 7 доказана.

Заметим, что в случае двухбуквенного алфавита приведённые результаты сохраняются. Действительно, рассмотрим отображение $f\colon 0 \mapsto 00$, $1 \mapsto 11$, $2 \mapsto 01$, которое естественным образом продолжается до отображения на языках. Отображение f инъективно, и обратное к нему также инъективно, хотя и не всюду определено. Остаётся заметить, что вычислима функция, сопоставляющая автомату A автомат B такой, что $\mathcal{L}(B) = f(\mathcal{L}(A))$. Действительно, достаточно, например, построить по автомату регулярное выражение, подействовать на все символы в нём отображением f, после чего построить автомат по регулярному выражению. Случай однобуквенного алфавита является вопросом для дальнейшего изучения.

7. Относительно регулярные алгебры

Так как булевы алгебры, состоящие из пересечений заданного языка с регулярными, образуют довольно богатый класс (а именно, любая атомная булева алгебра изоморфна некоторой в этом классе [21]), отдельный интерес вызывает приложение задач регулярной реализуемости к изучению их свойств. В данном разделе определения даются в соответствии с [22].

Определение 7. *Булевой алгеброй* называется множество M с бинарными алгебраическими операциями $\vee$ и $\wedge$, а также унарной алгебраической операцией $\neg$ такими, что для любых $a, b, c \in M$ имеют место равенства

(1) $a \vee b = b \vee a$;
(2) $a \vee (b \vee c) = (a \vee b) \vee c$;
(3) $a \vee a = a$;
(4) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$;
(5) $\neg(a \vee b) = \neg a \wedge \neg b$;
(6) $(a \wedge \neg a) \vee b = b$;
(7) $\neg\neg a = a$.

На произвольной булевой алгебре можно ввести отношение частичного порядка: $a \leqslant b \Leftrightarrow a \wedge b = a$. Минимальный и максимальный элементы обозначим через 0 и 1 соответственно. Нетрудно проверить, что в булевой алгебре такие элементы уникальны и обладают свойствами $a \vee 0 = a = a \wedge 1$, $a \wedge 0 = 0$ и $a \vee 1 = 1$. Особый интерес при изучении булевых алгебр играют атомы.

Определение 8. Ненулевой элемент a будем называть *атомом*, если из отношения $b \leqslant a$ следует, что $b = 0$ или $b = a$.

Далее будем иметь дело только со счётными атомными булевыми алгебрами.

Определение 9. Булева алгебра называется *счётной*, если мощность счётна.

Определение 10. Алгебру A будем называть *атомной*, если для любого элемента $a \in A$ существует атом $b \in A$ такой, что $b \leqslant a$.

Так как регулярные языки замкнуты относительно объединения, пересечения и дополнения, они образуют булеву алгебру REG.

Определение 11. *Относительно регулярной алгеброй* для языка L будем называть булеву алгебру $\mathrm{REG}_L = \{M \cap L \mid M \in \mathrm{REG}\}$ с операциями объединения, пересечения и дополнения до L.

Заметим, что REG_{Σ^*} совпадает с REG. Из определения также видно, что относительно регулярная алгебра всегда является гомоморфным образом алгебры регулярных языков, а соответствующий гомоморфизм имеет вид $\varphi(x) = x \cap L$. Стоит отметить, что относительно регулярные алгебры всегда атомны: в роли атомов в них выступают языки из одного слова.

Как и для других алгебраических структур, важным оказывается понятие идеала булевой алгебры, особую роль в изучении булевых алгебр играет идеал Фреше.

Определение 12. Подмножество $I \subset A$ булевой алгебры A называется *идеалом*, если I содержит 0, для каждого элемента i из I все меньшие i элементы также содержатся в I и I замкнуто относительно операции $\vee$.

Отдельно подчеркнём, что идеал булевой алгебры сам может не являться булевой алгеброй.

Определение 13. *Идеалом Фреше* $F(A)$ булевой алгебры A называется идеал, состоящий из всех конечных дизъюнкций её атомов.

В относительно регулярной алгебре для языка L есть простой способ указать на её элемент X: достаточно указать регулярный язык M, пересечение которого с L равно X. Такое представление неоднозначно, что создаёт дополнительные проблемы в алгоритмических вопросах о данной булевой алгебре. Задача проверки равенства двух элементов сводится к задаче $\mathtt{DRR}_L$ для их симметрической разности (это также регулярный язык). Если интересоваться фактором относительно регулярной алгебры по идеалу Фреше, то проверка равенства двух элементов сводится к задаче бесконечной регулярной реализуемости для их симметрической разности (при представлении элементов этой фактор-алгебры регулярными языками). Такая связь между задачами регулярной реализуемости и алгоритмическими вопросами о булевых алгебрах представляется нам важной и заслуживающей дальнейшего изучения.

Авторы благодарны рецензенту за полезные замечания и предложения, существенно улучшившие содержание работы.

Финансирование работы

Работа второго автора выполнена в рамках Программы фундаментальных исследований НИУ ВШЭ, а также имеет частичную финансовую поддержку в рамках гос. задания (проект № FFNG–2024–0003). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Bouajjani A., Esparza J., Maler O.** Reachability analysis of pushdown automata: Application to model-checking // CONCUR'97: Concurrency theory. Proc. 8th Int. Conf. (Warsaw, Poland, July 1–4, 1997). Heidelberg: Springer, 1997. P. 135–150. (Lect. Notes Comput. Sci.; V. 1243). DOI: 10.1007/3-540-63141-0_10.
2. **Rubtsov A., Vyalyi M.** Regular realizability problems and context-free languages // Descriptional complexity of formal systems. Proc. 17th Int. Workshop DCFS 2015 (Waterloo, ON, Canada, June 25–27, 2015). Cham: Springer, 2015. P. 256–267. (Lect. Notes Comput. Sci.; V. 9118). DOI: 10.1007/978-3-319-19225-3_22.
3. **Chistikov D., Majumdar R., Schepper P.** Subcubic certificates for CFL reachability // Proc. ACM Program. Lang. 2022. V. 6, No. POPL. Article ID 41. 29 p. DOI: 10.1145/3498702.
4. **Pavlogiannis A.** CFL/Dyck reachability: An algorithmic perspective // ACM SIGLOG News. 2023. V. 9, No. 4. P. 5–25. DOI: 10.1145/3583660.3583664.
5. **Kjelstrøm A. H., Pavlogiannis A.** The decidability and complexity of interleaved bidirected Dyck reachability // Proc. ACM Program. Lang. 2022. V. 6, No. POPL. Article ID 12. 26 p. DOI: 10.1145/3498673.
6. **Koutris P., Deep S.** The fine-grained complexity of CFL reachability // Proc. ACM Program. Lang. 2023. V. 7, No. POPL. Article ID 59. 27 p. DOI: 10.1145/3571252.
7. **Anderson T., Loftus J., Rampersad N., Santean N., Shallit J.** Detecting palindromes, patterns and borders in regular languages // Inf. Comput. 2009. V. 207. P. 1096–1118. DOI: 10.1016/j.ic.2008.06.007.
8. **Wolf P., Fernau H.** Regular intersection emptiness of graph problems: Finding a needle in a haystack of graphs with the help of automata. Ithaca, NY, 2020. 29 p. (e-Print Archive / Cornell Univ.; arXiv:2003.05826). DOI: 10.48550/arXiv.2003.05826.
9. **Wolf P.** From decidability to undecidability by considering regular sets of instances // Theor. Comput. Sci. 2022. V. 899. P. 25–38. DOI: 10.1016/j.tcs.2021.11.006.

10. **Diekert V., Fernau H., Wolf P.** Properties of graphs specified by a regular language // Acta Inform. 2022. V. 59. P. 357–385. DOI: 10.1007/s00236-022-00427-z.
11. **Wolf P.** On the decidability of finding a positive ILP-instance in a regular set of ILP-instances // Acta Inform. 2022. V. 59. P. 505–519. DOI: 10.1007/s00236-022-00429-x.
12. **Feder T., Vardi M. Y.** The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory // SIAM J. Comput. 1999. V. 28, No. 1. P. 57–104.
13. **Bulatov A. A.** A dichotomy theorem for nonuniform CSPs // Proc. 58th IEEE Annu. Symp. Foundations of Computer Science (Berkeley, CA, USA, Oct. 15–17, 2017). Piscataway: IEEE, 2017. P. 319–330. DOI: 10.1109/FOCS.2017.37
14. **Zhuk D.** A proof of CSP dichotomy conjecture // Proc. 58th IEEE Annu. Symp. Foundations of Computer Science (Berkeley, CA, USA, Oct. 15–17, 2017). Piscataway: IEEE, 2017. P. 331–342. DOI: 10.1109/FOCS.2017.38
15. **Bulatov A. A.** A dichotomy theorem for nonuniform CSPs. Ithaca, NY, 2017. 101 p. (e-Print Archive / Cornell Univ.; arXiv:1703.03021). DOI: 10.48550/arXiv.1703.03021.
16. **Zhuk D.** A proof of the CSP dichotomy conjecture // J. ACM. 2020. V. 67, No. 5. Article ID 30. 78 p. DOI: 10.1145/3402029.
17. **Вялый М. Н., Рубцов А. А.** Задачи регулярной реализуемости для описаний конечных отношений // Пробл. передачи информации. 2024. Т. 60, № 3. С. 46–58.
18. **Soare R.** Turing computability: Theory and applications. Heidelberg: Springer, 2016. 264 p. DOI: 10.1007/978-3-642-3.
19. **Kozen D.** Automata and computability. New York: Springer, 2012. 400 p. DOI: 10.1007/978-1-4612-1844-9.
20. **Shallit J. O.** A second course in formal languages and automata theory. New York: Camb. Univ. Press, 2008. 240 p. DOI: 10.1017/CBO9780511808876.
21. **Шиманогов И. Н., Вялый М. Н.** Классификация относительно регулярных алгебр // Тр. МФТИ. 2024. Т. 16, № 4. С. 128–134.
22. **Гончаров С. С.** Счётные булевы алгебры и разрешимость. Новосибирск: Науч. книга, 1996. 364 с.

Шиманогов Игорь Николаевич
Вялый Михаил Николаевич

Статья поступила
9 декабря 2024 г.
После доработки —
20 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 213–230

UDC 519.713 DOI: 10.33048/daio.2025.32.822

INFINITE REGULAR REALIZABILITY PROBLEMS

I. N. Shimanogov[1, a] and *M. N. Vyalyi*[1, 2, 3, b]

[1] Moscow Institute of Physics and Technology
(National Research University),
1A Bld. 1 Kerchenskaya Street, 117303 Moscow, Russia

[2] Federal Research Center "Computer Science and Control" RAS
44 Bld. 2 Vavilov Street, 119333 Moscow, Russia

[3] HSE University,
11 Pokrovsky Boulevard, 109028 Moscow, Russia

E-mail: [a] shimanogov.in@phystech.edu, [b] vyalyi@gmail.com

Abstract. A well-studied class of algorithmic problems is that of regular realizability, i.e. checking the non-emptiness of the intersection of a regular language with a given language. This problem has a natural algebraic interpretation which is verifying whether an element of a Boolean algebra belongs to the kernel of a certain homomorphism. This motivates the consideration of a similar problem of infinite regular realizability that is checking whether the intersection of a regular language with a given language is infinite. The paper examines the case of decidable languages and provides a comparative analysis of the complexity of infinite regular realizability problems versus regular realizability problems. Bibliogr. 22.

Keywords: regular language, regular realizability problem, arithmetic hierarchy.

References

1. **A. Bouajjani, J. Esparza,** and **O. Maler,** Reachability analysis of pushdown automata: Application to model-checking, in *CONCUR'97: Concurrency Theory*, Proc. 8th Int. Conf. (Warsaw, Poland, July 1–4, 1997) (Springer, Heidelberg, 1997), pp. 135–150 (Lect. Notes Comput. Sci., V. 1243), DOI: 10.1007/3-540-63141-0_10.

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

2. **A. Rubtsov** and **M. Vyalyi,** Regular realizability problems and context-free languages, in *Descriptional Complexity of Formal Systems*, Proc. 17th Int. Workshop DCFS 2015 (Waterloo, ON, Canada, June 25–27, 2015) (Springer, Cham, 2015), pp. 256–267 (Lect. Notes Comput. Sci., V. 9118), DOI: 10.1007/978-3-319-19225-3_22.
3. **D. Chistikov, R. Majumdar,** and **P. Schepper,** Subcubic certificates for CFL reachability, *Proc. ACM Program. Lang.* **6** (POPL), ID 41 (2022), DOI: 10.1145/3498702.
4. **A. Pavlogiannis,** CFL/Dyck reachability: An algorithmic perspective, *ACM SIGLOG News* **9** (4), 5–25 (2023), DOI: 10.1145/3583660.3583664.
5. **A. H. Kjelstrøm** and **A. Pavlogiannis,** The decidability and complexity of interleaved bidirected Dyck reachability, *Proc. ACM Program. Lang.* **6** (POPL), ID 12 (2022), DOI: 10.1145/3498673.
6. **P. Koutris** and **S. Deep,** The fine-grained complexity of CFL reachability, *Proc. ACM Program. Lang.* **7** (POPL), ID 59 (2023), DOI: 10.1145/3571252.
7. **T. Anderson, J. Loftus, N. Rampersad, N. Santean,** and **J. Shallit,** Detecting palindromes, patterns and borders in regular languages, *Inf. Comput.* **207**, 1096–1118 (2009), DOI: 10.1016/j.ic.2008.06.007.
8. **P. Wolf** and **H. Fernau,** Regular intersection emptiness of graph problems: Finding a needle in a haystack of graphs with the help of automata (Ithaca, NY, 2020) (e-Print Archive / Cornell Univ., arXiv:2003.05826), DOI: 10.48550/arXiv.2003.05826.
9. **P. Wolf,** From decidability to undecidability by considering regular sets of instances, *Theor. Comput. Sci.* **899**, 25–38 (2022), DOI: 10.1016/j.tcs.2021.11.006.
10. **V. Diekert, H. Fernau,** and **P. Wolf,** Properties of graphs specified by a regular language, *Acta Inform.* **59**, 357–385 (2022), DOI: 10.1007/s00236-022-00427-z.
11. **P. Wolf,** On the decidability of finding a positive ILP-instance in a regular set of ILP-instances, *Acta Inform.* **59**, 505–519 (2022), DOI: 10.1007/s00236-022-00429-x.
12. **T. Feder** and **M. Y. Vardi,** The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory, *SIAM J. Comput.* **28** (1), 57–104 (1999).
13. **A. A. Bulatov,** A dichotomy theorem for nonuniform CSPs, in *Proc. 58th IEEE Annu. Symp. Foundations of Computer Science* (Berkeley, CA, USA, Oct. 15–17, 2017) (IEEE, Piscataway, 2017), pp. 319–330, DOI: 10.1109/FOCS.2017.37
14. **D. Zhuk,** A proof of CSP dichotomy conjecture, in *Proc. 58th IEEE Annu. Symp. Foundations of Computer Science* (Berkeley, CA, USA, Oct. 15–17, 2017) (IEEE, Piscataway, 2017), pp. 331–342, DOI: 10.1109/FOCS.2017.38
15. **A. A. Bulatov,** A dichotomy theorem for nonuniform CSPs (Ithaca, NY, 2017) (e-Print Archive / Cornell Univ., arXiv:1703.03021), DOI: 10.48550/arXiv.1703.03021.

16. D. Zhuk, A proof of the CSP dichotomy conjecture, *J. ACM* **67** (5), ID 30 (2020), DOI: 10.1145/3402029.

17. M. N. Vyalyi and **A. A. Rubtsov,** Regular realizability problems for descriptions of finite relations, *Probl. Peredachi Inf.* **60** (3), 46–58 (2024) [Russian] [On universality of regular realizability problems, *Probl. Inf. Transm.* **60** (3), 209–232 (2024), DOI: 10.1134/S0032946024030050].

18. R. Soare, *Turing Computability: Theory and Applications* (Springer, Heidelberg, 2016), DOI: 10.1007/978-3-642-3.

19. D. Kozen, *Automata and Computability* (Springer, New York, 2012), DOI: 10.1007/978-1-4612-1844-9.

20. J. O. Shallit, *A Second Course in Formal Languages and Automata Theory* (Camb. Univ. Press, New York, 2008), DOI: 10.1017/CBO9780511808876.

21. I. N. Shimanogov and **M. N. Vyalyi,** Classification of relative regular algebras, *Tr. MFTI* **16** (4), 128–134 (2024) [Russian].

22. S. C. Goncharov, *Countable Boolean Algebras and Decidability* (Nauchn. Kniga, Novosibirsk, 1996) [Russian].

Igor N. Shimanogov
Mikhail N. Vyalyi

Received December 9, 2024
Revised August 20, 2025
Accepted September 22, 2025

СОДЕРЖАНИЕ ТОМА 32

№ 1

№ 2

№ 3

№ 4

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/

CONTENTS OF VOLUME 32

No. 1

No. 2

No. 3

No. 4

ДИСКРЕТНЫЙ АНАЛИЗ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

2025. Том 32, № 4

Зав. редакцией Е. В. Горкунов

Журнал подготовлен с использованием макропакета LaTeX 2_ε.
The present publication has been typeset using LaTeX 2_ε.

Журнал зарегистрирован в Федеральной службе по надзору
в сфере связи, информационных технологий и массовых коммуникаций.
Свидетельство о регистрации ЭЛ № ФС77-85978 от 26.09.2023 г.
Размещение в сети Интернет: `math-sobolev.ru`.

Дата размещения в сети Интернет 29.05.2026 г.
Формат $70 \times 100\,1/16$. Усл. печ. л. 19,2. Объём 1,70 МБ.

Издательство Института математики,
пр. Академика Коптюга, 4, 630090 Новосибирск, Россия