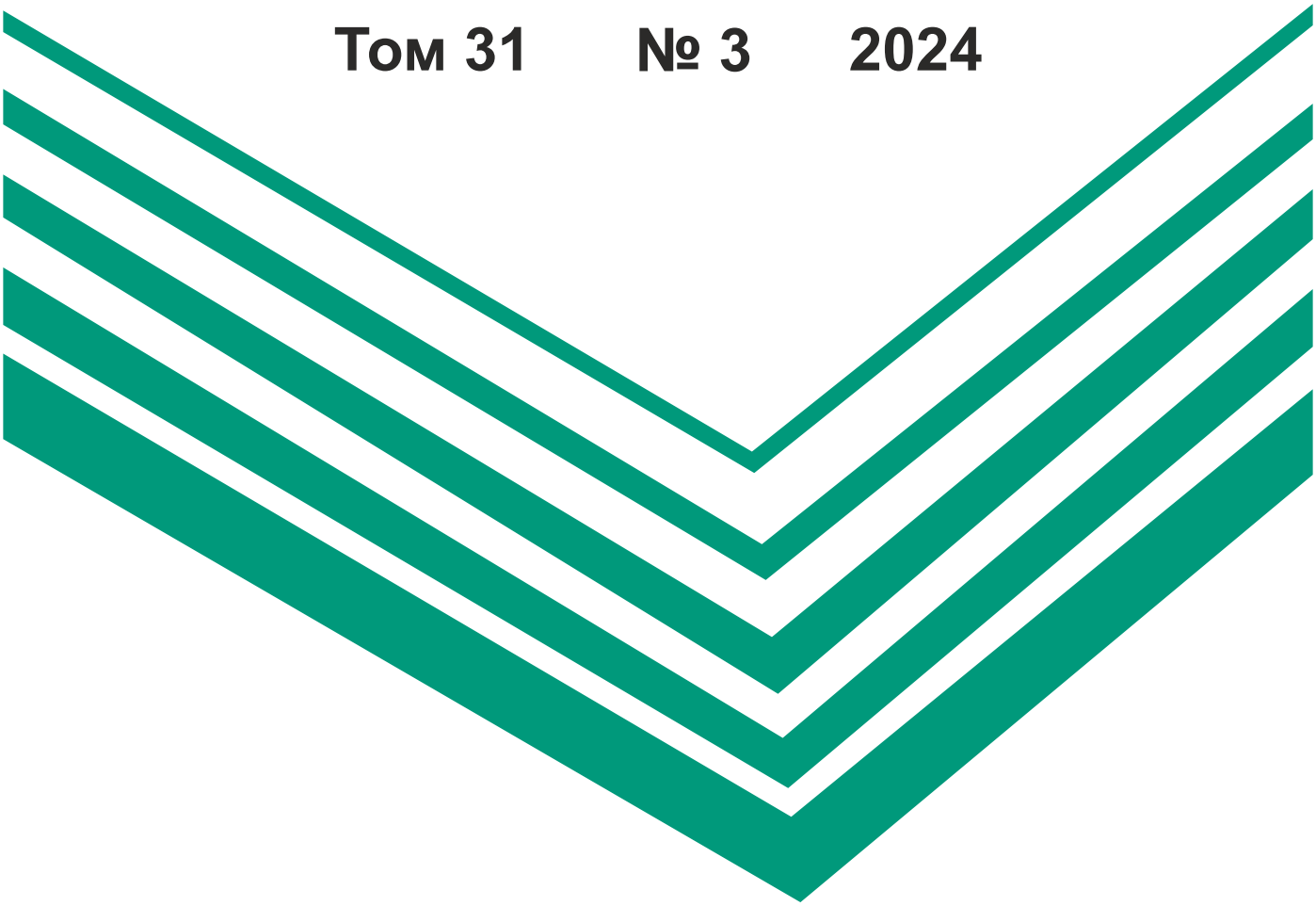
ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 31 № 3 2024



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 31, № 3 (161) **Июль–сентябрь 2024**

СОДЕРЖАНИЕ

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. 31, No. 3 (161) **July–September, 2024**

CONTENTS

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
Июль–сентябрь 2024. Т. 31, № 3. С. 5–23

УДК 519.8+518.25 DOI: 10.33048/daio.2024.31.789

ВЫПУКЛЫЕ ПРОДОЛЖЕНИЯ НЕКОТОРЫХ ДИСКРЕТНЫХ ФУНКЦИЙ

Д. Н. Баротов

Финансовый университет при Правительстве Российской Федерации,
4-й Вешняковский пр-д, 4, 109456 Москва, Россия

E-mail: dnbarotov@fa.ru

Аннотация. Построены выпуклые продолжения для дискретных функций, заданных на вершинах n-мерного единичного куба $[0,1]^n$, произвольного куба $[a,b]^n$ и параллелепипеда $[c_1,d_1]\times[c_2,d_2]\times\cdots\times[c_n,d_n]$. В каждом случае доказано, что, во-первых, для произвольной дискретной функции f, определённой на вершинах множества $\mathbb{G}\in\{[0,1]^n,[a,b]^n,[c_1,d_1]\times[c_2,d_2]\times\cdots\times[c_n,d_n]\}$, существует бесконечно много её выпуклых продолжений на $\mathbb{G}$ и, во-вторых, существует единственная функция вида $f_{DM}\colon\mathbb{G}\to\mathbb{R}$, которая является максимумом среди всех выпуклых продолжений f на $\mathbb{G}$, причём f_{DM} непрерывна на $\mathbb{G}$. Библиогр. 24.

Ключевые слова: дискретная функция, выпуклое продолжение дискретной функции, булева функция, псевдобулева функция.

Введение

В настоящее время теория булевых функций представляет собой замечательную область исследований в дискретной математике с обширными приложениями в криптографии и теории кодирования [1]. Встречается много основных задач, связанных с булевыми переменными, а некоторые задачи, несмотря на зрелость области, не имеют удовлетворительных методов решения. Среди них — проблема решения булевых уравнений и систем булевых уравнений [2]. Эта задача имеет множество приложений, таких как синтез, моделирование и тестирование цифровых сетей и систем СБИС, кодирование выходных данных и назначение состояний конечных автоматов, временной анализ и генерация тестов с задержкой-сбоем для комбинационных схем, автоматическая генерация тестовых

шаблонов, определение начального состояния в схемах, содержащих петли обратной связи [2–4]. В области криптографии она имеет приложения при анализе и взломе блочных шифров, поскольку их можно свести к проблеме решения крупномасштабной системы булевых уравнений [5–12]. В связи с этим развивается множество новых направлений и алгоритмов решения систем булевых уравнений. Одно из направлений заключается в том, что, во-первых, система булевых уравнений, заданная над кольцом булевых полиномов, преобразуется в систему уравнений над полем действительных чисел, а во-вторых, преобразованная система сводится либо к задаче численной минимизации соответствующей целевой функции [13–15], либо к задаче MILP или QUBO [16], либо к системе полиномиальных уравнений, решаемой на множестве целых чисел [2], либо к эквивалентной системе полиномиальных уравнений, анализируемой и решаемой символьными методами [17].

Имеется много способов, позволяющих преобразовать систему булевых уравнений в задачу непрерывной минимизации, поскольку принципиальное отличие таких методов от «переборных» алгоритмов локального поиска в том, что на каждой итерации алгоритма сдвиг по антиградиенту производится по всем переменным одновременно [18–22]. Одна из основных проблем, возникающих при применении этих способов, заключается в том, что минимизируемая целевая функция в искомой области может иметь множество локальных минимумов, что значительно усложняет их практическое использование [13–15, 18, 19, 21, 22]. По теореме Д. Н. Баротова полилинейное продолжение булевой функции тоже играет важную роль для уменьшения числа локальных минимумов целевой функции [18, 22]. По данной тематике недавно в работе [18] были найдены явные формы полилинейных продолжений для произвольных функций, определённых на множестве вершин n-мерного единичного куба, произвольного куба и параллелепипеда, и в каждом конкретном случае была доказана единственность соответствующего полилинейного продолжения.

С учётом этой мотивации в настоящей работе совершенствуются (обобщаются) результаты, полученные недавно в [23]. В разд. 1 приводятся необходимые определения и обозначения. В разд. 2–4 для произвольной вещественнозначной дискретной функции f, определённой на вершинах множества $\mathbb{G} \in \{[0,1]^n, [a,b]^n, [c_1,d_1] \times [c_2,d_2] \times \cdots \times [c_n,d_n]\}$, в частности для любой булевой функции, конструктивно доказывается, что существуют, во-первых, бесконечно много её выпуклых продолжений на $\mathbb{G}$ и, во-вторых, функция $f_{DM} \colon \mathbb{G}^n \to \mathbb{R}$, которая является единственным максимумом среди всех её выпуклых продолжений $\mathbb{G}$.

1. Используемые обозначения и определения

Введём в рассмотрение следующие множества и обозначения:

- $\mathbb{B}^n = \{0,1\}^n$ — множество двоичных слов (булевых векторов) длины n;
- $\mathbb{K}^n = [0,1]^n$ — n-мерный куб, натянутый на множество булевых векторов длины n;
- $\operatorname{int}\mathbb{K}^n = (0,1)^n$ — множество внутренних точек куба $\mathbb{K}^n$;
- $\Lambda_{\mathbb{K}^n}(x) = \left\{\lambda = (\lambda_v)_{v\in\mathbb{B}^n} \in \mathbb{K}^{2^n} \mid \sum\limits_{v\in\mathbb{B}^n} \lambda_v \cdot (1,v) = (1,x)\right\}$ — множество весовых коэффициентов, используемых для представления точки $x \in \mathbb{K}^n$ как выпуклой комбинации вершин куба $\mathbb{K}^n$;
- $\mathbb{K}^n(a,b) = [a,b]^n$ — n-мерный куб со стороной $[a,b] \subset \mathbb{R}$, $a \neq b$;
- $\mathbb{B}^n(a,b) = \{a,b\}^n$ — множество вершин куба $\mathbb{K}^n(a,b)$;
- $\Lambda_{\mathbb{K}^n(a,b)}(x) = \left\{\lambda = (\lambda_v)_{v\in\mathbb{B}^n(a,b)} \in \mathbb{K}^{2^n} \mid \sum\limits_{v\in\mathbb{B}^n(a,b)} \lambda_v \cdot (1,v) = (1,x)\right\}$ — множество весовых коэффициентов, используемых для представления точки $x \in \mathbb{K}^n(a,b)$ как выпуклой комбинации вершин куба $\mathbb{K}^n(a,b)$;
- $\mathbb{P}^n = [c_1,d_1]\times[c_2,d_2]\times\cdots\times[c_n,d_n]$ — параллелепипед, определяемый парой различных точек $c = (c_1,c_2,\ldots,c_n)$, $d = (d_1,d_2,\ldots,d_n) \in \mathbb{R}^n$;
- $\mathbb{BP}^n = \{c_1,d_1\}\times\{c_2,d_2\}\times\cdots\times\{c_n,d_n\}$ — множество вершин параллелепипеда $\mathbb{P}^n$;
- $\Lambda_{\mathbb{P}^n}(x) = \left\{\lambda = (\lambda_v)_{v\in\mathbb{BP}^n} \in \mathbb{K}^{2^n} \mid \sum\limits_{v\in\mathbb{BP}^n} \lambda_v \cdot (1,v) = (1,x)\right\}$ — множество весовых коэффициентов, используемых для представления точки $x \in \mathbb{P}^n$ как выпуклой комбинации вершин параллелепипеда $\mathbb{P}^n$;
- $\rho(x,y) = \sum\limits_{k=1}^{n} I(x_k,y_k)$ — расстояние Хэмминга между векторами $x = (x_1,x_2,\ldots,x_n)$, $y = (y_1,y_2,\ldots,y_n) \in \mathbb{R}^n$, $I(x_k,y_k) = \begin{cases}0, & \text{если } x_k = y_k,\\ 1, & \text{если } x_k \neq y_k.\end{cases}$

Определение 1. Отображение вида $f\colon \mathbb{B}^n \to \mathbb{B}$ назовём *булевой функцией.*

Определение 2. Отображение вида $f\colon \mathbb{B}^n \to \mathbb{R}$ назовём *псевдобулевой функцией.*

Определение 3. Отображение вида $f\colon \mathbb{G} \to \mathbb{R}$, определённое на некотором выпуклом множестве $\mathbb{G}$, назовём *выпуклой функцией,* если для любых $x,y \in \mathbb{G}$ и любого $\alpha \in [0,1]$ выполняется

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

Определение 4. Отображение вида $f_D\colon \mathbb{K}^n \to \mathbb{R}$ назовём *выпуклым продолжением на* $\mathbb{K}^n$ (псевдо-) булевой функции $f\colon \mathbb{B}^n \to \mathbb{B}$ ($\mathbb{R}$), если выполнены следующие условия:

1) функция f_D выпукла на $\mathbb{K}^n$,
2) $f_D(v) = f_B(v)$ для любого $v \in \mathbb{B}^n$.

Определение 5. Отображение вида $f_{DM}\colon \mathbb{K}^n \to \mathbb{R}$ назовём *максимумом* среди всех выпуклых продолжений на $\mathbb{K}^n$ (псевдо-) булевой функции $f\colon \mathbb{B}^n \to \mathbb{B}$ ($\mathbb{R}$), если $f_D(x) \leqslant f_{DM}(x)$ для любого $x \in \mathbb{K}^n$ и любого выпуклого продолжения f_D на $\mathbb{K}^n$ функции f.

В начале работы обоснуем справедливость следующего вспомогательного утверждения.

Лемма 1. *Для любого $v \in \mathbb{BP}^n$ множество $\Lambda_{\mathbb{P}^n}(v)$ состоит из одного вектора, в котором $\lambda_v = 1$, а на остальных местах стоят нули.*

Доказательство. Рассмотрим три случая.

Случай 1. Пусть $v = c$. Тогда

$$\Lambda_{\mathbb{P}^n}(v) = \Lambda_{\mathbb{P}^n}(c) = \Big\{\lambda \in \mathbb{K}^{2^n} \mid \sum_{u\in\mathbb{BP}^n} \lambda_u = 1,\ \sum_{u\in\mathbb{BP}^n} \lambda_u u = c\Big\}.$$

Поскольку $c_k < d_k$, $k \in \{1, 2, \ldots, n\}$, и $\lambda_u \in [0, 1]$, $u \in \mathbb{BP}^n$, а $\sum\limits_{u\in\mathbb{BP}^n} \lambda_u = 1$, приравняв покоординатно, заметим, что

$$\lambda_u = \begin{cases} 1, & \text{если } u = c, \\ 0, & \text{если } u \in \mathbb{BP}^n \setminus \{c\}. \end{cases}$$

Действительно,

$$\sum_{u\in\mathbb{BP}^n} \lambda_u u - c = \sum_{u\in\mathbb{BP}^n} \lambda_u u - \sum_{u\in\mathbb{BP}^n} \lambda_u \cdot c = \\ = \sum_{u\in\mathbb{BP}^n} \lambda_u (u - c) = \sum_{u\in\mathbb{BP}^n\setminus\{c\}} \lambda_u (u - c) \geqslant 0,$$

и хотя бы одна координата этого вектора положительна, если существует $u^* \in \mathbb{BP}^n \setminus \{c\}$ такой, что $\lambda_{u^*} > 0$.

Случай 2. Пусть $v = d$. Тогда аналогичными рассуждениями приходим к выводу, что

$$\sum_{u\in\mathbb{BP}^n} \lambda_u u - d = \sum_{u\in\mathbb{BP}^n\setminus\{d\}} \lambda_u (u - d) \leqslant 0,$$

и хотя бы одна координата этого вектора отрицательна, если существует $u^* \in \mathbb{BP}^n \setminus \{d\}$ такой, что $\lambda_{u^*} > 0$, поэтому

$$\lambda_u = \begin{cases} 1, & \text{если } u = d, \\ 0, & \text{если } u \in \mathbb{BP}^n \setminus \{d\}. \end{cases}$$

Случай 3. Пусть $v \in \mathbb{BP}^n \setminus \{c, d\}$. Тогда

$$\sum_{u \in \mathbb{BP}^n} \lambda_u u - v = \sum_{u \in \mathbb{BP}^n \setminus \{v\}} \lambda_u (u - v). \tag{1}$$

Без ограничения общности предположим, что

$$v = (c_1, c_2, \ldots, c_p, d_{p+1}, d_{p+2}, \ldots, d_n)$$

для некоторого $p \in \{1, 2, \ldots, n-1\}$. В этом случае, если $1 \leqslant k \leqslant p$ и $\lambda_u > 0$ для некоторого вектора $u \in \mathbb{BP}^n \setminus \{v\}$ такого, что $u_k = d_k$, то k-я координата вектора (1) положительна. Аналогично если $p+1 \leqslant k \leqslant n$ и $\lambda_u > 0$ для некоторого вектора $u \in \mathbb{BP}^n \setminus \{v\}$ такого, что $u_k = c_k$, то k-я координата вектора (1) отрицательна. Отсюда

$$\lambda_u = \begin{cases} 1, & \text{если } u = v, \\ 0, & \text{если } u \in \mathbb{BP}^n \setminus \{v\}. \end{cases}$$

Лемма 1 доказана.

В качестве следствий приведём два факта, непосредственно вытекающих из леммы 1.

Следствие 1. *Для любого $v \in \mathbb{B}^n(a, b)$ множество $\Lambda_{\mathbb{K}^n(a,b)}(v)$ состоит из одного вектора, в котором $\lambda_v = 1$, а на остальных местах стоят нули.*

Следствие 2. *Для любого $v \in \mathbb{B}^n$ множество $\Lambda_{\mathbb{K}^n}(v)$ состоит из одного вектора, в котором $\lambda_v = 1$, а на остальных местах стоят нули.*

2. Выпуклые продолжения псевдобулевых функций

В этом разделе конструктивно докажем, что, во-первых, для любой псевдобулевой функции $f\colon \mathbb{B}^n \to \mathbb{R}$, в частности для любой булевой функции, существует бесконечно много функций, каждая из которых является её выпуклым продолжением на $\mathbb{K}^n$ и, во-вторых, для псевдобулевой функции $f\colon \mathbb{B}^n \to \mathbb{R}$, в частности для любой булевой функции, существует функция f_{DM}, которая является единственным максимумом среди всех её выпуклых продолжений $\mathbb{K}^n$.

Согласно следствию 1, приведённому в [18], произвольная псевдобулева функция $f\colon \mathbb{B}^n \to \mathbb{R}$ может быть задана как линейная комбинация базисных функций вида

$$f(x) = \sum_{b \in \mathbb{B}^n} f(b) I_b(x), \tag{2}$$

где базисная функция $I_b\colon \mathbb{B}^n \to \mathbb{B}$ с индексом $b = (b_1, b_2, \ldots, b_n) \in \mathbb{B}^n$ может быть задана в виде

$$I_b(x) = \prod_{k=1}^{n}((2b_k - 1)x_k + 1 - b_k) = \begin{cases} 1, & \text{если } x = b, \\ 0, & \text{если } x \in \mathbb{B}^n \setminus \{b\}. \end{cases}$$

Согласно лемме 1, приведённой в [23], функция

$$f_{DM}^b(x) = \frac{1}{2}\left(1 + \sum_{k=1}^{n}((2b_k - 1)x_k - b_k) + \left|1 + \sum_{k=1}^{n}((2b_k - 1)x_k - b_k)\right|\right) \quad (3)$$

является единственным максимумом среди всех выпуклых продолжений на $\mathbb{K}^n$ булевой функции $I_b(x)$.

Теперь на основе (2) и (3) для произвольной псевдобулевой функции $f\colon \mathbb{B}^n \to \mathbb{R}$ конструируем соответствующее ей выпуклое продолжение на $\mathbb{K}^n$.

Лемма 2. *Пусть $f\colon \mathbb{B}^n \to \mathbb{R}$ — произвольная псевдобулева функция, $f_{\min} = \min\limits_{b\in\mathbb{B}^n} f(b)$. Тогда вещественная функция*

$$f_C(x) = f_{\min} + \sum_{b\in\mathbb{B}^n}(f(b) - f_{\min})f_{DM}^b(x) \quad (4)$$

является выпуклым продолжением на $\mathbb{K}^n$ функции f.

Доказательство. Действительно, сначала заметим, что функция $f_C(x)$ выпукла по построению как сумма некоторых выпуклых (непрерывных) на множестве $\mathbb{K}^n$ функций. Пусть $x, y \in \mathbb{K}^n$ и $\alpha \in [0, 1]$. Тогда в силу выпуклости функции $f_{DM}^b(x)$ и неравенства $f(b) - f_{\min} \geqslant 0$ для $b \in \mathbb{B}^n$, имеем

$$\begin{aligned} f_C(\alpha x + (1-\alpha)y) &= f_{\min} + \sum_{b\in\mathbb{B}^n}(f(b) - f_{\min})f_{DM}^b(\alpha x + (1-\alpha)y) \leqslant \\ &\leqslant f_{\min} + \sum_{b\in\mathbb{B}^n}(f(b) - f_{\min})\big(\alpha f_{DM}^b(x) + (1-\alpha)f_{DM}^b(y)\big) = \\ &= \alpha f_{\min} + (1-\alpha)f_{\min} + \alpha\sum_{b\in\mathbb{B}^n}(f(b) - f_{\min})f_{DM}^b(x) + \\ &\quad + (1-\alpha)\sum_{b\in\mathbb{B}^n}(f(b) - f_{\min})f_{DM}^b(y) = \alpha f_C(x) + (1-\alpha)f_C(y). \end{aligned}$$

Остаётся показать, что $f_C(a) = f(a)$ для любого $a \in \mathbb{B}^n$. В силу (2)–(4) получаем

$$f_C(a) = f_{\min} + \sum_{b\in\mathbb{B}^n}(f(b) - f_{\min}) \cdot f_{DM}^b(a) =$$

$$= f_{\min} + (f(a) - f_{\min}) f^a_{DM}(a) + \sum_{b \in \mathbb{B}^n \setminus \{a\}} (f(b) - f_{\min}) f^b_{DM}(a) =$$

$$= f_{\min} + (f(a) - f_{\min}) \cdot 1 + \sum_{b \in \mathbb{B}^n \setminus \{a\}} (f(b) - f_{\min}) \cdot 0 = f(a).$$

Лемма 2 доказана.

Замечание 1. Сконструированное выпуклое продолжение (4) в общем случае не является максимумом среди всех выпуклых продолжений на $\mathbb{K}^n$ псевдобулевой функции f. Наглядным примером является булева функция от трёх переменных $f(x_1, x_2, x_3) = x_1x_2x_3 \vee \overline{x}_1x_2x_3$.

Далее сформулируем и докажем теорему о том, что для любой псевдобулевой функции f существует бесконечно много функций, каждая из которых является её выпуклым продолжением на $\mathbb{K}^n$.

Теорема 1. *Для произвольной псевдобулевой функции* $f\colon \mathbb{B}^n \to \mathbb{R}$ *существует бесконечно много её выпуклых продолжений на* $\mathbb{K}^n$.

Доказательство. Существование выпуклого продолжения на $\mathbb{K}^n$ псевдобулевой функции f доказано в лемме 2. Бесконечность множества таких продолжений докажем от противного.

Пусть имеется конечное множество $S_C = \{g_1, g_2, \ldots, g_N\}$ выпуклых продолжений на $\mathbb{K}^n$ функции f. Тогда найдётся $N_0 \in \{1, 2, \ldots, N\}$ такое, что

$$g_{N_0}\left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right) \leqslant g_k\left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right), \quad k \in \{1, 2, \ldots, N\}.$$

Рассмотрим функцию

$$g_{\text{new}}(x) = g_{N_0}(x) - A \min\{x_1, 1 - x_1, x_2, 1 - x_2, \ldots, x_n, 1 - x_n\},$$

где $A > 0$ — произвольное число. Докажем, что функция g_{new} также является выпуклым продолжением на $\mathbb{K}^n$ функции f. Для этого достаточно показать, что

1) $g_{\text{new}}(a) = f(a)$ для любого $a \in \mathbb{B}^n$;
2) функция g_{new} выпукла на множестве $\mathbb{K}^n$.

1) Действительно, для любого $a \in \mathbb{B}^n$

$$g_{\text{new}}(a) = g_{N_0}(a) - A \min\{a_1, 1 - a_1, a_2, 1 - a_2, \ldots, a_n, 1 - a_n\} =$$
$$= g_{N_0}(a) - A \cdot 0 = g_{N_0}(a) = f(a).$$

2) Пусть $x^*, x^{**} \in \mathbb{K}^n$, $\alpha \in [0, 1]$. Тогда для любого $k \in \{1, 2, \ldots, n\}$ справедливы неравенства

$$\alpha x^*_k + (1 - \alpha) x^{**}_k \geqslant \alpha \min\{x^*_1, 1 - x^*_1, \ldots, x^*_n, 1 - x^*_n\} +$$
$$+ (1 - \alpha) \min\{x^{**}_1, 1 - x^{**}_1, \ldots, x^{**}_n, 1 - x^{**}_n\},$$

$$\alpha(1-x_k^*)+(1-\alpha)(1-x_k^{**}) \geqslant \alpha\min\{x_1^*,1-x_1^*,\ldots,x_n^*,1-x_n^*\}+ \\ +(1-\alpha)\min\{x_1^{**},1-x_1^{**},\ldots,x_n^{**},1-x_n^{**}\}.$$

Следовательно,

$$\min\{\alpha x_k^*+(1-\alpha)x_k^{**},\,\alpha(1-x_k^*)+(1-\alpha)(1-x_k^{**})\}_{k=1}^n \geqslant \\ \geqslant \alpha\min\{x_1^*,1-x_1^*,\ldots,x_n^*,1-x_n^*\}+ \\ +(1-\alpha)\min\{x_1^{**},1-x_1^{**},\ldots,x_n^{**},1-x_n^{**}\}.$$

Отсюда в силу того, что $A>0$, и выпуклости $g_{N_0}(x)$ получаем

$$g_{\text{new}}(\alpha x^*+(1-\alpha)x^{**}) = g_{N_0}(\alpha x^*+(1-\alpha)x^{**}) - \\ -A\min\{\alpha x_k^*+(1-\alpha)x_k^{**},1-\alpha x_k^*-(1-\alpha)x_k^{**}\}_{k=1}^n = \\ = g_{N_0}(\alpha x^*+(1-\alpha)x^{**}) - \\ -A\min\{\alpha x_k^*+(1-\alpha)x_k^{**},\alpha(1-x_k^*)+(1-\alpha)(1-x_k^{**})\}_{k=1}^n \leqslant \\ \leqslant \alpha g_{N_0}(x^*)+(1-\alpha)g_{N_0}(x^{**})-\alpha A\min\{x_1^*,1-x_1^*,\ldots,x_n^*,1-x_n^*\}- \\ -(1-\alpha)A\min\{x_1^{**},1-x_1^{**},\ldots,x_n^{**},1-x_n^{**}\} = \\ = \alpha g_{\text{new}}(x^*)+(1-\alpha)g_{\text{new}}(x^{**}).$$

Далее, заметим, что $g_{\text{new}}(x)<g_{N_0}(x)$ при $x\in\operatorname{int}\mathbb{K}^n$, поскольку $A>0$ и $\min\{x_1,1-x_1,x_2,1-x_2,\ldots,x_n,1-x_n\}>0$ при $x\in\operatorname{int}\mathbb{K}^n$. Отсюда непосредственно следует, что

$$g_{\text{new}}\left(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2}\right)<g_{N_0}\left(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2}\right)$$

и тем самым в силу выбора N_0 продолжение g_{new} на $\mathbb{K}^n$ функции f не принадлежит множеству S_C; противоречие. Теорема 1 доказана.

Замечание 2. Теорема 1 также доказывает, что не существует минимального выпуклого продолжения на $\mathbb{K}^n$ произвольной псевдобулевой функции f.

Далее докажем конструктивно, что для любой псевдобулевой функции f существует единственный максимум f_{DM} среди всех её выпуклых продолжений на $\mathbb{K}^n$.

Теорема 2. *Для произвольной псевдобулевой функции $f\colon\mathbb{B}^n\to\mathbb{R}$ функция*

$$f_{DM}(x)=\min_{\lambda\in\Lambda_{\mathbb{K}^n}(x)}\sum_{b\in\mathbb{B}^n}\lambda_b f(b)$$

является единственным максимумом среди всех её выпуклых продолжений на $\mathbb{K}^n$.

Замечание 3. Функция f_{DM} корректно определена и непрерывна на $\mathbb{K}^n$ в силу компактности множества $\Lambda_{\mathbb{K}^n}(x)$ для любого $x \in \mathbb{K}^n$, непрерывности функции $\sum\limits_{b\in\mathbb{B}^n} \lambda_b f(b)$ и теоремы Вейерштрасса.

Доказательство теоремы 2. Сначала покажем, что для любого выпуклого продолжения g_C на $\mathbb{K}^n$ функции f имеет место неравенство

$$g_C(x) \leqslant f_{DM}(x), \quad x \in \mathbb{K}^n. \tag{5}$$

Действительно, если $x \in \mathbb{K}^n$, то $\Lambda_{\mathbb{K}^n}(x) \neq \varnothing$ в силу выпуклости множества $\mathbb{K}^n$. Ввиду выпуклости функции g_C и неравенства Йенсена [24] получаем

$$g_C(x) = g_C\Big(\sum_{b\in\mathbb{B}^n} \lambda_b b\Big) \leqslant \sum_{b\in\mathbb{B}^n} \lambda_b g_C(b) = \sum_{b\in\mathbb{B}^n} \lambda_b f(b)$$

для любого $\lambda \in \Lambda_{\mathbb{K}^n}(x)$. В частности,

$$g_C(x) \leqslant \min_{\lambda\in\Lambda_{\mathbb{K}^n}(x)} \sum_{b\in\mathbb{B}^n} \lambda_b f(b) = f_{DM}(x),$$

что доказывает справедливость (5).

Остаётся убедиться в том, что функция f_{DM} также является выпуклым продолжением f. Для этого достаточно показать, что

1) $f_{DM}(a) = f(a)$ для любого $a \in \mathbb{B}^n$;
2) функция f_{DM} выпукла на множестве $\mathbb{K}^n$.

1) Действительно, для любого $a \in \mathbb{B}^n$ имеем

$$f_{DM}(a) = \min_{\lambda\in\Lambda_{\mathbb{K}^n}(a)} \sum_{b\in\mathbb{B}^n} \lambda_b f(b) = f(a),$$

так как согласно следствию 2 множество $\Lambda_{\mathbb{K}^n}(a)$ состоит из одного вектора, в котором $\lambda_a = 1$, а на остальных местах нули.

2) Пусть $x^*, x^{**} \in \mathbb{K}^n$, $\alpha \in [0,1]$. По теореме Вейерштрасса существуют $\lambda^* \in \Lambda_{\mathbb{K}^n}(x^*)$ и $\lambda^{**} \in \Lambda_{\mathbb{K}^n}(x^{**})$ такие, что

$$f_{DM}(x^*) = \sum_{b\in\mathbb{B}^n} \lambda_b^* f(b), \quad f_{DM}(x^{**}) = \sum_{b\in\mathbb{B}^n} \lambda_b^{**} f(b).$$

Тогда

$$f_{DM}(\alpha x^* + (1-\alpha)x^{**}) = \min_{\lambda\in\Lambda_{\mathbb{K}^n}(\alpha x^*+(1-\alpha)x^{**})} \sum_{b\in\mathbb{B}^n} \lambda_b f(b) \leqslant$$

$$\leqslant \sum_{b\in\mathbb{B}^n} (\alpha\lambda_b^* + (1-\alpha)\lambda_b^{**}) f(b) = \alpha f_{DM}(x^*) + (1-\alpha) f_{DM}(x^{**}),$$

так как нетрудно заметить, что $\alpha\lambda^* + (1-\alpha)\lambda^{**} \in \Lambda_{\mathbb{K}^n}(\alpha x^* + (1-\alpha)x^{**})$. В силу произвольности x^*, x^{**} функция f_{DM} выпукла на $\mathbb{K}^n$.

Единственность максимума следует из (5) ввиду произвольности продолжения g_C на $\mathbb{K}^n$ функции f. Теорема 2 доказана.

3. Выпуклые продолжения дискретных функций вида $f\colon \mathbb{B}^n(a,b) \to \mathbb{R}$

В этом разделе конструктивно докажем, что для любой функции f, определённой на вершинах куба $\mathbb{K}^n(a,b)$, существует, во-первых, единственный максимум f_{DM} среди всех выпуклых продолжений на весь куб $\mathbb{K}^n(a,b)$ функции f, а во-вторых, бесконечно много таких продолжений.

Теорема 3. *Для произвольной дискретной функции $f\colon \mathbb{B}^n(a,b) \to \mathbb{R}$ функция*

$$f_{DM}(x) = \min_{\lambda \in \Lambda_{\mathbb{K}^n(a,b)}(x)} \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v f(v) \tag{6}$$

является единственным максимумом среди всех её выпуклых продолжений на $\mathbb{K}^n(a,b)$.

Замечание 4. Функция f_{DM} корректно определена и непрерывна на $\mathbb{K}^n(a,b)$. Обоснование этого аналогично обоснованию замечания 3.

Доказательство теоремы 3. Сначала покажем, что для любого выпуклого продолжения g_C на $\mathbb{K}^n(a,b)$ функции f имеет место неравенство

$$g_C(x) \leqslant f_{DM}(x), \quad x \in \mathbb{K}^n(a,b). \tag{7}$$

Действительно, если $x \in \mathbb{K}^n(a,b)$, то $\Lambda_{\mathbb{K}^n(a,b)}(x) \neq \varnothing$ ввиду выпуклости множества $\mathbb{K}^n(a,b)$. В силу выпуклости функции g_C и неравенства Йенсена [24] получаем

$$g_C(x) = g_C\Big(\sum_{v \in \mathbb{B}^n(a,b)} \lambda_v v\Big) \leqslant \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v g_C(v) = \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v f(v)$$

для любого $\lambda \in \Lambda_{\mathbb{K}^n(a,b)}(x)$. В частности,

$$g_C(x) \leqslant \min_{\lambda \in \Lambda_{\mathbb{K}^n(a,b)}(x)} \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v f(v) = f_{DM}(x),$$

что доказывает справедливость (7).

Остаётся убедиться в том, что функция f_{DM} также является выпуклым продолжением f. Для этого достаточно показать, что

1) $f_{DM}(u) = f(u)$ для любого $u \in \mathbb{B}^n(a,b)$;
2) функция f_{DM} выпукла на множестве $\mathbb{K}^n(a,b)$.

1) Действительно, для любого $u \in \mathbb{B}^n(a,b)$ имеем

$$f_{DM}(u) = \min_{\lambda \in \Lambda_{\mathbb{K}^n(a,b)}(u)} \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v f(v) = f(u),$$

так как согласно следствию 1 множество $\Lambda_{\mathbb{K}^n(a,b)}(u)$ состоит из одного вектора, в котором $\lambda_u = 1$, а на остальных местах нули.

2) Пусть $x^*, x^{**} \in \mathbb{K}^n(a,b)$, $\alpha \in [0,1]$. По теореме Вейерштрасса существуют $\lambda^* \in \Lambda_{\mathbb{K}^n(a,b)}(x^*)$ и $\lambda^{**} \in \Lambda_{\mathbb{K}^n(a,b)}(x^{**})$ такие, что

$$f_{DM}(x^*) = \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v^* f(v), \quad f_{DM}(x^{**}) = \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v^{**} f(v).$$

Тогда

$$f_{DM}(\alpha x^* + (1-\alpha)x^{**}) = \min_{\lambda \in \Lambda_{\mathbb{K}^n(a,b)}(\alpha x^* + (1-\alpha)x^{**})} \sum_{v \in \mathbb{B}^n(a,b)} \lambda_v f(v) \leqslant$$

$$\leqslant \sum_{v \in \mathbb{B}^n(a,b)} (\alpha\lambda_v^* + (1-\alpha)\lambda_v^{**}) f(v) = \alpha f_{DM}(x^*) + (1-\alpha) f_{DM}(x^{**}),$$

так как нетрудно заметить, что $\alpha\lambda^* + (1-\alpha)\lambda^{**} \in \Lambda_{\mathbb{K}^n(a,b)}(\alpha x^* + (1-\alpha)x^{**})$. В силу произвольности x^*, x^{**} функция f_{DM} выпукла на $\mathbb{K}^n(a,b)$.

Единственность максимума следует из (7) в силу произвольности продолжения g_C на $\mathbb{K}^n(a,b)$ функции f. Теорема 3 доказана.

Теорема 4. *Для произвольной дискретной функции $f\colon \mathbb{B}^n(a,b) \to \mathbb{R}$ существует бесконечно много её выпуклых продолжений на $\mathbb{K}^n(a,b)$.*

Доказательство. Существование для функции f выпуклого продолжения на $\mathbb{K}^n(a,b)$ доказано в теореме 3, согласно которой таковым является функция f_{DM}, определённая в (6). Бесконечность множества таких продолжений докажем от противного.

Пусть имеется конечное множество $S_C = \{g_1, g_2, \ldots, g_N\}$ выпуклых продолжений на $\mathbb{K}^n(a,b)$ функции f. Тогда найдётся $N_0 \in \{1, 2, \ldots, N\}$ такое, что для любого $k \in \{1, 2, \ldots, N\}$ имеем

$$g_{N_0}\Big(\frac{a+b}{2}, \frac{a+b}{2}, \ldots, \frac{a+b}{2}\Big) \leqslant g_k\Big(\frac{a+b}{2}, \frac{a+b}{2}, \ldots, \frac{a+b}{2}\Big).$$

Рассмотрим функцию

$$g_{\text{new}}(x) = g_{N_0}(x) - A \min\{x_k - a,\, b - x_k\}_{k=1}^n,$$

где $A > 0$ — произвольное число. Аналогично доказательству бесконечности в теореме 1 нетрудно показать, что, с одной стороны, функция g_{new} также является выпуклым продолжением на $\mathbb{K}^n(a,b)$ функции f, а с другой стороны, в силу выбора N_0 это выпуклое продолжение не принадлежит множеству S_C; противоречие. Теорема 4 доказана.

Замечание 5. Теорема 4 также доказывает, что не существует минимального выпуклого продолжения на $\mathbb{K}^n(a,b)$ произвольной дискретной функции $f\colon \mathbb{B}^n(a,b) \to \mathbb{R}$.

4. Выпуклые продолжения дискретных функций вида $f\colon \mathbb{BP}^n \to \mathbb{R}$

В этом разделе конструктивно докажем, что для любой функции f, определённой на вершинах параллелепипеда $\mathbb{P}^n$, существуют, во-первых, единственный максимум f_{DM} среди всех выпуклых продолжений на весь параллелепипед $\mathbb{P}^n$ функции f а во-вторых, бесконечно много таких продолжений. Напомним, что параллелепипед $\mathbb{P}^n$ определяется парой различных точек $c = (c_1, c_2, \ldots, c_n)$, $d = (d_1, d_2, \ldots, d_n) \in \mathbb{R}^n$ таких, что $c_k < d_k$, $k \in \{1, 2, \ldots, n\}$.

Теорема 5. *Для произвольной дискретной функции $f\colon \mathbb{BP}^n \to \mathbb{R}$ функция*

$$f_{DM}(x) = \min_{\lambda \in \Lambda_{\mathbb{P}^n}(x)} \sum_{v \in \mathbb{BP}^n} \lambda_v f(v) \tag{8}$$

является единственным максимумом среди всех её выпуклых продолжений на $\mathbb{P}^n$.

Замечание 6. Функция f_{DM} корректно определена и непрерывна на $\mathbb{P}^n$. Обоснование этого аналогично обоснованию замечания 3.

Доказательство теоремы 5. Сначала покажем, что для любого выпуклого продолжения g_C на $\mathbb{P}^n$ функции f имеет место неравенство

$$g_C(x) \leqslant f_{DM}(x), \quad x \in \mathbb{P}^n. \tag{9}$$

Действительно, если $x \in \mathbb{P}^n$, то $\Lambda_{\mathbb{P}^n}(x) \neq \varnothing$ в силу выпуклости множества $\mathbb{P}^n$. Ввиду выпуклости функции g_C и неравенства Йенсена [24] получаем

$$g_C(x) = g_C\Big(\sum_{v \in \mathbb{BP}^n} \lambda_v v\Big) \leqslant \sum_{v \in \mathbb{BP}^n} \lambda_v g_C(v) = \sum_{v \in \mathbb{BP}^n} \lambda_v f(v)$$

для любого $\lambda \in \Lambda_{\mathbb{P}^n}(x)$. В частности,

$$g_C(x) \leqslant \min_{\lambda \in \Lambda_{\mathbb{P}^n}(x)} \sum_{v \in \mathbb{BP}^n} \lambda_v f(v) = f_{DM}(x),$$

что доказывает справедливость (9).

Остаётся убедиться в том, что функция f_{DM} также является выпуклым продолжением f. Для этого достаточно доказать, что

1) $f_{DM}(u) = f(u)$ для любого $u \in \mathbb{BP}^n$;
2) функция f_{DM} выпукла на множестве $\mathbb{P}^n$.

1) Действительно, для любого $u \in \mathbb{BP}^n$ имеем

$$f_{DM}(u) = \min_{\lambda \in \Lambda_{\mathbb{P}^n}(u)} \sum_{v \in \mathbb{BP}^n} \lambda_v f(v) = f(u),$$

так как согласно лемме 1 множество $\Lambda_{\mathbb{P}^n}(u)$ состоит из одного вектора, в котором $\lambda_u = 1$, а на остальных местах нули.

2) Пусть $x^*, x^{**} \in \mathbb{P}^n$, $\alpha \in [0,1]$. По теореме Вейерштрасса существуют $\lambda^* \in \Lambda_{\mathbb{P}^n}(x^*)$ и $\lambda^{**} \in \Lambda_{\mathbb{P}^n}(x^{**})$ такие, что

$$f_{DM}(x^*) = \sum_{v \in \mathbb{BP}^n} \lambda_v^* f(v), \quad f_{DM}(x^{**}) = \sum_{v \in \mathbb{BP}^n} \lambda_v^{**} f(v).$$

Тогда

$$f_{DM}(\alpha x^* + (1-\alpha)x^{**}) = \min_{\lambda \in \Lambda_{\mathbb{P}^n}(\alpha x^* + (1-\alpha)x^{**})} \sum_{v \in \mathbb{BP}^n} \lambda_v f(v) \leqslant$$

$$\leqslant \sum_{v \in \mathbb{BP}^n} (\alpha\lambda_v^* + (1-\alpha)\lambda_v^{**}) f(v) = \alpha f_{DM}(x^*) + (1-\alpha) f_{DM}(x^{**}),$$

так как нетрудно заметить, что $\alpha\lambda^* + (1-\alpha)\lambda^{**} \in \Lambda_{\mathbb{P}^n}(\alpha x^* + (1-\alpha)x^{**})$. В силу произвольности x^*, x^{**} функция f_{DM} выпукла на $\mathbb{P}^n$.

Единственность максимума следует из (9) в силу произвольности продолжения g_C на $\mathbb{P}^n$ функции f. Теорема 5 доказана.

Теорема 6. *Для произвольной дискретной функции $f\colon \mathbb{BP}^n \to \mathbb{R}$ существует бесконечно много её выпуклых продолжений на $\mathbb{P}^n$.*

Доказательство. Существование для функции f выпуклого продолжения на $\mathbb{P}^n$ доказано в теореме 5, согласно которой таковым является функция f_{DM}, определённая в (8). Бесконечность множества таких продолжений докажем от противного.

Пусть имеется конечное множество $S_C = \{g_1, g_2, \ldots, g_N\}$ выпуклых продолжений на $\mathbb{P}^n$ функции f. Тогда найдётся $N_0 \in \{1, 2, \ldots, N\}$ такое, что для любого $k \in \{1, 2, \ldots, N\}$ имеем

$$g_{N_0}\left(\frac{c+d}{2}\right) \leqslant g_k\left(\frac{c+d}{2}\right).$$

Рассмотрим функцию

$$g_{\text{new}}(x) = g_{N_0}(x) - A \min\{x_k - c_k,\, d_k - x_k\}_{k=1}^n,$$

где $A > 0$ — произвольное число. Аналогично доказательству бесконечности в теореме 1 нетрудно показать, что, с одной стороны, функция g_{new} также является выпуклым продолжением на $\mathbb{P}^n$ функции f, а с другой стороны, в силу выбора N_0 это выпуклое продолжение не принадлежит множеству S_C; противоречие. Теорема 6 доказана.

Замечание 7. Теорема 6 также доказывает, что не существует минимального выпуклого продолжения на $\mathbb{P}^n$ произвольной дискретной функции $f\colon \mathbb{BP}^n \to \mathbb{R}$.

Заключение

В настоящей работе рассмотрены выпуклые продолжения дискретных функций, заданных на вершинах n-мерного единичного куба $\mathbb{K}^n$, произвольного куба $\mathbb{K}^n(a,b)$ и параллелепипеда $\mathbb{P}^n$. В каждом конкретном случае конструктивно доказано, что для произвольной дискретной функции f, определённой на вершинах $\mathbb{G}$, где $\mathbb{G} \in \{\mathbb{K}^n, \mathbb{K}^n(a,b), \mathbb{P}^n\}$, во-первых, существует бесконечно много её выпуклых продолжений на множество $\mathbb{G}$, а во-вторых, указана функция вида $f_{DM}\colon \mathbb{G} \to \mathbb{R}$, которая является единственным максимумом среди всех выпуклых продолжений f на $\mathbb{G}$. Обосновано также, что функция f_{DM} непрерывна на $\mathbb{G}$.

Финансирование работы

Исследование выполнено за счёт бюджета Финансового университета при Правительстве Российской Федерации. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Armario J. A.** Boolean functions and permanents of Sylvester Hadamard matrices // Mathematics. 2021. V. 9, No. 2. Paper ID 177. 8 p. DOI: 10.3390/math9020177.
2. **Abdel-Gawad A. H., Atiya A. F., Darwish N. M.** Solution of systems of Boolean equations via the integer domain // Inf. Sci. 2010. V. 180, No. 2. P. 288–300. DOI: 10.1016/j.ins.2009.09.010.
3. **Brown F. M.** Boolean reasoning: The logic of Boolean equations. Boston: Kluwer Acad. Publ., 1990. 304 p.
4. **Hammer P. L., Rudeanu S.** Boolean methods in operations research and related areas. Heidelberg: Springer, 1968. 330 p.
5. **Bard G. V.** Algorithms for solving linear and polynomial systems of equations over finite fields, with applications to cryptanalysis: PhD Thes. College Park, MD: Univ. Maryland, 2007. 178 p.
6. **Faugère J.-C., Joux A.** Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases // Advances in cryptology — CRYPTO 2003. Proc. 23rd Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 17–21, 2003). Heidelberg: Springer, 2003. P. 44–60. (Lect. Notes Comput. Sci.; V. 2729). DOI: 10.1007/978-3-540-45146-4_3.
7. **Armknecht F.** Improving fast algebraic attacks // Fast software encryption. Rev. Pap. 11th Int. Workshop (Delhi, India, Feb. 5–7, 2004). Heidelberg: Springer, 2004. P. 65–82. DOI: 10.1007/978-3-540-25937-4_5.

8. **Bardet M., Faugère J.-C., Salvye B., Spaenlehauer P. J.** On the complexity of solving quadratic boolean systems // J. Complex. 2013. V. 29. P. 53–75. DOI: 10.1016/j.jco.2012.07.001.
9. **Courtois N. T.** Fast algebraic attacks on stream ciphers with linear feedback // Advances in cryptology — CRYPTO 2003. Proc. 23rd Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 17–21, 2003). Heidelberg: Springer, 2003. P. 176–194. (Lect. Notes Comput. Sci.; V. 2729). DOI: 10.1007/978-3-540-45146-4_11.
10. **Faugère J.-C.** A new efficient algorithm for computing Gröbner bases (F4) // J. Pure Appl. Algebra. 1999. V. 139. P. 61–88. DOI: 10.1016/S0022-4049(99)00005-5.
11. **Faugère J.-C.** A new efficient algorithm for computing Gröbner bases without reduction to zero (F5) // Proc. 2002 Int. Symp. Symbolic and Algebraic Computation (Lille, France, July 7–10, 2002). New York: ACM, 2002. P. 75–83. DOI: 10.1145/780506.780516.
12. **Liu M., Lin D., Pei D.** Fast algebraic attacks and decomposition of symmetric Boolean functions // IEEE Trans. Inf. Theory. 2011. V. 57. P. 4817–4821. DOI: 10.1109/TIT.2011.2145690.
13. **Файзуллин Р. Т., Дулькейт В. И., Огородников Ю. Ю.** Гибридный метод поиска приближённого решения задачи 3-выполнимость, ассоциированной с задачей факторизации // Тр. Ин-та математики и механики. 2013. Т. 19, № 2. С. 285–294.
14. **Gu J.** Global optimization for satisfiability (SAT) problem // IEEE Trans. Knowl. Data Eng. 1994. V. 6, No. 3. P. 361–381. DOI: 10.1109/69.334864.
15. **Gu J., Gu Q., Du D.** On optimizing the satisfiability (SAT) problem // J. Comput. Sci. Technol. 1999. V. 14, No. 1. P. 1–17. DOI: 10.1007/BF02952482.
16. **Pakhomchik A. I., Voloshinov V. V., Vinokur V. M., Lesovik G. B.** Converting of Boolean expression to linear equations, inequalities and QUBO penalties for cryptanalysis // Algorithms. 2022. V. 15, No. 2. Paper ID 33. 22 p. DOI: 10.3390/a15020033.
17. **Barotov D. N., Barotov R. N., Soloviev V., Feklin V., Muzafarov D., Ergashboev T., Egamov Kh.** The development of suitable inequalities and their application to systems of logical equations // Mathematics. 2022. V. 10, No. 11. Paper ID 1851. 9 p. DOI: 10.3390/math10111851.
18. **Баротов Д. Н., Баротов Р. Н.** Полилинейные продолжения некоторых дискретных функций и алгоритм их нахождения // Вычисл. методы и программирование. 2023. Т. 24, вып. 1. С. 10–23. DOI: 10.26089/NumMet.v24r102.
19. **Barotov D. N., Osipov A., Korchagin S., Pleshakova E., Muzafarov D., Barotov R. N., Serdechnyi D.** Transformation method for solving system of Boolean algebraic equations // Mathematics. 2021. V. 9, No. 24. Paper ID 3299. 12 p. DOI: 10.3390/math9243299.
20. **Owen G.** Multilinear extensions of games // Manage. Sci. 1972. V. 18, No. 5-2. P. 64–79. DOI: 10.1287/mnsc.18.5.64.

21. Barotov D. N., Barotov R. N. Polylinear transformation method for solving systems of logical equations // Mathematics. 2022. V. 10, No. 6. Paper ID 918. 10 p. DOI: 10.3390/math10060918.

22. Barotov D. N. Target function without local minimum for systems of logical equations with a unique solution // Mathematics. 2022. V. 10, No. 12. Paper ID 2097. 8 p. DOI: 10.3390/math10122097.

23. Баротов Д. Н. Выпуклое продолжение булевой функции и его приложения // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 1. С. 5–18.

24. Jensen J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes // Acta Math. 1906. V. 30. P. 175–193. [French]. DOI: 10.1007/BF02418571.

Баротов Достонжон Нумонжонович

Статья поступила
7 декабря 2023 г.
После доработки —
12 февраля 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 5–23

UDC 519.8+518.25 DOI: 10.33048/daio.2024.31.789

CONVEX CONTINUATIONS OF SOME DISCRETE FUNCTIONS

D. N. Barotov

Financial University under the Government of the Russian Federation,
4 Chetvyortyi Veshnyakovskii Passage, 109456 Moscow, Russia

E-mail: dnbarotov@fa.ru

Abstract. We construct convex continuations of discrete functions defined on the vertices of the n-dimensional unit cube $[0,1]^n$, an arbitrary cube $[a,b]^n$, and a parallelepiped $[c_1,d_1]\times[c_2,d_2]\times\cdots\times[c_n,d_n]$. In each of these cases, we constructively prove that, for any discrete function f defined on the vertices of $\mathbb{G}\in\{[0,1]^n,[a,b]^n,[c_1,d_1]\times[c_2,d_2]\times\cdots\times[c_n,d_n]\}$, first, there exist infinitely many convex continuations to the set $\mathbb{G}$, and second, there exists a unique function $f_{DM}\colon\mathbb{G}\to\mathbb{R}$ that is the maximum of convex continuations of f to $\mathbb{G}$. We also show that the function f_{DM} is continuous on $\mathbb{G}$. Bibliogr. 24.

Keywords: discrete function, convex continuation of a discrete function, Boolean function, pseudo-Boolean function.

References

1. **J. A. Armario,** Boolean functions and permanents of Sylvester Hadamard matrices, *Mathematics* **9** (2), ID 177 (2021), DOI: 10.3390/math9020177.
2. **A. H. Abdel-Gawad, A. F. Atiya,** and **N. M. Darwish,** Solution of systems of Boolean equations via the integer domain, *Inf. Sci.* **180** (2), 288–300 (2010), DOI: 10.1016/j.ins.2009.09.010.
3. **F. M. Brown,** *Boolean Reasoning: The Logic of Boolean Equations* (Kluwer Acad. Publ., Boston, 1990).
4. **P. L. Hammer** and **S. Rudeanu,** *Boolean Methods in Operations Research and Related Areas* (Springer, Heidelberg, 1968).
5. **G. V. Bard,** Algorithms for solving linear and polynomial systems of equations over finite fields, with applications to cryptanalysis, *PhD Thesis* (Univ. Maryland, College Park, MD, 2007).

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 412–423 (2024), DOI: 10.1134/S1990478924030049.

6. **J.-C. Faugère** and **A. Joux,** Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases, in *Advances in Cryptology — CRYPTO 2003* (Proc. 23rd Annu. Int. Cryptology Conf., Santa Barbara, USA, Aug. 17–21, 2003) (Springer, Heidelberg, 2003), pp. 44–60 (Lect. Notes Comput. Sci., Vol. 2729), DOI: 10.1007/978-3-540-45146-4_3.
7. **F. Armknecht,** Improving fast algebraic attacks, in *Fast Software Encryption* (Rev. Pap. 11th Int. Workshop, Delhi, India, Feb. 5–7, 2004) (Springer, Heidelberg, 2004), pp. 65–82, DOI: 10.1007/978-3-540-25937-4_5.
8. **M. Bardet, J.-C. Faugère, B. Salvye,** and **P. J. Spaenlehauer,** On the complexity of solving quadratic boolean systems, *J. Complex.* **29**, 53–75 (2013), DOI: 10.1016/j.jco.2012.07.001.
9. **N. T. Courtois,** Fast algebraic attacks on stream ciphers with linear feedback, in *Advances in Cryptology — CRYPTO 2003* (Proc. 23rd Annu. Int. Cryptology Conf., Santa Barbara, USA, Aug. 17–21, 2003) (Springer, Heidelberg, 2003), pp. 176–194 (Lect. Notes Comput. Sci., Vol. 2729), DOI: 10.1007/978-3-540-45146-4_11.
10. **J.-C. Faugère,** A new efficient algorithm for computing Gröbner bases (F4), *J. Pure Appl. Algebra* **139**, 61–88 (1999), DOI: 10.1016/S0022-4049(99)00005-5.
11. **J.-C. Faugère,** A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), in *Proc. 2002 Int. Symp. Symbolic and Algebraic Computation, Lille, France, July 7–10, 2002* (ACM, New York, 2002), pp. 75–83, DOI: 10.1145/780506.780516.
12. **M. Liu, D. Lin,** and **D. Pei,** Fast algebraic attacks and decomposition of symmetric Boolean functions, *IEEE Trans. Inf. Theory* **57**, 4817–4821 (2011), DOI: 10.1109/TIT.2011.2145690.
13. **R. T. Faizullin, V. I. Dul'keit,** and **Yu. Yu. Ogorodnikov,** A hybrid method for the approximate solution of the 3-satisfiability problem associated with the factorization problem, *Tr. Inst. Mat. Mekh.* **19** (2), 285–294 (2013) [Russian].
14. **J. Gu,** Global optimization for satisfiability (SAT) problem, *IEEE Trans. Knowl. Data Eng.* **6** (3), 361–381 (1994), DOI: 10.1109/69.334864.
15. **J. Gu, Q. Gu,** and **D. Du,** On optimizing the satisfiability (SAT) problem, *J. Comput. Sci. Technol.* **14** (1), 1–17 (1999), DOI: 10.1007/BF02952482.
16. **A. I. Pakhomchik, V. V. Voloshinov, V. M. Vinokur,** and **G. B. Lesovik,** Converting of Boolean expression to linear equations, inequalities and QUBO penalties for cryptanalysis, *Algorithms* **15** (2), ID 33 (2022), DOI: 10.3390/a15020033.
17. **D. N. Barotov, R. N. Barotov, V. Soloviev, V. Feklin, D. Muzafarov, T. Ergashboev,** and **Kh. Egamov,** The development of suitable inequalities and their application to systems of logical equations, *Mathematics* **10** (11), ID 1851 (2022), DOI: 10.3390/math10111851.
18. **D. N. Barotov** and **R. N. Barotov,** Polylinear continuations of some discrete functions and an algorithm for finding them, *Vychisl. Metody Program.* **24** (1), 10–23 (2023) [Russian], DOI: 10.26089/NumMet.v24r102.

19. D. N. Barotov, A. Osipov, S. Korchagin, E. Pleshakova, D. Muzafarov, R. N. Barotov, and **D. Serdechnyi,** Transformation method for solving system of Boolean algebraic equations, *Mathematics* **9** (24), ID 3299 (2021), DOI: 10.3390/math9243299.

20. G. Owen, Multilinear extensions of games, *Manage. Sci.* **18** (5-2), 64–79 (1972), DOI: 10.1287/mnsc.18.5.64.

21. D. N. Barotov and **R. N. Barotov,** Polylinear transformation method for solving systems of logical equations, *Mathematics* **10** (6), ID 918 (2022), DOI: 10.3390/math10060918.

22. D. N. Barotov, Target function without local minimum for systems of logical equations with a unique solution, *Mathematics* **10** (12), ID 2097 (2022), DOI: 10.3390/math10122097.

23. D. N. Barotov, Convex continuation of a Boolean function and its applications, *Diskretn. Anal. Issled. Oper.* **31** (1), 5–18 (2024) [Russian] [*J. Appl. Ind. Math.* **18** (1), 1–9 (2024)].

24. J. L. W. V. Jensen, Sur les fonctions convexes et les inégalités entre les valeurs moyennes, *Acta Math.* **30**, P. 175–193 (1906) [French], DOI: 10.1007/BF02418571.

Dostonjon N. Barotov

Received December 7, 2023
Revised February 12, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 24–53

УДК 519.7 DOI: 10.33048/daio.2024.31.777

НОВАЯ МОДЕЛЬ КВАНТОВОГО ОРАКУЛА ДЛЯ ГИБРИДНОЙ КВАНТОВО-КЛАССИЧЕСКОЙ АТАКИ НА ПОСТКВАНТОВЫЕ КРИПТОСИСТЕМЫ, ОСНОВАННЫЕ НА РЕШЁТКАХ

А. О. Бахарев

Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

E-mail: a.bakharev@g.nsu.ru

Аннотация. Криптосистемы на основе решёток являются одними из основных постквантовых альтернатив асимметричной криптографии, используемой в настоящее время. Большинство атак на такие криптосистемы можно свести к задаче нахождения кратчайшего вектора в решётке (SVP). Ранее авторами была предложена модель квантового оракула из алгоритма Гровера для реализации гибридного квантово-классического алгоритма, основанного на алгоритме GaussSieve и решающего SVP. В настоящей работе предложена и проанализирована новая модель квантового оракула. Предложены и оценены две реализации новой модели квантового оракула. Проанализирована сложность реализации новой модели квантового оракула для атаки на постквантовые криптосистемы, основанные на решётках и являющиеся финалистами конкурса постквантовой криптографии NIST. Приведено сравнение полученных результатов для новой и уже существующей моделей квантового оракула. Табл. 4, ил. 10, библиогр. 48.

Ключевые слова: квантовый поиск, криптография с открытым ключом, криптография на решётках, постквантовая криптография, алгоритм Гровера, квантовые вычисления.

Введение

Развитие квантовых вычислений ведёт к необходимости в разработке и анализе криптосистем, устойчивых к атакам с использованием квантовых компьютеров — алгоритмов постквантовой криптографии [1–3]. Программа на квантовом компьютере может быть представлена квантовой

схемой. Ключевыми параметрами квантовых схем являются число кубитов, число вентилей и глубина схемы. Существует ряд открытых вопросов, связанных с квантовыми схемами, таких, как оценки сложности реализации квантовых схем, оптимальная реализация квантовых схем и др.

В 2016 г. Национальный институт стандартов и технологий США (NIST) опубликовал отчёт, в котором было проанализировано влияние квантовых вычислений на действующие стандарты шифрования. Согласно выводам отчёта симметричные криптосистемы (AES [4]) и хеш-функции (SHA-2, SHA-3 [5]) требуют увеличения размерностей ключей и входных последовательностей, а асимметричные криптосистемы (RSA [6], ECDSA, ECDH, DSA [7]) не являются постквантовыми. Большое влияние на отсутствие стойкости действующих стандартов асимметричного шифрования относительно квантовых вычислений приписывается алгоритму Шора [8], решающему задачи дискретного логарифмирования и факторизации за полиномиальное от длины двоичной записи порядка группы и факторизуемого числа время соответственно. Вследствие этого в том же году NIST объявил о начале конкурса «Post-Quantum Cryptography Competition», направленного на выявление новых — квантово устойчивых — стандартов цифровой подписи и инкапсуляции ключа. 5 июля 2022 г. завершился третий этап конкурса NIST [9], по итогам которого стандартами были выбраны криптосистемы, основанные на теории решёток и хеш-функциях. В феврале 2022 г. начался конкурс «Korean Post-Quantum Cryptography Competition», направленный на стандартизацию постквантовой криптографии в Южной Корее [10].

Настоящая работа посвящена криптоанализу криптосистем, основанных на решётках. Одной из задач в теории решёток является задача нахождения кратчайшего вектора (SVP), которая заключается в нахождении в заданной своим базисом решётке ненулевого вектора, имеющего наименьшую длину. В общем случае SVP является NP-трудной задачей [11]. Большинство атак на решётчатые криптосистемы сводится к нахождению вектора из решётки, отношение длины которого к длине вектора, являющегося решением SVP, не превосходит некоторого полинома от размерности решётки. Существует множество алгоритмов, решающих SVP, таких, как алгоритмы перечисления [12–14], алгоритмы редукции базиса [15], их комбинация [16–18], алгоритмы просеивания [19–23] и др. [24–26].

Известен ряд работ, направленных на разработку и анализ алгоритмов квантового криптоанализа, как симметричных [27–35], так и асимметричных криптосистем [36–39]. На данный момент остаётся открытым вопрос точных оценок параметров квантовых схем, используемых при квантовом криптоанализе. Для алгоритма GaussSieve в работе [40] уже

рассматривался данный вопрос, и была предложена модель квантового оракула, хранящая список векторов в квантовой памяти. В настоящей работе предлагается новая модель квантового оракула, хранящая список векторов в классической памяти. Под классической памятью подразумевается память, используемая обычным, не квантовым компьютером.

1. Основные определения и понятия

Пусть $\mathbb{F}_2$ — поле, состоящее из двух элементов, а $\mathbb{F}_2^n$ — векторное пространство размерности n над полем $\mathbb{F}_2$. *Весом* $wt(x)$ двоичного вектора $x \in \mathbb{F}_2^n$ называется число его ненулевых координат. Введём отношение частичного порядка $\preceq$ на множестве $\mathbb{F}_2^n$. Для $x, y \in \mathbb{F}_2^n$ положим

$$x \preceq y \Leftrightarrow x_i \leqslant y_i \quad \text{для любого } i \in \overline{1,n}.$$

Произвольная функция $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ называется *булевой функцией от n переменных*. Через $\oplus$ будем обозначать операцию сложения по модулю 2, т. е. $a \oplus b = (a + b) \bmod 2$. Любая булева функция f от n переменных единственным образом представляется в виде

$$f(x_1, \ldots, x_n) = \bigoplus_{k=1}^{n} \bigoplus_{i_1,\ldots,i_k} a_{i_1,\ldots,i_k} x_{i_1} \cdots x_{i_k} \oplus a_0,$$

где при каждом k все индексы $i_1, \ldots, i_k$ различны и $a_0, a_{i_1,\ldots,i_k} \in \mathbb{F}_2$. Такое представление называется *полиномом Жегалкина* или *алгебраической нормальной формой* функции f. *Степенью* булевой функции называется число переменных в самом длинном слагаемом АНФ булевой функции f. *Векторной булевой функцией* называется произвольное отображение $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$. Векторную булеву функцию $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ можно представить в виде $F(x) = (f_1(x), \ldots, f_m(x))$, где f_j — булева функция от n переменных. Функции f_j называют *координатными*.

1.1. Решётки. В настоящей работе рассматриваются криптосистемы, основанные на задачах из теории решёток.

Определение 1. Пусть $u_1, \ldots, u_d \in \mathbb{R}^n$, $d \leqslant n$, — линейно независимые векторы. *Решёткой* размерности d называется множество

$$\Lambda = \left\{ \sum_{i=1}^{d} b_i u_i \mid b_i \in \mathbb{Z} \right\}.$$

Линейно независимая система векторов, порождающая решётку, называется *базисом решётки*.

Пусть $p \geqslant 1$ — вещественное число. Тогда для вектора $x = (x_1, \ldots, x_n)$ из $\mathbb{R}^n$ *норма* l_p равна

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

Определение 2. *Задача нахождения кратчайшего вектора* (SVP) — найти в заданной своим базисом решётке ненулевой вектор, имеющий наименьшую длину относительно нормы l_p.

В данной работе рассматривается вариант SVP в евклидовой норме l_2. Далее для удобства вместо $\|\cdot\|_2$ будем писать $\|\cdot\|$.

В 2010 г. в [21] был предложен алгоритм GaussSieve, решающий SVP.

Алгоритм 1. Алгоритм GaussSieve (Миччанчо, Вулгарис, 2010)

Вход: B — базис решётки.
Выход: v — кратчайший вектор решётки.

1: Инициализировать пустой неупорядоченный список L и пустой стек S;
2: **repeat**
3: получить вектор v из стека (или сгенерировать новый);
4: **while** $w \leftarrow$ ПОИСК$\{w \in L \mid \|v \pm w\| \leqslant \|v\|\}$ **do**
5: уменьшить v с помощью w $(v \leftarrow v \pm w)$;
6: **while** $w \leftarrow$ ПОИСК$\{w \in L \mid \|w \pm v\| \leqslant \|w\|\}$ **do**
7: удалить w из списка L;
8: уменьшить w с помощью v $(w \leftarrow w \pm v)$;
9: добавить w в стек S;
10: **if** v изменился, **then** добавить v в стек S;
11: **else** добавить v в список L;
12: **until** v — кратчайший вектор;
13: **return** вектор v;

На вход алгоритма 1 поступает базис решётки, на основе которого будут строиться новые векторы при условии пустого стека S. Функция ПОИСК перебирает векторы w в списке и проверяет их на одно из условий поиска:

$$\|v \pm w\| \leqslant \|v\| \text{ или } \|w \pm v\| \leqslant \|w\|. \tag{1}$$

Если такой вектор существует, то функция возвращает его, иначе она прерывает первый цикл, в котором находится. Авторами [21] предложено эвристическое условие остановки, основанное на численных экспериментах. Таким образом, алгоритм работает до тех пор, пока не сработает эвристическое условие остановки.

Так как длина списка L увеличивается экспоненциально с ростом размерности решётки, самой трудозатратной операцией данного алгоритма является функция ПОИСК. В рамках предложенного в [38] подхода ускорение достигается за счёт использования в функции ПОИСК квантового алгоритма поиска.

1.2. Задача поиска. Задача, решаемая в функции ПОИСК, называется *задачей поиска*. Предполагается, что есть неупорядоченный список из K элементов, в котором, как минимум, один элемент удовлетворяет некоторому условию. Требуется найти по крайней мере один такой элемент. Другими словами, определена булева функция f, которая по номеру элемента (его двоичному представлению x) определяет, является ли элемент подходящим. Если элемент подходящий, то $f(x) = 1$, иначе $f(x) = 0$. В такой постановке задача поиска сводится к нахождению решения уравнения $f(x) = 1$.

В классическом варианте при условии, что решение одно, требуется $\sim K/2$ обращений к функции f для нахождения решения. Квантовый алгоритм поиска элемента в неупорядоченном списке (алгоритм Гровера [41]) решает данную задачу за $\sim \frac{\pi}{4}\sqrt{K}$ обращений к *оракулу* — квантовому аналогу функции f. О том, как булева функция моделируется на квантовом компьютере, будет рассказано далее.

2. Квантовые вычисления

Далее будут изложены необходимые сведения о квантовых вычислениях и принципах их работы. Более подробную информацию можно найти в работах [42, 43].

2.1. Кубит. Квантовый компьютер, в отличие от обычного, оперирует *квантовыми битами* (*кубитами*). Подобно классическому биту, который может находиться в состоянии 0 или 1, кубит имеет возможные состояния $|0\rangle$ и $|1\rangle$. Здесь используется *дираковское обозначение* $|\cdot\rangle$, которое является стандартным обозначением состояния в квантовой механике. Различие между битами и кубитами в том, что кубит может находиться в состоянии, отличном от $|0\rangle$ или $|1\rangle$. Можно составить *линейную комбинацию* состояний (*суперпозицию*):

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Числа α и β комплексные, и $|\alpha|^2+|\beta|^2 = 1$. Иначе говоря, состояние одного кубита можно представить как единичный вектор из $\mathbb{C}^2$. Однако мы не можем измерить кубит, чтобы определить его квантовое состояние, т. е. значения α и β. Из квантовой механики следует, что при измерении кубита получаем либо результат $|0\rangle$ с вероятностью $|\alpha|^2$, либо результат $|1\rangle$ с вероятностью $|\beta|^2$.

Подобно случаю одиночного кубита система двух кубитов имеет четыре *состояния вычислительного базиса,* обозначаемых как $|00\rangle$, $|01\rangle$, $|10\rangle$ и $|11\rangle$. Здесь для любых $x, y \in \mathbb{F}_2$ выполнено $|xy\rangle \equiv |x\rangle|y\rangle$, где $|x\rangle$ — состояние первого кубита, а $|y\rangle$ — состояние второго кубита. Тогда вектор состояния, описывающий два кубита, имеет вид

$$|\varphi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle,$$

где $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Систему с произвольным числом кубитов описывает

Постулат 1. С каждой изолированной физической системой связывается комплексное векторное пространство со скалярным произведением, которое называется *пространством состояний* системы. Система полностью описывается *вектором состояния*, который представляет собой единичный вектор в пространстве состояний системы.

Для квантовых схем будем применять обозначения, представленные на рис. 1. В этих обозначениях временна́я ось направлена слева направо.

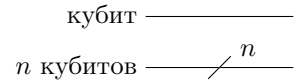


Рис. 1. Обозначения кубита и n кубитов

Постоянными будем называть кубиты, используемые на протяжении всей работы оракула, а *вре́менными* — те, которые используются только во время проведения промежуточных операций.

2.2. Эволюция квантовомеханической системы. Изменения состояния $|\psi\rangle$ квантовомеханической системы во времени описывает

Постулат 2. Эволюция замкнутой квантовой системы описывается унитарным преобразованием. Другими словами, состояние $|\psi\rangle$ системы в момент времени t_1 связано с её состоянием $|\psi'\rangle$ в момент времени t_2 посредством унитарного оператора U, зависящего только от моментов времени t_1 и t_2:

$$|\psi'\rangle = U|\psi\rangle. \tag{2}$$

Пусть $|\psi\rangle = |x_1, x_2, \ldots, x_k\rangle$ и $|\psi'\rangle = |y_1, y_2, \ldots, y_k\rangle$. Тогда равенство (2) можно переписать в обозначениях квантовых схем, как это показано на рис. 2. Базовое преобразование квантового компьютера будем называть *вентилем.* Для того чтобы описать работу вентиля, достаточно указать принцип работы данного вентиля на вычислительном базисе кубитов, на которых он действует. Для произвольного числа кубитов n вычислительный базис имеет вид $\{|x\rangle \mid x \in \mathbb{F}_2^n\}$. В настоящей работе для

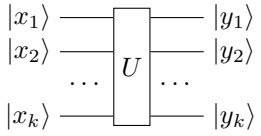
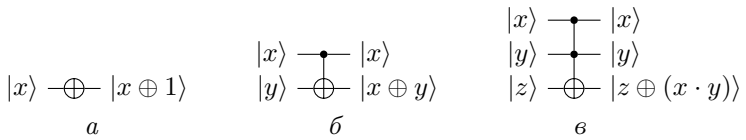


Рис. 2. Квантовая схема равенства (2)

построения всех операций и функций на квантовом компьютере используются базисные вентили, представленные на рис. 3 ($x, y, z \in \mathbb{F}_2$).



а *б* *в*

Рис. 3. Используемые вентили:
а) Паули-X (NOT), б) CNOT, в) Тоффоли (CCNOT)

Обозначение процесса измерения в вычислительном базисе кубита в квантовых схемах изображено на рис. 4. Здесь $\psi = 0$ с вероятностью $|\alpha|^2$ и $\psi = 1$ с вероятностью $|\beta|^2$.

Рис. 4. Измерение кубита

Пусть $x \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2$ и U_f — квантовый аналог булевой функции f от n переменных. Тогда действие U_f на кубитах $|x\rangle$ и $|y\rangle$

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

моделируется схемой на рис. 5.

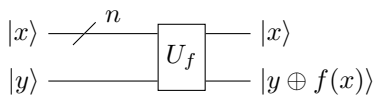


Рис. 5. Моделирование булевой функции на квантовом компьютере

Важным оператором, используемым во многих квантовых алгоритмах, является оператор Адамара H, изображённый на рис. 6. В матричном представлении этот оператор имеет вид

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

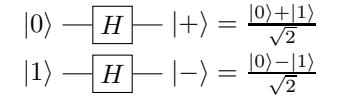


Рис. 6. Вентиль Адамара

Определим *глубину квантовой схемы* как число слоёв, которые содержит схема. Один слой состоит из базисных вентилей, применённых к непересекающимся множествам кубитов.

Так как информация, записанная во временных кубитах, не нужна после получения результата промежуточной операции, данные кубиты нужно очистить для возможности их дальнейшего использования. *Очистка кубитов* заключается в применении в обратном порядке ранее используемых вентилей. Значит, для очистки всех временных кубитов необходимо после получения результата операции применить в обратном порядке все ранее применённые вентили, которые не участвуют в изменении выходных кубитов операции.

2.3. Алгоритм Гровера. *Квантовый параллелизм* — это фундаментальное свойство многих квантовых компьютеров, позволяющее вычислять функцию $f(x)$ для многих различных значений x одновременно. Для понимания работы квантового параллелизма рассмотрим следующие рассуждения. Обозначим через $H^{\otimes n}$ преобразование Адамара, действующее на n кубитов. Результатом применения $H^{\otimes n}$ к кубитами, изначально находящимися в состоянии $|0\dots0\rangle$, будет состояние

$$H^{\otimes n}|0\dots0\rangle = \frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x\rangle.$$

Иными словами, преобразование Адамара приводит к суперпозиции всех состояний вычислительного базиса с одинаковыми коэффициентами. Тогда параллельное вычисление булевой функции $f(x)$ от n переменных может быть выполнено следующим образом. Приготавливаем $n+1$ кубитов в состоянии $|0\dots0\rangle$, затем применяем к первым n кубитам преобразование Адамара, после чего задействуем квантовую схему, реализующую U_f. Это даёт состояние

$$U_f\Big[\Big(\frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x\rangle\Big)|0\rangle\Big] = \frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x, f(x)\rangle. \tag{3}$$

Однако этим параллелизмом нельзя воспользоваться непосредственно. Измерение состояния (3) даёт значение $f(x)$ только для одного x. Для получения пользы от квантового параллелизма нужно иметь возможность «извлекать» информацию о более чем одном значении $f(x)$ из суперпозиции состояний (3).

Квантовым алгоритмом, решающим задачу поиска, является *алгоритм Гровера* [41] (рис. 7), в котором $1 \leqslant M \leqslant 2^{n-1}$ — число решений уравнения $f(x) = 1$, т. е.

$$M = \left|\left\{x \in \mathbb{F}_2^n \mid f(x) = 1\right\}\right|,$$

и G обозначает итерацию Гровера (рис. 8). Преобразование «Фаза» является известным вентилем, в отличие от вентиля «Оракул», который строится под каждую задачу отдельно.

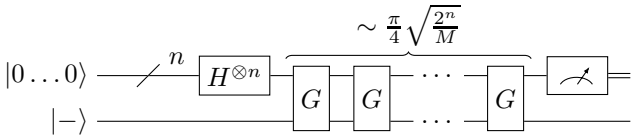


Рис. 7. Алгоритм Гровера

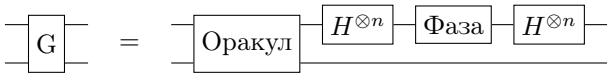


Рис. 8. Итерация Гровера

Принцип работы алгоритма Гровера можно найти в [40–43]. В [44] представлен подход, использующий алгоритм Гровера и позволяющий решать задачу поиска при неизвестном числе элементов, удовлетворяющих условию поиска.

3. Сложность реализации некоторых операций на квантовом компьютере

В [40] были получены реализации некоторых операций на квантовом компьютере, представленные в табл. 1, а также выражения для числа кубитов и глубины схемы, достаточных для реализации суммы нескольких целых положительных чисел.

Утверждение 1 [40]**.** *Пусть есть d целых положительных чисел, длина двоичного кода каждого из которых равна m. Тогда число кубитов, достаточное для их сложения, равно*

$$\sum_{i \in A} \left[\sum_{j=1}^{i-1} 2^{i-j-1}(m+j) + (m+i) \right] - (m + \min A),$$

где

$$A = \left\{ i \in \{1, 2, \ldots, \lceil \log_2(d+1) \rceil\} \mid \left\lfloor \frac{d}{2^{i-1}} \right\rfloor \bmod 2 \neq 0 \right\}.$$

Таблица 1

Число кубитов и глубина схемы, достаточные для реализации некоторых операций на квантовом компьютере [40]

Операция	Кубиты		Глубина
	постоянные	временные	
Сложение двух целых m-битных чисел, представленных в дополнительном коде	$m+1$	—	$2m+1$
Возведение в квадрат целого m-битного числа, представленного в прямом коде	$2m-2$	m^2-2m	$8m^2-24m+12$
Смена знака целого m-битного числа, представленного в дополнительном коде	m	—	$m+2$
Перевод целого m-битного числа из дополнительного кода в прямой	m	$\lceil\frac{m-1}{2}\rceil-1$	$2\lceil\log_2(m-1)+1\rceil+m$
Сравнение двух положительных целых m-битных чисел	1	$m+1$	$7m$

При этом глубина схемы равна

$$2\lceil\log_2 d\rceil\left(m+\frac{\lceil\log_2 d\rceil+1}{2}\right).$$

В табл. 1 и далее число постоянных и временных кубитов не учитывает входные кубиты схемы. Кроме этих операций для построения предложенной в разд. 4 модели квантового оракула понадобится также реализация векторной булевой функции.

3.1. Реализация векторной булевой функции, минимизирующая число кубитов. Рассмотрим реализацию произвольной векторной булевой функции $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ на квантовом компьютере. Обозначим через $|x_1\rangle,\ldots,|x_n\rangle$ кубиты, поступившие на вход функции F, а через $|y_1\rangle,\ldots,|y_m\rangle$ — кубиты на выходе функции F, инициализированные нулями.

Пусть булева функция $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ равна сумме всевозможных мономов от переменных $x_1,\ldots,x_n$. Описанный ниже метод реализации функции f даст оценку сверху на общее число вентилей и глубину схемы для реализации любой булевой функции от n переменных. Для временных

вычислений выделим дополнительно $n-2$ кубитов с начальной инициализацией $|0^{\otimes(n-2)}\rangle$ и обозначим их $|z_1\rangle, \ldots, |z_{n-2}\rangle$. Используем их для вычисления мономов. Первым шагом с помощью вентилей CCNOT последовательно присвоим в $|z_1\rangle$ состояние $|x_1 \cdot x_2\rangle$, $|z_2\rangle$ — состояние $|z_1 \cdot x_3\rangle$, $\ldots$, $|z_{n-2}\rangle$ — состояние $|z_{n-3} \cdot x_{n-1}\rangle$. После этого запишем с помощью вентиля CCNOT в $|y_1\rangle$ состояние $|z_{n-2} \cdot x_n\rangle = |x_1 \cdot \cdots \cdot x_n\rangle$, а с помощью вентиля CNOT — состояние, содержащееся в $|z_{n-2}\rangle$. Далее, используя вентили CCNOT, очистим кубит $|z_{n-2}\rangle$ и запишем в него $|z_{n-3} \cdot x_n\rangle$. Добавим в $|y_1\rangle$ состояние $|z_{n-2}\rangle$ и очистим $|z_{n-2}\rangle$. Таким образом можно добавить в $|y_1\rangle$ все возможные состояния, соответствующие мономам от переменных $x_1, \ldots, x_n$. Стоит отметить, что прибавление в $|y_1\rangle$ состояния, соответствующего моному первой степени, происходит за один шаг с помощью вентиля CNOT. Состояние $|1\rangle$ прибавляется в $|y_1\rangle$ с помощью действия вентиля NOT на $|y_1\rangle$. Тогда каждое состояние, соответствующее моному степени не равной 0, 1 и n, прибавляется в $|y_1\rangle$ за 2 вентиля CCNOT и один вентиль CNOT. Тогда общее число вентилей данной схемы равно $3 \cdot (2^n - n - 2) + n + 2$. Глубину схемы считаем равной общему числу вентилей.

Также процесс построения булевой функции, содержащей все мономы, можно представить как обход всех вершин некоторого полного двоичного дерева, начинающийся и заканчивающийся в корне. Вершины данного дерева являются мономами. Корень соответствует моному 1. Дети корневой вершины строятся следующим образом: если это вершина, соответствующая спуску влево, то её моном определяется как произведение монома, соответствующего родительской вершине, и переменной x_1; если это вершина, соответствующая спуску вправо, то её моном определяется как моном родительской вершины. Дети новых вершин определяются аналогичным образом с использованием переменной x_2. Данный процесс повторяется для всех переменных $x_1, \ldots, x_n$ до построения полного дерева. Так, самому левому листу данного дерева соответствует моном $x_1 \ldots x_n$, самому правому — моном 1. Представим дополнительно выделенные $n-2$ кубитов как стек. Тогда во время обхода каждый спуск влево, кроме спуска к вершинам с мономами степени 1 и n, соответствует добавлению в стек нового монома с помощью одного вентиля CCNOT. Подъём вправо соответствует удалению из стека монома с помощью одного вентиля CCNOT. Также каждый моном, полученный после спуска влево, необходимо добавить в $|y_1\rangle$, что соответствует использованию вентиля CNOT. Моном 1 добавляется в $|y_1\rangle$ с помощью вентиля NOT.

Рассмотрим случай, когда $m > 1$ и в каждой координатной булевой функции используются все возможные мономы. Этот случай будет являться оценкой сверху на общее число вентилей и глубину схемы. Тогда добавим в случай $m = 1$ некоторые изменения.

• После добавления в $|y_1\rangle$ состояния $|x_1 \cdot \dots \cdot x_n\rangle$, используя вентили CNOT, размножим данное состояние на все остальные $|y_i\rangle$, $i = 2, \dots, m$. Глубина такой операции составляет $\lceil \log_2 m \rceil$.

• Для состояний, соответствующих мономам степени, отличной от 0 и n, используем $\lceil \frac{m}{4} \rceil - 1$ дополнительных кубитов для клонирования состояния и параллельного добавления в $|y_1\rangle, \dots, |y_m\rangle$. После очищаем дополнительные кубиты. Глубина такой операции равна $2\lceil \log_2 m \rceil$.

Тогда глубина квантовой схемы функции $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ равна

$$(2\lceil \log_2 m \rceil + 2)(2^n - n - 2) + 2\lceil \log_2 m \rceil n + \lceil \log_2 m \rceil + 1 = \\ = 2^{n+1} - 2n + \lceil \log_2 m \rceil (2^{n+1} - 3) - 3.$$

3.2. Реализация векторной булевой функции, минимизирующая глубину схемы. Рассмотрим вторую реализацию векторной булевой функции. Аналогично первой реализации сначала покажем, как построить булеву функцию $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$. Заметим, что любую булеву функцию можно представить как $f = g_1 \oplus g_2 x_n$, где g_1, g_2 — булевы функции от переменных $x_1, \dots, x_{n-1}$. Аналогичное представление имеют функции $g_1 = h_{11} \oplus h_{12} x_{n-1}$ и $g_2 = h_{21} \oplus h_{22} x_{n-1}$, где $h_{11}, h_{12}, h_{21}, h_{22}$ — булевы функции от переменных $x_1, \dots, x_{n-2}$. Пусть построены функции $h_{11}, h_{12}, h_{21}, h_{22}$. Покажем, как с помощью вентилей CCNOT и входных кубитов $|x_{n-1}\rangle$ и $|x_n\rangle$ построить функцию f.

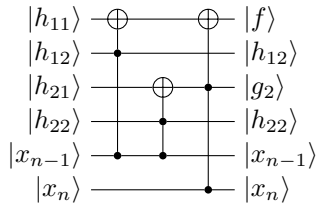


Рис. 9. Пример построения булевой функции

Тем самым каждую новую функцию предлагается разбивать на две функции от меньшего числа переменных до получения функций от переменной x_1. Таким образом, необходимо 2^{n-1} кубитов для всех булевых функций от переменной x_1. Так как булевы функции от переменных $x_1, \dots, x_k$ хранятся в кубитах, ранее используемых для хранения булевых функций от переменных $x_1, \dots, x_{k-1}$, для хранения всех функций, используемых в построении функции f, нужны 1 постоянный кубит для выхода функции f и $2^{n-1} - 1$ временных кубитов для промежуточных функций. Покажем, как построить функции от переменной x_1

(случай для получения верхней оценки). Предполагая, что каждая булева функция содержит моном x_1, добавим входное состояние, соответствующее данному моному, во все кубиты, выделенные для функций и инициализированные в начале значением $|0\rangle$. Эта операция описана в [45], и её глубина равна $\log_2(2^{n-1}) + 1$. Далее применим вентили NOT к тем кубитам, которые соответствуют функциям, содержащим константу 1. Далее построим функции от переменных x_1, x_2. Число таких функций равно 2^{n-1}. Для уменьшения глубины схемы предлагается размножить состояние $|x_2\rangle$ на 2^{n-1} временных кубитов так, как это описано в работе [45]. Глубина схемы, выполняющей данную операцию, равна $\log_2(2^{n-2})$. Затем, применяя вентили CCNOT к кубитам с состоянием $|x_2\rangle$ и кубитам, соответствующим функциям от переменной x_1, получим состояния, соответствующим функциям от переменных x_1, x_2. Повторим данный процесс до получения состояния, соответствующего функции f. Итоговая схема использует 1 постоянный кубит и

$$(2^{n-1} - 1) + (2^{n-2} - 1) + \cdots + (2^2 - 1) = \\ = \sum_{i=3}^{n} 2^{i-1} - (n-2) = \sum_{i=1}^{n} 2^{i-1} - (n+1) = 2^n - (n+2)$$

временных кубитов. Итоговая глубина схемы с очисткой временных кубитов равна

$$2(\log_2 2^{n-1} + 2 + \log_2 2^{n-2} + 1 + \cdots + \log_2 2^2 + 1 + 2) + 1 = n^2 + n + 1.$$

Теперь опишем построение векторной булевой функции F из $\mathbb{F}_2^n$ в $\mathbb{F}_2^m$. Сначала выделим m постоянных кубитов под выход операции. Так как векторную булеву функцию F можно представить как m булевых функций $f_1, \ldots, f_m \colon \mathbb{F}_2^n \to \mathbb{F}_2$, построение функции F состоит из размножения входного состояния до m состояний, параллельного вычисления состояний, соответствующих булевым функциям $f_1, \ldots, f_m$, и итоговой очистки

Таблица 2

Число кубитов и глубина схемы, достаточные для реализации функции $F \colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ на квантовом компьютере

Реализация	Минимизация кубитов	Минимизация глубины
Постоянные кубиты	m	m
Временные кубиты	$n + \left\lceil \frac{m}{4} \right\rceil - 3$	$(m-1)n + m(2^n - (n+2))$
Глубина	$2^{n+1} - 2n + \lceil \log_2 m \rceil (2^{n+1} - 3) - 3$	$2\log_2 m + n^2 + n + 1$

всей схемы. Тогда итоговое число временных кубитов равно

$$(m-1)n + m(2^n - (n+2)),$$

а итоговая глубина схемы равна

$$2\log_2 m + n^2 + n + 1.$$

3.3. Итоги. В табл. 2 представлены выражения для числа кубитов и глубины схемы, достаточных для предложенных реализаций векторной булевой функции $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ на квантовом компьютере.

4. Модель квантового оракула

В [40] была представлена модель квантового оракула, реализующая итерацию алгоритма Гровера для функции ПОИСК из алгоритма Gauss-Sieve. В этом разделе представлена новая модель данного оракула. Обозначим через K длину списка L, хранящего векторы размерности d, каждая координата которых кодируется битовой строкой длины m.

4.1. Описание модели. Рассмотрим модель оракула, представленную на рис. 10, у которой неупорядоченный список хранится в классической памяти.

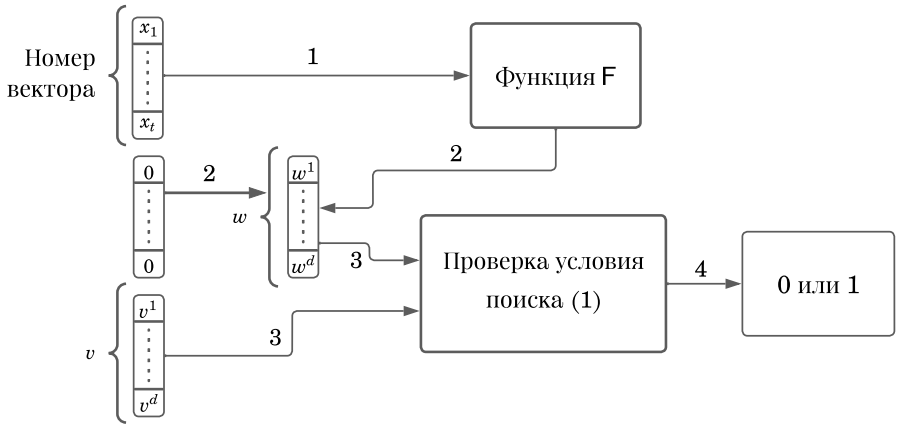


Рис. 10. Предлагаемая модель оракула с классическим списком

Работа данной модели оракула происходит следующим образом:

1) получение номера вектора на вход и передача его в функцию F;

2) получение в качестве выхода функции F вектора из списка L, соответствующего заданному номеру;

3) проверка полученного вектора на условие поиска (1);

4) вывод ответа: 1 — если вектор удовлетворяет условию, 0 — если нет.

Данная модель помогает избежать линейного роста числа кубитов, используемых оракулом, при увеличении размера списка L. Для этого построим векторную булеву функцию $\mathsf{F}\colon \mathbb{F}_2^{\lceil \log_2 K \rceil} \to \mathbb{F}_2^{dm}$, которая по номеру элемента списка возвращает соответствующий элемент. Если в списке нет соответствующего элемента, то F возвращает нулевой вектор. Поскольку в ходе алгоритма GaussSieve в списке отсутствует элемент, у которого каждая координата равняется нулю, возвращённый функцией F нулевой вектор будет указывать на отсутствие в списке соответствующего номеру элемента. Построение подобной векторной булевой функции может потребовать вычислений на классической части алгоритма, которые будут линейно зависеть от длины списка L. Однако не будем строить функцию F заново при каждом новом поиске, а будем использовать её перестроение, которое подробнее опишем ниже. Остальная часть оракула остаётся неизменной, поскольку построение булевой функции, соответствующей проверке на условие поиска, эквивалентно проверке каждого элемента списка на условие поиска.

Перестроение функции F. Чтобы избежать построения новой векторной булевой функции F при каждом изменении списка, будем использовать перестроение функции F. В ходе работы алгоритма GaussSieve между двумя поисками список может измениться следующим образом:

- из списка удалили один элемент;
- в список добавили один элемент.

В первом случае на место, в котором находился удалённый элемент, записывается нулевой вектор. Во втором случае возможны два варианта.

- В списке нет свободного места. Тогда необходимо расширить список, что потребует полного перестроения функции F.
- В списке есть свободное место. Тогда элемент занимает это свободное место.

Рассмотрим случаи, когда не требуется полного перестроения функции F. Пусть изменения произошли в элементе с номером k, тогда обозначим через $w_k^{\text{old}} \in \mathbb{F}_2^{dm}$ и $w_k^{\text{new}} \in \mathbb{F}_2^{dm}$ состояние этого элемента списка до и после изменения соответственно. Через $\mathsf{F}^{\text{old}}\colon \mathbb{F}_2^{\lceil \log_2 K \rceil} \to \mathbb{F}_2^{dm}$ обозначим функцию, соответствующую списку до изменения элемента с номером k, а через $\mathsf{F}^{\text{new}}\colon \mathbb{F}_2^{\lceil \log_2 K \rceil} \to \mathbb{F}_2^{dm}$ — функцию, соответствующую списку после изменения элемента с номером k. *Индикатором* неотрицательного целого числа $t \leqslant 2^{\lceil \log_2 K \rceil} - 1$ назовём булеву функцию $I_t\colon \mathbb{F}_2^{\lceil \log_2 K \rceil} \to \mathbb{F}_2$, определённую формулой

$$I_t(x) = \begin{cases} 1, & \text{если } x \text{ — двоичное представление } t, \\ 0 & \text{иначе.} \end{cases}$$

Пусть $\Delta_k = w_k^{\text{old}} \oplus w_k^{\text{new}}$. Тогда $\mathsf{F}^{\text{new}}(x)$ можно представить следующим образом:

$$\mathsf{F}^{\text{new}}(x) = \mathsf{F}^{\text{old}}(x) \oplus I_k(x) \cdot \Delta_k.$$

Такое преобразование может быть выполнено быстрее, чем построение новой векторной булевой функции.

4.2. Оценки сложности реализации, минимизирующей число кубитов квантовой схемы. В этом пункте рассматривается реализация предложенной модели квантового оракула, использующая реализацию функции F с меньшим числом кубитов.

Теорема 1. *Пусть имеется список длины K, состоящий из целочисленных векторов размерности $d \geqslant 2$, каждая координата которых кодируется битовой строкой длины $m \geqslant 3$. Тогда для реализации квантового оракула, представленного на рис.* 10, *потребуется не более*

$$\lceil \log_2 K \rceil + 14dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3 + \\ + \max\left\{3d(m^2-1),\ \lceil \log_2 K \rceil + \left\lceil \frac{dm}{4} \right\rceil - 3\right\}$$

кубитов. При этом глубина не превосходит

$$2^{\lceil \log_2 K \rceil + 2} - 4\lceil \log_2 K \rceil + \lceil \log_2 dm \rceil (2^{\lceil \log_2 K \rceil + 2} - 6) + \\ + 16m^2 + 8m + 4\lceil \log_2 m \rceil + \lceil \log_2 d \rceil (2\lceil \log_2 d \rceil + 8m + 10) + 13.$$

Доказательство. Разделим доказательство на две части: оценка числа кубитов и оценка глубины схемы.

Оценка числа кубитов. Сначала оценим число постоянных кубитов. В отличие от представленной в работе [40] модели оракула, данная модель не использует переключатель и не хранит список L в квантовой памяти. Функция F использует dm постоянных кубитов, что соответствует dm кубитам, выделенным под копирование вектора в реализации [40, теорема 1], минимизирующей число кубитов. Поэтому, убирая из выражения для числа постоянных кубитов

$$\lceil \log_2 K \rceil + 2^{\lceil \log_2 K \rceil} + Kdm + 14dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3$$

слагаемые Kdm и $2^{\lceil \log_2 K \rceil}$, соответствующие постоянным кубитам переключателя и кубитам, выделенным для хранения списка L, получаем число постоянных кубитов для предложенной реализации модели:

$$\lceil \log_2 K \rceil + 14dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3.$$

Оценим число временных кубитов. Для реализации [40, теорема 1] это число равнялось временным кубитам $3d$ операций вычисления квадрата числа, представленного в прямом коде длины $m+1$. В данной же модели

вместо переключателя и копирования векторов используется функция F, а значит, число временных кубитов равно максимуму временных кубитов функции F и $3d$ операций вычисления квадрата числа, представленного в прямом коде длины $m+1$. Следовательно, общее число кубитов данной реализации модели квантового оракула равно

$$\lceil \log_2 K \rceil + 14dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3 + \\ + \max\left\{3d(m^2-1),\ \lceil \log_2 K \rceil + \left\lceil \frac{dm}{4} \right\rceil - 3\right\}.$$

Оценка глубины схемы. В схеме из [40, теорема 1] глубина переключателя и копирования векторов из списка равнялась $(3^{\lceil \log_2 K \rceil} - 1)$ и $K(2\lceil \log_2 dm \rceil + 1)$ соответственно. Тогда, заменив в

$$2 \cdot 3^{\lceil \log_2 K \rceil} + 2K(2\lceil \log_2 dm \rceil + 1) + 16m^2 + 8m + \\ + 4\lceil \log_2 m \rceil + \lceil \log_2 d \rceil (2\lceil \log_2 d \rceil + 8m + 10) + 17$$

эти слагаемые на $2^{\lceil \log_2 K \rceil + 1} - 2\lceil \log_2 K \rceil + \lceil \log_2 dm \rceil (2^{\lceil \log_2 K \rceil + 1} - 3) - 3$ (глубину функции F), получим итоговую глубину схемы квантового оракула, в котором список хранится в классической памяти, с учётом очистки кубитов равную

$$2^{\lceil \log_2 K \rceil + 2} - 4\lceil \log_2 K \rceil + \lceil \log_2 dm \rceil (2^{\lceil \log_2 K \rceil + 2} - 6) + \\ + 16m^2 + 8m + 4\lceil \log_2 m \rceil + \lceil \log_2 d \rceil (2\lceil \log_2 d \rceil + 8m + 10) + 13.$$

Теорема 1 доказана.

Обозначим через $\mathcal{Q}$ схему, реализующую модель квантового оракула, представленного на рис. 10, и имеющую минимальное число кубитов. Тогда из теоремы 1 получим

Следствие 1. *В условиях теоремы* 1 *число кубитов, используемых в схеме $\mathcal{Q}$, не превосходит*

$$\lceil \log_2 K \rceil + 14dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3 + \\ + \max\left\{3d(m^2-1),\ \lceil \log_2 K \rceil + \left\lceil \frac{dm}{4} \right\rceil - 3\right\}.$$

4.3. Оценки сложности реализации, минимизирующей глубину квантовой схемы. Здесь рассматривается реализация предложенной модели квантового оракула, использующая реализацию функции F с меньшей глубиной схемы.

Теорема 2. *Пусть имеется список длины K, состоящий из целочисленных векторов размерности $d \geqslant 2$, каждая координата которых кодируется битовой строкой длины $m \geqslant 3$. Тогда для реализации квантового*

оракула, представленного на рис. 10, *потребуется квантовая схема, глубина которой равна*

$$2\lceil\log_2 K\rceil^2 + 2\lceil\log_2 K\rceil + 4\lceil\log_2 dm\rceil + 16m^2 + 8m + \\ + 4\lceil\log_2 m\rceil + \lceil\log_2 d\rceil(2\lceil\log_2 d\rceil + 8m + 10) + 21.$$

При этом число кубитов данной схемы равно

$$\max\{3d(m^2-1),\ (dm-1)\lceil\log_2 K\rceil + dm(2^{\lceil\log_2 K\rceil} - (\lceil\log_2 K\rceil + 2))\} + \\ + \lceil\log_2 K\rceil + 14dm + 5d + 6m + 3\lceil\log_2 d\rceil + 3.$$

Доказательство. Аналогично теореме 1 разделим доказательство на две части: оценка числа кубитов и оценка глубины схемы.

Оценка числа кубитов. Так как данная реализация модели отличается от реализации модели из теоремы 1 только использованием реализации функции F, минимизирующей глубину схемы, число постоянных кубитов совпадает с числом постоянных кубитов из теоремы 1 и равно

$$\lceil\log_2 K\rceil + 14dm + 5d + 6m + 3\lceil\log_2 d\rceil + 3.$$

Теперь оценим число временных кубитов. Для первой реализации это число равнялось

$$\max\left\{3d(m^2-1),\ \lceil\log_2 K\rceil + \left\lceil\frac{dm}{4}\right\rceil - 3\right\}.$$

Тогда, заменив в этой формуле число кубитов для первого способа построения функции F, равное

$$\lceil\log_2 K\rceil + \left\lceil\frac{dm}{4}\right\rceil - 3,$$

числом кубитов для второго способа построения функции F, равным

$$(dm-1)\lceil\log_2 K\rceil + dm(2^{\lceil\log_2 K\rceil} - (\lceil\log_2 K\rceil + 2)),$$

получим число временных кубитов данной реализации

$$\max\{3d(m^2-1),\ (dm-1)\lceil\log_2 K\rceil + dm(2^{\lceil\log_2 K\rceil} - (\lceil\log_2 K\rceil + 2))\}.$$

Следовательно, общее число используемых кубитов равно

$$\max\{3d(m^2-1),\ (dm-1)\lceil\log_2 K\rceil + dm(2^{\lceil\log_2 K\rceil} - (\lceil\log_2 K\rceil + 2))\} + \\ + \lceil\log_2 K\rceil + 14dm + 5d + 6m + 3\lceil\log_2 d\rceil + 3.$$

Оценка глубины схемы. В схеме из теоремы 1 глубина реализации функции F вместе с очисткой равнялась

$$2^{\lceil\log_2 K\rceil+2} - 4\lceil\log_2 K\rceil + \lceil\log_2 dm\rceil(2^{\lceil\log_2 K\rceil+2} - 6) - 6.$$

Тогда, заменив в

$$2^{\lceil \log_2 K\rceil+2} - 4\lceil \log_2 K\rceil + \lceil \log_2 dm\rceil(2^{\lceil \log_2 K\rceil+2} - 6) + \\ + 16m^2 + 8m + 4\lceil \log_2 m\rceil + \lceil \log_2 d\rceil(2\lceil \log_2 d\rceil + 8m + 10) + 13.$$

эти слагаемые на

$$4\lceil \log_2(dm)\rceil + 2\lceil \log_2 K\rceil^2 + 2\lceil \log_2 K\rceil + 2$$

(глубину реализации функции F, минимизирующей глубину схемы, вместе с очисткой), получим итоговую глубину реализации предложенной модели квантового оракула с учётом очистки кубитов

$$2\lceil \log_2 K\rceil^2 + 2\lceil \log_2 K\rceil + 4\lceil \log_2 dm\rceil + 16m^2 + 8m + \\ + 4\lceil \log_2 m\rceil + \lceil \log_2 d\rceil(2\lceil \log_2 d\rceil + 8m + 10) + 21.$$

Теорема 2 доказана.

Обозначим через $\mathcal{D}$ схему, реализующую модель квантового оракула, представленного на рис. 10, и имеющую минимальную глубину. Тогда из теоремы 2 получим

Следствие 2. *В условиях теоремы* 2 *глубина схемы* $\mathcal{D}$ *не превосходит*

$$2\lceil \log_2 K\rceil^2 + 2\lceil \log_2 K\rceil + 4\lceil \log_2 dm\rceil + 16m^2 + 8m + \\ + 4\lceil \log_2 m\rceil + \lceil \log_2 d\rceil(2\lceil \log_2 d\rceil + 8m + 10) + 21.$$

4.4. Асимптотики. Важным параметром в полученных оценках является длина списка, которая на практике увеличивается экспоненциально с ростом размерности решётки d, т. е. $K \sim 2^{0{,}21d}$. Из теоремы 1 следует, что верхняя асимптотическая оценка для числа кубитов в реализации, минимизирующей число кубитов, второй модели оракула равна $O(\log_2 K + dm^2)$. Однако верхняя асимптотическая оценка для глубины данной реализации полиномиально зависит от длины списка K. Аналогично из теоремы 2 следует, что верхняя асимптотическая оценка для глубины реализации, минимизирующей глубину схемы, равна $O\big(\log_2^2 K + (m + \log_2 d)^2\big)$, но верхняя асимптотическая оценка для числа кубитов данной реализации полиномиально зависит от длины списка K.

5. Сравнение моделей

В этом разделе приведено сравнение новой модели квантового оракула с моделью из [40].

5.1. Связь числа кубитов и глубины схемы новой модели квантового оракула с параметрами постквантовых криптосистем. В табл. 3 для криптосистем NTRU [46], Saber [47], CRYSTALS-Kyber [48] указана связь уровней защищённости с числом кубитов и глубиной схемы, достаточных для новой модели квантового оракула.

Таблица 3

Число кубитов и глубина схемы, достаточные для предложенных реализаций новой модели квантового оракула

Реализация		Минимизация кубитов			Минимизация глубины		
Уровень защищённости		1	3	5	1	3	5
NTRU	кубиты	$2^{19,43}$	$2^{19,84}$	$2^{20,31}$	$2^{227,7}$	$2^{299,11}$	$2^{359,49}$
	глубина	$2^{219,91}$	2^{291}	2^{351}	$2^{16,92}$	$2^{17,64}$	$2^{18,15}$
SABER	кубиты	$2^{19,8}$	$2^{20,38}$	$2^{20,8}$	$2^{229,91}$	$2^{337,5}$	$2^{445,91}$
	глубина	$2^{221,91}$	2^{329}	2^{437}	2^{17}	2^{18}	$2^{18,75}$
CRYSTALS-Kyber	кубиты	$2^{19,62}$	$2^{20,21}$	$2^{20,62}$	$2^{229,81}$	$2^{337,4}$	$2^{445,81}$
	глубина	$2^{221,91}$	2^{329}	2^{437}	$2^{16,97}$	$2^{17,98}$	$2^{18,74}$

Сравнивая полученные численные результаты с результатами из [40], можно заметить, что хранение списка в классической памяти уменьшает число кубитов, используемых при атаках на криптосистемы. Также реализация предложенной в настоящей работе модели квантового оракула, минимизирующая число кубитов, имеет меньшую глубину схемы, чем аналогичная реализация предложенной ранее модели.

5.2. Сравнительный анализ моделей. В табл. 4 представлены асимптотические оценки сложности реализаций, минимизирующих число кубитов и минимизирующих глубины схемы, новой модели квантового оракула и модели из работы [40].

Как видно из таблицы, хранение списка в классической памяти позволяет избавиться от линейной зависимости от длины списка в асимптотических оценках числа кубитов. При этом асимптотические оценки глубины схемы либо остаются неизменными, либо некритично ухудшаются.

Заключение

Разработана и описана новая модель квантового оракула, применимая в алгоритме Гровера для реализации гибридно квантово-классического

Таблица 4

Асимптотические оценки сложности реализаций

Модель		Работа [40]	Настоящая работа
Минимизация кубитов	кубиты	$O(Kdm + dm^2)$	$O(\log_2 K + dm^2)$
	глубина	$O(K \log_2(dm) + (m + \log_2 d)^2)$	$O(K \log_2(dm) + (m + \log_2 d)^2)$
Минимизация глубины	кубиты	$O(K^2 + dm(K + m))$	$O(dm(K + m))$
	глубина	$O(\log_2 K + (m + \log_2 d)^2)$	$O\big(\log_2^2 K + (m + \log_2 d)^2\big)$

алгоритма на основе GaussSieve. Получены верхние оценки числа кубитов и глубины схемы на сложность реализации предлагаемой модели. Проведено сравнение новой модели с альтернативной моделью квантового оракула.

Финансирование работы

Исследование выполнено при поддержке Математического центра в Академгородке в рамках соглашения № 075–15–2022–282 с Министерством науки и высшего образования Российской Федерации. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Bernstein D. J.** Introduction to post-quantum cryptography // Post-quantum cryptography. Heidelberg: Springer, 2009. P. 1–14.
2. **Малыгина Е. С., Куценко А. В., Новосёлов С. А.** [и др.]. Постквантовые криптосистемы: открытые вопросы и существующие решения. Криптосистемы на решётках // Дискрет. анализ и исслед. операций. 2023. Т. 30, № 4. С. 46–90.
3. **Малыгина Е. С., Куценко А. В., Новосёлов С. А.** [и др.]. Постквантовые криптосистемы: открытые вопросы и существующие решения. Криптосистемы на изогениях и кодах, исправляющих ошибки // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 1. С. 52–84.
4. **Daemen J., Rijmen V.** The design of Rijndael. Heidelberg: Springer, 2002. 238 p. DOI: 10.1007/978-3-662-04722-4.

5. **Dworkin M. J.** SHA-3 standard: Permutation-based hash and extendable-output functions. Gaithersburg, MD: Nat. Inst. Stand. Technol., 2015. 37 p. (Fed. Inf. Process. Stand. Publ.; V. 202). DOI: 10.6028/NIST.FIPS.202.
6. **Rivest R. L., Shamir A., Adleman L.** A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. 1978. V. 21, No. 2. P. 120–126.
7. **Barker E.** Digital signature standard (DSS). Gaithersburg, MD: Nat. Inst. Stand. Technol., 2013. 131 p. (Fed. Inf. Process. Stand. Publ.; V. 186-4). DOI: 10.6028/NIST.FIPS.186-4.
8. **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.
9. **Alagic G., Apon D., Cooper D.** [et al.]. Status report on the third round of the NIST post-quantum cryptography standardization process. Nat. Inst. Stand. Technol. Interagency Internal Rep. NIST IR 8413-upd1. Gaithersburg, MD: NIST, 2022. 102 p. DOI: 10.6028/NIST.IR.8413-upd1.
10. Korean Post-Quantum Cryptography Competition. Seoul: Natl. Intell. Serv., 2023. URL: kpqc.or.kr/competition.html (accessed: 9.10.2023).
11. **Micciancio D.** Inapproximability of the shortest vector problem: Toward a deterministic reduction // Theory Comput. 2012. V. 8, No. 1. P. 487–512.
12. **Kannan R.** Improved algorithms for integer programming and related lattice problems // Proc. 15th Annu. ACM Symp. Theory of Computing (Boston, USA, Apr. 25–27, 1983). New York: ACM, 1983. P. 193–206.
13. **Fincke U., Pohst M.** Improved methods for calculating vectors of short length in a lattice, including a complexity analysis // Math. Comput. 1985. V. 44, No. 170. P. 463–471.
14. **Gama N., Nguyen P. Q., Regev O.** Lattice enumeration using extreme pruning // Advances in cryptology — EUROCRYPT 2010. Proc. 29th Annu. Int. Conf. Theory and Application of Cryptographic Techniques (French Riviera, May 30 – June 3, 2010). Heidelberg: Springer, 2010. P. 257–278. (Lect. Notes Comput. Sci.; V. 6110).
15. **Lenstra A. K., Lenstra H. W., Lovász L.** Factoring polynomials with rational coefficients // Math. Ann. 1982. V. 261, No. 4. P. 515–534. DOI: 10.1007/BF01457454.
16. **Chen Y., Nguyen P. Q.** BKZ 2.0: Better lattice security estimates // Advances in cryptology — ASIACRYPT 2011. Proc. 17th Int. Conf. Theory and Application of Cryptology and Information Security (Seoul, South Korea, Dec. 4–8, 2011). Heidelberg: Springer, 2011. P. 1–20. (Lect. Notes Comput. Sci.; V. 7073).
17. **Schnorr C. P.** A hierarchy of polynomial time lattice basis reduction algorithms // Theor. Comput. Sci. 1987. V. 53, No. 2–3. P. 201–224. DOI: 10.1016/0304-3975(87)90064-8.
18. **Schnorr C. P., Euchner M.** Lattice basis reduction: Improved practical algorithms and solving subset sum problems // Math. Program. 1994. V. 66. P. 181–199. DOI: 10.1007/BF01581144.

19. **Becker A., Ducas L., Gama G., Laarhoven T.** New directions in nearest neighbor searching with applications to lattice sieving // Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms (Arlington, VA, USA, Jan. 10–12, 2016). Philadelphia, PA: SIAM, 2016. P. 10–24.
20. **Herold G., Kirshanova E., Laarhoven T.** Speed-ups and time–memory trade-offs for tuple lattice sieving // Public-key cryptography — PKC 2018. Proc. 21st IACR Int. Conf. Practice and Theory of Public-Key Cryptography (Rio de Janeiro, Brazil, Mar. 25–29, 2018). Pt. I. Cham: Springer, 2018. P. 407–436. (Lect. Notes Comput. Sci.; V. 10769).
21. **Micciancio D., Voulgaris P.** Faster exponential time algorithms for the shortest vector problem // Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms (Austin, TX, USA, Jan. 17–19, 2010). Philadelphia, PA: SIAM, 2010. P. 1468–1480.
22. **Nguyen P. Q., Vidick T.** Sieve algorithms for the shortest vector problem are practical // J. Math. Cryptol. 2008. V. 2, No. 2. P. 181–207. DOI: 10.1515/JMC.2008.009.
23. **Pujol X., Stehlé D.** Solving the shortest lattice vector problem in time $2^{2.465n}$. San Diego: Univ. California, 2009. (Cryptol. ePrint Archive; ID 2009/605). URL: eprint.iacr.org/2009/605 (accessed: 9.10.2023).
24. **Aggarwal D., Dadush D., Regev O., Stephens-Davidowitz N.** Solving the shortest vector problem in 2^n time using discrete Gaussian sampling // Proc. 47th ACM Symp. Theory of Computing (Portland, OR, USA, June 14–17, 2015). New York: ACM, 2015. P. 733–742.
25. **Doulgerakis E., Laarhoven T., de Weger B.** Finding closest lattice vectors using approximate Voronoi cells // Post-quantum cryptography. Rev. Sel. Pap. 10th Int. Conf. (Chongqing, China, May 8–10, 2019). Cham: Springer, 2019. P. 3–22. (Lect. Notes Comput. Sci.; V. 11505).
26. **Micciancio D., Voulgaris P.** A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations // Proc. 42nd ACM Symp. Theory of Computing (Cambridge, MA, USA, June 5–8, 2010). New York: ACM, 2010. P. 351–358.
27. **Денисенко Д. В., Никитенкова М. В.** Применение квантового алгоритма Гровера в задаче поиска ключа блочного шифра SDES // Журн. эксперим. и теор. физики. 2019. Т. 155, вып. 1. С. 32–53.
28. **Денисенко Д. В, Маршалко Г. Б., Никитенкова М. В., Рудской В. И., Шишкин В. А.** Оценка сложности реализации алгоритма Гровера для перебора ключей алгоритмов блочного шифрования ГОСТ Р 34.12-2015 // Журн. эксперим. и теор. физики. 2019. Т. 155, вып. 4. С. 645–653.
29. **Almazrooie M., Samsudin A., Abdullah R., Mutter K. N.** Quantum exhaustive key search with simplified-DES as a case study // SpringerPlus. 2016. V. 5, No. 1. P. 1–19.
30. **Dong X., Dong B., Wang X.** Quantum attacks on some Feistel block ciphers // Des. Codes Cryptogr. 2020. V. 88, No. 6. P. 1179–1203. DOI: 10.1007/s10623-020-00741-y.

31. **Frixons P., Naya-Plasencia M., Schrottenloher A.** Quantum boomerang attacks and some applications // Selected areas in cryptography. Proc. 28th Int. Conf. (Virtual Event, Sept. 29 – Oct. 1, 2021). Cham: Springer, 2021. P. 332–352. (Lect. Notes Comput. Sci.; V. 13203).
32. **Jaques S., Naehrig M., Roetteler M., Virdia F.** Implementing Grover oracles for quantum key search on AES and LowMC // Advances in cryptology — EUROCRYPT 2020. Proc. 39th Annu. Int. Conf. Theory and Application of Cryptographic Techniques (Zagreb, Croatia, May 10–14, 2020). Pt. II. Cham: Springer, 2020. P. 280–310. (Lect. Notes Comput. Sci.; V. 12106).
33. **Grassl M., Langenberg B., Roetteler M., Steinwandt R.** Applying Grover's algorithm to AES: quantum resource estimates // Post-quantum cryptography. Proc. 7th Int. Workshop (Fukuoka, Japan, Feb. 24–26, 2016). Cham: Springer, 2016. P. 29–43. (Lect. Notes Comput. Sci.; V. 9606).
34. **Langenberg B., Pham H., Steinwandt R.** Reducing the cost of implementing the advanced encryption standard as a quantum circuit // IEEE Trans. Quantum Eng. 2020. V. 1. P. 1–12.
35. **Zou J., Wei Z., Sun S., Liu X., Wu W.** Quantum circuit implementations of AES with fewer qubits // 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. II. Cham: Springer, 2020. P. 697–726. (Lect. Notes Comput. Sci.; V. 12492).
36. **Albrecht M. R., Gheorghiu V., Postlethwaite E. W., Schanck J. M.** Estimating quantum speedups for lattice sieves // 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. II. Cham: Springer, 2020. P. 583–613. (Lect. Notes Comput. Sci.; V. 12492).
37. **Gidney C., Ekerå M.** How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits // Quantum. 2021. V. 5. P. 433. DOI: 10.22331/q-2021-04-15-433.
38. **Laarhoven T., Mosca M., van de Pol J.** Finding shortest lattice vectors faster using quantum search // Des. Codes Cryptogr. 2015. V. 77, No. 2–3. P. 375–400.
39. **Perriello S., Barenghi A., Pelosi G.** A complete quantum circuit to solve the information set decoding problem // Proc. 2021 IEEE Int. Conf. Quantum Computing and Engineering (Broomfield, CO, USA, Oct. 17–22, 2021). Los Alamitos, CA: IEEE Comput. Soc., 2021. P. 366–377.
40. **Бахарев А. О.** Оценки сложности реализации квантового криптоанализа постквантовых криптосистем, основанных на решётках // Дискрет. анализ и исслед. операций. 2023. Т. 30, № 3. С. 5–42.
41. **Grover L. K.** A fast quantum mechanical algorithm for database search // Proc. 28th ACM Symp. Theory of Computing (Philadelphia, PA, USA, May 22–24, 1996). New York: ACM, 1996. P. 212–219. DOI: 10.1145/237814.237866.
42. **Nielsen M. A., Chuang I. L.** Quantum computation and quantum information. Cambridge: Camb. Univ. Press, 2010. 676 p.

43. Китаев А. Ю., Шень А. Х., Вялый М. Н. Классические и квантовые вычисления. М.: МЦНМО; ЧеРо, 1999. 192 с.
44. Boyer M., Brassard G., Høyer P., Tapp A. Tight bounds on quantum searching // Fortschr. Phys. 1998. V. 46, No. 4–5. P. 493–505.
45. Moore C., Nilsson M. Parallel quantum computation and quantum codes // SIAM J. Comput. 2001. V. 31, No. 3. P. 799–815.
46. Chen C., Danba O., Hoffstein J. [et al.]. NTRU algorithm specifications and supporting documentation. Eindhoven: Eindh. Univ. Technol., 2019. URL: `https://ntru.org/f/ntru-20190330.pdf` (accessed: 9.10.2023).
47. D'Anvers J.-P., Karmakar A., Roy S. S., Vercauteren F. SABER: Mod-LWR based KEM. Leuven: KU Leuven, 2017. URL: `https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround1.pdf` (accessed: 9.10.2023).
48. Avanzi R., Bos J., Ducas L. [et al.]. CRYSTALS-Kyber algorithm specifications and supporting documentation. Amsterdam: Cent. Wiskd. Inform., 2021. URL: `https://cryptojedi.org/papers/kybernist-20171130.pdf` (accessed: 9.10.2023).

Бахарев Александр Олегович

Статья поступила
27 июня 2023 г.
После доработки —
27 ноября 2023 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 24–53

UDC 519.7 DOI: 10.33048/daio.2024.31.777

A NEW QUANTUM ORACLE MODEL FOR A HYBRID QUANTUM-CLASSICAL ATTACK ON POST-QUANTUM LATTICE-BASED CRYPTOSYSTEMS

A. O. Bakharev

Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
E-mail: a.bakharev@g.nsu.ru

Abstract. Lattice-based cryptosystems are one of the main post-quantum alternatives to asymmetric cryptography currently in use. Most attacks on these cryptosystems can be reduced to the shortest vector problem (SVP) in a lattice. Previously, the authors proposed a quantum oracle model from Grover's algorithm to implement a hybrid quantum-classical algorithm based on the GaussSieve algorithm and solving SVP. In this paper, a new model of a quantum oracle is proposed and analyzed. Two implementations of the new quantum oracle model are proposed and estimated. The complexity of implementing the new quantum oracle model to attack post-quantum lattice-based cryptosystems that are finalists of the NIST post-quantum cryptography competition is analyzed. Comparison of obtained results for new and existing models of quantum oracle is given. Tab. 4, illustr. 10, bibliogr. 48.

Keywords: quantum search, public-key cryptography, lattice-based cryptography, post-quantum cryptography, Grover's algorithm, quantum computation.

References

1. **D. J. Bernstein,** Introduction to post-quantum cryptography, in *Post-Quantum Cryptography* (Springer, Heidelberg, 2009), pp. 1–14.
2. **E. S. Malygina, A. V. Kutsenko, S. A. Novoselov,** [et al.], Post-quantum cryptosystems: Open problems and solutions. Lattice-based cryptosystems, *Diskretn. Anal. Issled. Oper.* **30** (4), 46–90 (2023) [Russian] [*J. Appl. Ind. Math.* **17** (4), 767–790 (2023)].

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 395–411 (2024), DOI: 10.1134/S1990478924030037.

3. **E. S. Malygina, A. V. Kutsenko, S. A. Novoselov,** [et al.], Post-quantum cryptosystems: Open problems and solutions. Isogeny-based and code-based cryptosystems, *Diskretn. Anal. Issled. Oper.* **31** (1), 52–84 (2024) [Russian] [*J. Appl. Ind. Math.* **18** (1), 103–121 (2024)].
4. **J. Daemen** and **V. Rijmen,** *The Design of Rijndael* (Springer, Heidelberg, 2002), DOI: 10.1007/978-3-662-04722-4.
5. **M. J. Dworkin,** SHA-3 standard: Permutation-based hash and extendable-output functions (Nat. Inst. Stand. Technol., Gaithersburg, MD, 2015) (Fed. Inf. Process. Stand. Publ., Vol. 202), DOI: 10.6028/NIST.FIPS.202.
6. **R. L. Rivest, A. Shamir,** and **L. Adleman,** A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM* **21** (2), 120–126 (1978).
7. **E. Barker,** Digital signature standard (DSS) (Nat. Inst. Stand. Technol., Gaithersburg, MD, 2013) (Fed. Inf. Process. Stand. Publ., Vol. 186-4), DOI: 10.6028/NIST.FIPS.186-4.
8. **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.
9. **G. Alagic, D. Apon, D. Cooper,** [et al.], Status report on the third round of the NIST post-quantum cryptography standardization process, *Nat. Inst. Stand. Technol. Interagency Internal Rep. NIST IR 8413-upd1* (NIST, Gaithersburg, MD, 2022), DOI: 10.6028/NIST.IR.8413-upd1.
10. Korean Post-Quantum Cryptography Competition (Natl. Intell. Serv., Seoul, 2023), URL: kpqc.or.kr/competition.html (accessed: 9.10.2023).
11. **D. Micciancio,** Inapproximability of the shortest vector problem: Toward a deterministic reduction, *Theory Comput.* **8** (1), 487–512 (2012).
12. **R. Kannan,** Improved algorithms for integer programming and related lattice problems, in *Proc. 15th Annu. ACM Symp. Theory of Computing, Boston, USA, Apr. 25–27, 1983* (ACM, New York, 1983), pp. 193–206.
13. **U. Fincke** and **M. Pohst,** Improved methods for calculating vectors of short length in a lattice, including a complexity analysis, *Math. Comput.* **44** (170), 463–471 (1985).
14. **N. Gama, P. Q. Nguyen,** and **O. Regev,** Lattice enumeration using extreme pruning, in *Advances in Cryptology — EUROCRYPT 2010* (Proc. 29th Annu. Int. Conf. Theory and Application of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010) (Springer, Heidelberg, 2010), pp. 257–278 (Lect. Notes Comput. Sci., Vol. 6110).
15. **A. K. Lenstra, H. W. Lenstra,** and **L. Lovász,** Factoring polynomials with rational coefficients, *Math. Ann.* **261** (4), 515–534 (1982), DOI: 10.1007/BF01457454.
16. **Y. Chen** and **P. Q. Nguyen,** BKZ 2.0: Better lattice security estimates, in *Advances in Cryptology — ASIACRYPT 2011* (Proc. 17th Int. Conf. Theory and Application of Cryptology and Information Security, Seoul, South Korea, Dec. 4–8, 2011) (Springer, Heidelberg, 2011), pp. 1–20 (Lect. Notes Comput. Sci., Vol. 7073).

17. **C. P. Schnorr,** A hierarchy of polynomial time lattice basis reduction algorithms, *Theor. Comput. Sci.* **53** (2–3), 201–224 (1987), DOI: 10.1016/0304-3975(87)90064-8.
18. **C. P. Schnorr** and **M. Euchner,** Lattice basis reduction: Improved practical algorithms and solving subset sum problems, *Math. Program.* **66**, 181–199 (1994), DOI: 10.1007/BF01581144.
19. **A. Becker, L. Ducas, G. Gama,** and **T. Laarhoven,** New directions in nearest neighbor searching with applications to lattice sieving, in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms, Arlington, VA, USA, Jan. 10–12, 2016* (SIAM, Philadelphia, PA, 2016), pp. 10–24.
20. **G. Herold, E. Kirshanova,** and **T. Laarhoven,** Speed-ups and time-memory trade-offs for tuple lattice sieving, in *Public-Key Cryptography — PKC 2018* (Proc. 21st IACR Int. Conf. Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, Mar. 25–29, 2018), Pt. I (Springer, Cham, 2018), pp. 407–436 (Lect. Notes Comput. Sci., Vol. 10769).
21. **D. Micciancio** and **P. Voulgaris,** Faster exponential time algorithms for the shortest vector problem, in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms, Austin, TX, USA, Jan. 17–19, 2010* (SIAM, Philadelphia, PA, 2010), pp. 1468–1480.
22. **P. Q. Nguyen** and **T. Vidick,** Sieve algorithms for the shortest vector problem are practical, *J. Math. Cryptol.* **2** (2), 181–207 (2008), DOI: 10.1515/JMC.2008.009.
23. **X. Pujol** and **D. Stehlé,** Solving the shortest lattice vector problem in time $2^{2.465n}$ (Univ. California, San Diego, 2009) (Cryptology ePrint Archive, Pap. 2009/605). URL: eprint.iacr.org/2009/605 (accessed: 9.10.2023).
24. **D. Aggarwal, D. Dadush, O. Regev,** and **N. Stephens-Davidowitz,** Solving the shortest vector problem in 2^n time using discrete Gaussian sampling, in *Proc. 47th ACM Symp. Theory of Computing, Portland, OR, USA, June 14–17, 2015* (ACM, New York, 2015), pp. 733–742.
25. **E. Doulgerakis, T. Laarhoven,** and **B. de Weger,** Finding closest lattice vectors using approximate Voronoi cells, in *Post-Quantum Cryptography* (Rev. Sel. Pap. 10th Int. Conf. Chongqing, China, May 8–10, 2019) (Springer, Cham, 2019), pp. 3–22 (Lect. Notes Comput. Sci., Vol. 11505).
26. **D. Micciancio** and **P. Voulgaris** A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations, in *Proc. 42nd ACM Symp. Theory of Computing, Cambridge, MA, USA, June 5–8, 2010* (ACM, New York, 2010), pp. 351–358.
27. **D. V. Denisenko** and **M. V. Nikitenkova,** Application of Grover's quantum algorithm for SDES key searching, *Zh. Eksp. Teor. Fiz.* **155** (1), 32–53 (2019) [Russian] [*J. Exp. Teor. Phys.* **128** (1), 25–44 (2019)].
28. **D. V. Denisenko, G. B. Marshalko, M. V. Nikitenkova, V. I. Rudskoi,** and **V. A. Shishkin,** Estimating the complexity of Grover's algorithm for key search of block ciphers defined by GOST R 34.12-2015, *Zh. Eksp. Teor. Fiz.* **155** (4), 645–653 (2019) [Russian] [*J. Exp. Teor. Phys.* **128** (4), 552–559 (2019)].

29. **M. Almazrooie, A. Samsudin, R. Abdullah,** and **K. N. Mutter,** Quantum exhaustive key search with simplified-DES as a case study, *SpringerPlus* **5** (1), 1–19 (2016).
30. **X. Dong, B. Dong,** and **X. Wang,** Quantum attacks on some Feistel block ciphers, *Des. Codes Cryptogr.* **88** (6), 1179–1203 (2020), DOI: 10.1007/s10623-020-00741-y.
31. **P. Frixons, M. Naya-Plasencia,** and **A. Schrottenloher,** Quantum boomerang attacks and some applications, in *Selected Areas in Cryptography* (Proc. 28th Int. Conf., Virtual Event, Sept. 29 – Oct. 1, 2021) (Springer, Cham, 2021), pp. 332–352 (Lect. Notes Comput. Sci., Vol. 13203).
32. **S. Jaques, M. Naehrig, M. Roetteler,** and **F. Virdia,** Implementing Grover oracles for quantum key search on AES and LowMC, in *Advances in Cryptology — EUROCRYPT 2020* (Proc. 39th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020), Pt. II (Springer, Cham, 2020), pp. 280–310 (Lect. Notes Comput. Sci., Vol. 12106).
33. **M. Grassl, B. Langenberg, M. Roetteler,** and **R. Steinwandt,** Applying Grover's algorithm to AES: quantum resource estimates, in *Post-Quantum Cryptography* (Proc. 7th Int. Workshop, Fukuoka, Japan, Feb. 24–26, 2016) (Springer, Cham, 2016), pp. 29–43 (Lect. Notes Comput. Sci., Vol. 9606).
34. **B. Langenberg, H. Pham,** and **R. Steinwandt,** Reducing the cost of implementing the advanced encryption standard as a quantum circuit, *IEEE Trans. Quantum Eng.* **1**, 1–12 (2020).
35. **J. Zou, Z. Wei, S. Sun, X. Liu,** and **W. Wu,** Quantum circuit implementations of AES with fewer qubits, 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. II (Springer, Cham, 2020), pp. 697–726 (Lect. Notes Comput. Sci., Vol. 12492).
36. **M. R. Albrecht, V. Gheorghiu, E. W. Postlethwaite,** and **J. M. Schanck,** Estimating quantum speedups for lattice sieves, 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. II (Springer, Cham, 2020), pp. 583–613 (Lect. Notes Comput. Sci., Vol. 12492).
37. **C. Gidney** and **M. Ekerå,** How to factor 2048 bit RSA integers in 8 shours using 20 million noisy qubits, *Quantum* **5**, 433 (2021), DOI: 10.22331/q-2021-04-15-433.
38. **T. Laarhoven, M. Mosca,** and **J. van de Pol,** Finding shortest lattice vectors faster using quantum search, *Des. Codes Cryptogr.* **77** (2–3), 375–400 (2015).
39. **S. Perriello, A. Barenghi,** and **G. Pelosi,** A complete quantum circuit to solve the information set decoding problem, in *Proc. 2021 IEEE Int. Conf. Quantum Computing and Engineering, Broomfield, CO, USA, Oct. 17–22, 2021* (IEEE Comput. Soc., Los Alamitos, CA, 2021), pp. 366–377.

40. A. O. Bakharev, Estimates of implementation complexity for quantum cryptanalysis of post-quantum lattice-based cryptosystems, *Diskretn. Anal. Issled. Oper.* **30** (3), 5–42 (2023) [Russian] [*J. Appl. Ind. Math.* **17** (3), 459–482 (2023)].
41. L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proc. 28th ACM Symp. Theory of Computing, Philadelphia, PA, USA, May 22–24, 1996* (ACM, New York, 1996), pp. 212–219.
42. M. A. Nielsen and **I. L. Chuang,** *Quantum computation and quantum information* (Camb. Univ. Press, Cambridge, 2010).
43. A. Yu. Kitaev, A. Kh. Shen, and **M. N. Vyalyi,** Classical and quantum computations (MTsNMO; CheRo, Moscow, 1999).
44. M. Boyer, G. Brassard, P. Høyer, and **A. Tapp,** Tight bounds on quantum searching, *Fortschr. Phys.* **46** (4–5), 493–505 (1998), DOI: `10.1145/237814.237866`.
45. C. Moore and **M. Nilsson,** Parallel quantum computation and quantum codes, *SIAM J. Comput.* **31** (3), 799–815 (2001).
46. C. Chen, O. Danba, J. Hoffstein, [et al.], NTRU Algorithm Specifications and Supporting Documentation (Eindh. Univ. Technol., Eindhoven, 2019), URL: `ntru.org/f/ntru-20190330.pdf` (accessed: 9.10.2023).
47. J.-P. D'Anvers, A. Karmakar, S. S. Roy, and **F. Vercauteren,** SABER: Mod-LWR Based KEM (KU Leuven, Leuven, 2017), URL: `esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround1.pdf` (accessed: 9.10.2023).
48. R. Avanzi, J. Bos, L. Ducas, [et al.], CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation (Cent. Wiskd. Inform., Amsterdam, 2021), URL: `cryptojedi.org/papers/kybernist-20171130.pdf` (accessed: 9.10.2023).

Aleksandr O. Bakharev

Received June 27, 2023
Revised November 27, 2023
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 54–78

УДК 519.179.1+519.174.7 DOI: 10.33048/daio.2024.31.784

СОВЕРШЕННЫЕ РАСКРАСКИ ГИПЕРГРАФА ПОДМАТРИЦ

С. О. Бородин [a], *А. А. Тараненко* [b]

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: [a] s.borodin@g.nsu.ru, [b] taa@math.nsc.ru

Аннотация. *Гиперграфом подматриц* $G_{n\times m}$ назовём гиперграф, вершинами которого являются элементы матрицы размера $n \times m$, а гиперрёбрами — все возможные подматрицы порядка 2. В настоящей работе рассматриваются совершенные раскраски гиперграфов $G_{n\times m}$ и условия на их параметры инцидентности. Предложено несколько конструкций совершенных раскрасок $G_{n\times m}$ и доказано, что матрицы инцидентности 2-схем являются совершенными раскрасками гиперграфа подматриц. Кроме того, описаны совершенные 2-раскраски гиперграфов $G_{2\times m}$ и $G_{3\times m}$. Ил. 1, библиогр. 12.

Ключевые слова: гиперграф, симметричная 2-схема, совершенная раскраска.

Введение

Совершенные раскраски неоднократно возникали под разными названиями. Одно из первых упоминаний, где они назывались «partition design», можно найти в книге Дельсарта [1]. В работе Годсила [2] применяется термин «equitable partitions», а в работах В. Г. Визинга [3] — «дистрибутивная раскраска». Термин «совершенная раскраска» впервые использован С. А. Пузыниной в [4].

Исследованиям совершенных раскрасок графов посвящено множество работ. Например, в [4–6] изучались совершенные раскраски графа бесконечной прямоугольной решётки. Совершенные раскраски графов Джонсона рассматривались в [7], совершенные раскраски циркулянтных графов — в [8].

Существенно менее изучены совершенные раскраски гиперграфов — обобщений графов, в которых ребром могут соединяться не только две вершины, но и любое подмножество вершин.

В [9] замечено, что совершенная раскраска гиперграфа эквивалентна двудольной совершенной раскраске его графа инцидентности. Взаимно

однозначное соответствие между некоторыми комбинаторными схемами и специальными совершенными раскрасками гиперграфов установлено в [10].

Детальное исследование совершенных раскрасок гиперграфов выполнено в работе [11], в которой, в частности, обобщены основные результаты из теории совершенных раскрасок графов на гиперграфы. Кроме того, в ней введены многомерные матрицы параметров совершенной раскраски гиперграфов и исследованы их спектральные свойства.

Настоящая работа посвящена исследованию совершенных раскрасок одного семейства гиперграфов, которые будут называться гиперграфами подматриц. Вершинами этих гиперграфов являются элементы матрицы размера $n \times m$, а гиперрёбрами — все возможные подматрицы порядка два.

В разд. 1 приводится необходимая в дальнейшем терминология, определения совершенных раскрасок гиперграфов и полупараметров инцидентности, а также описываются некоторые их свойства.

В разд. 2 установлены соотношения между полупараметрами инцидентности совершенных 2-раскрасок, а также приводятся основные примеры совершенных раскрасок гиперграфа подматриц: блочные, раскраски линиями и конфигурационно однородные раскраски. Доказано, что блочные раскраски и раскраски линиями можно восстановить по полупараметрам инцидентности.

Далее, в разд. 3 доказано необходимое условие существования совершенных 2-раскрасок гиперграфа подматриц в терминах конфигураций пар строк и столбцов, откуда следует, что в любой совершенной 2-раскраске, отличной от раскраски линиями, строчные и столбцовые суммы совпадают. Доказано, что конфигурационно однородные совершенные 2-раскраски эквивалентны матрицам инцидентности 2-схем, и получены соотношения между их параметрами. Кроме того, приведены примеры не конфигурационно однородных совершенных раскрасок.

Разд. 4 посвящён построению совершенных 2-раскрасок гиперграфа подматриц с помощью произведения Кронекера из конфигурационно однородных раскрасок и многоцветных блочных совершенных раскрасок на основе совершенных раскрасок полных графов.

Наконец, в разд. 5 описаны совершенные 2-раскраски гиперграфов $G_{2\times m}$ и $G_{3\times m}$.

1. Определения и основные свойства

Гиперграфом G называется пара множеств (X, E), где X — конечное множество вершин, а $E \subseteq 2^X$ — множество гиперрёбер. Гиперграф G *d-однородный,* если каждое гиперребро состоит из d вершин, и *r-регулярный*, если каждая вершина содержится в r гиперрёбрах.

Гиперграф $G(X, E)$ можно представить в виде двудольного *графа инцидентности* (графа Леви), одна доля которого — это множество вершин X гиперграфа, а вторая доля — множество его гиперрёбер E, причём вершины x и e графа инцидентности соединены ребром тогда и только тогда, когда вершина x принадлежит гиперребру e в гиперграфе G.

Матрицей инцидентности B гиперграфа $G(X, E)$ называется $(0, 1)$-матрица размера $|X| \times |E|$, в которой элемент b_{xe} равен 1 тогда и только тогда, когда $x \in e$. Заметим, что матрица инцидентности гиперграфа является матрицей смежности долей его графа инцидентности.

Раскраской вершин гиперграфа $G(X, E)$ в k цветов (k-*раскраской*) называется сюръективная функция $f\colon X \to \{0, \ldots, k-1\}$. Каждой раскраске можно сопоставить *матрицу раскраски* P размера $|X| \times k$, элемент которой p_{xi} равен 1 тогда и только тогда, когда $f(x) = i$. Одноцветные раскраски (в которых все вершины покрашены в один цвет) тривиальны и в дальнейшем не рассматриваются.

Для выбранной раскраски f гиперграфа G и гиперребра e *цветовым составом гиперребра* назовём мультимножество цветов инцидентных ему вершин: $f(e) = \{f(x) \mid x \in e\}$. Каждой раскраске f гиперграфа G можно сопоставить *индуцированную раскраску* g его графа инцидентности, покрасив вершины x доли X в цвета $f(x)$, а вершины e доли E — в цвета $f(e)$.

Раскраску f гиперграфа G назовём *совершенной*, если цветовой состав гиперрёбер, инцидентных вершине цвета i, зависит только от цвета вершины, а не от выбора конкретной вершины этого цвета. В случае, когда гиперрёбра имеют мощность 2, имеем стандартное определение совершенной раскраски обыкновенного графа.

В [11] доказано, что раскраска f гиперграфа G совершенна тогда и только тогда, когда совершенна индуцированная раскраска g его графа инцидентности, а именно, верно равенство

$$\begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & P \\ R & 0 \end{bmatrix} = \begin{bmatrix} 0 & P \\ R & 0 \end{bmatrix} \begin{bmatrix} 0 & W \\ V & 0 \end{bmatrix}.$$

Здесь B — матрица инцидентности гиперграфа G, P и R — матрицы раскрасок вершин и гиперрёбер соответственно, а пара матриц (V, W) названа *параметрами инцидентности*. Элемент v_{ij} матрицы V равен числу гиперрёбер цвета j, инцидентных вершине цвета i, а w_{ji} равен числу вершин цвета i, содержащихся в гиперребре цвета j.

В данной работе будут рассматриваться только гиперграфы, множество вершин которых X есть множество элементов $(n \times m)$-матрицы, а множество гиперрёбер E — все её подматрицы порядка 2. Получившийся гиперграф обозначим $G_{n\times m}$ и назовём *гиперграфом подматриц*. Всюду далее предполагается, что $n, m \geqslant 2$.

Заметим, что каждое гиперребро $G_{n\times m}$ содержит ровно 4 вершины (т. е. $G_{n\times m}$ — 4-однородный гиперграф), всего имеется $\frac{1}{4}nm(n-1)(m-1)$ гиперрёбер, а степень каждой вершины $G_{n\times m}$ равна $r=(n-1)(m-1)$.

Раскраску f вершин гиперграфа $G_{n\times m}$ будем представлять в виде $(n\times m)$-матрицы A с элементами $a_{ij}=f(x_{ij})$. Строки и столбцы этой матрицы будем называть *линиями*. Отметим, что перестановки строк и столбцов не влияют на свойство раскраски быть совершенной.

Полупараметрами инцидентности совершенной k-раскраски гиперграфа $G_{n\times m}$ назовём матрицу V размера $k\times L$ такую, что элемент v_{ij} равен числу гиперрёбер цветового состава j, инцидентных вершине цвета i. Здесь L — число всех возможных цветовых составов гиперрёбер $G_{n\times m}$, равное

$$L=k+k(k-1)+\binom{k}{2}+k\binom{k-1}{2}+\binom{k}{4}.$$

Совершенная 2-раскраска $f\colon X\to\{0,1\}$ вершин гиперграфа $G_{n\times m}$ имеет матрицу полупараметров инцидентности V размеров 2×5. При этом под цветовым составом (или просто цветом) $j\in\{0,1,2,3,4\}$ гиперребра будем понимать число вершин цвета 1 в нём. Отметим, что $v_{04}=v_{10}=0$.

Кроме того, для любой совершенной k-раскраски гиперграфа $G_{n\times m}$ сумма элементов в любой строке матрицы полупараметров инцидентности V равна степени гиперграфа G:

$$\sum_{j=0}^{L-1}v_{ij}=(n-1)(m-1),\quad i\in\overline{0,k-1}.$$

Для 2-раскрасок инверсия цветов вершин $f(x)\to f(x)\oplus 1$ сохраняет свойство раскраски быть совершенной, при этом полупараметры инцидентности изменяются по правилу $v_{i,j}\to v_{1-i,4-j}$. Если раскраска $G_{n\times m}$ была совершенной, то транспонированная раскраска гиперграфа $G_{m\times n}$ тоже будет совершенной.

2. Соотношения на полупараметры инцидентности совершенных 2-раскрасок $G_{n\times m}$

Обозначим через n_i, $i\in\{0,1\}$, число вершин цвета i в 2-раскраске гиперграфа $G_{n\times m}$.

Утверждение 1. *Пусть V — полупараметры инцидентности совершенной 2-раскраски $G_{n\times m}$. Тогда*

1) $v_{02}\neq 0$, $v_{12}\neq 0$ *и* $\frac{v_{02}}{v_{12}}=\frac{n_1}{n_0}$;

2) v_{01} *и* v_{11} *одновременно равны или не равны* 0; *если не равны, то* $\frac{v_{01}}{v_{11}}=\frac{3n_1}{n_0}$;

3) v_{03} *и* v_{13} *одновременно равны или не равны* 0; *если не равны, то* $\frac{v_{03}}{v_{13}} = \frac{n_1}{3n_0}$.

Доказательство. 1) В силу того, что раскраска не одноцветная, найдётся линия, содержащая и 0, и 1. Если $v_{02} = 0$, то все гиперрёбра, содержащие 0 и 1, имеют либо цвет 1, либо цвет 3. В таком случае имеем раскраску, в которой все строки (или столбцы), кроме одной, одноцветные, но которая не будет совершенной; противоречие.

Посчитаем число всех гиперрёбер цвета 2 в нашей совершенной 2-раскраске $G_{n\times m}$. Поскольку они содержат равно по две вершины каждого цвета, их число равно $\frac{1}{2}n_0v_{02} = \frac{1}{2}n_1v_{12}$, откуда $\frac{v_{02}}{v_{12}} = \frac{n_1}{n_0}$.

2) Заметим, что число гиперрёбер цвета 1 в совершенной 2-раскраске равно $\frac{1}{3}n_0v_{01} = n_1v_{11}$, откуда $\frac{v_{01}}{v_{11}} = \frac{3n_1}{n_0}$.

3) Аналогично число гиперрёбер цвета 3 равно $n_0v_{03} = \frac{1}{3}n_1v_{13}$, откуда $\frac{v_{03}}{v_{13}} = \frac{n_1}{3n_0}$. Утверждение 1 доказано.

Следствие 1. *Для совершенной* 2*-раскраски гиперграфа* $G_{n\times m}$

$$n_0 = \frac{v_{12}}{v_{12}+v_{02}}nm, \quad n_1 = \frac{v_{02}}{v_{12}+v_{02}}nm.$$

Доказательство следует из равенств $\frac{n_1}{n_0} = \frac{v_{02}}{v_{12}}$ и $n_0 + n_1 = mn$.

Теорема 1. *Для полупараметров инцидентности совершенной* 2*-раскраски* $G_{n\times m}$ *выполнены неравенства*

$$v_{11} \leqslant v_{01} + v_{00}, \quad v_{03} \leqslant v_{13} + v_{14}.$$

Доказательство. Для $v_{03} = v_{11} = 0$ соотношения, очевидно, выполнены. Без ограничения общности докажем первое неравенство, так как второе получается из него инверсией цветов совершенной раскраски.

Пусть $v_{11} \neq 0$. Приведём раскраску к виду

$$\begin{bmatrix}
1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1 & \dots & 1 \\
0 & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 0 & \dots & 0 & 1 & \dots & 1 \\
\mathbf{0} & \mathbf{0} & & A & & & B & & & * & & & * & \\
\mathbf{0} & \mathbf{1} & & C & & & D & & & * & & & * & \\
\mathbf{1} & \mathbf{0} & & * & & & * & & & * & & & * & \\
\mathbf{1} & \mathbf{1} & & * & & & * & & & * & & & * &
\end{bmatrix},$$

где $\mathbf{0}$ и $\mathbf{1}$ — столбцы подходящих размеров. Обозначим через N суммарное число нулей в подматрицах A, B, C и D. Посчитав число гиперрёбер цвета 1, содержащих единицу из первой строки и первого столбца, получаем $v_{11} = N$.

В свою очередь, для нуля из первой строки второго столбца нули в этих областях дают вклад в полупараметры инцидентности v_{01} и v_{00},

однако, не обязательно исчерпывают все гиперрёбра цветов 0 и 1, инцидентные рассматриваемой вершине цвета 0. Теорема 1 доказана.

Утверждение 2. *Пусть f — совершенная 2-раскраска $G_{n\times m}$, в которой число вершин цвета $i \in \{0,1\}$ равно n_i. Если $3n_1 \leqslant n_0$, то $v_{14} < v_{00}$.*

Доказательство. Поскольку в каждой строке матрицы полупараметров сумма элементов равна степени гиперграфа $G_{n\times m}$, то

$$v_{00} + v_{01} + v_{02} + v_{03} = v_{11} + v_{12} + v_{13} + v_{14}.$$

Из утверждения 1 следует, что $v_{01} = \frac{3n_1}{n_0}v_{11}$, $v_{02} = \frac{n_1}{n_0}v_{12}$ и $v_{03} = \frac{n_1}{3n_0}v_{13}$, поэтому

$$v_{00} - v_{14} = v_{11}\Big(1 - \frac{3n_1}{n_0}\Big) + v_{12}\Big(1 - \frac{n_1}{n_0}\Big) + v_{13}\Big(1 - \frac{n_1}{3n_0}\Big) > 0.$$

Утверждение 2 доказано.

Определение 1. Раскраску гиперграфа $G_{n\times m}$ назовём *раскраской линиями,* если она состоит из одноцветных строк или столбцов.

Раскраску $G_{n\times m}$ назовём *блочной,* если перестановками строк и столбцов её можно привести к виду

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix},$$

где $\mathbf{0}$ и $\mathbf{1}$ — прямоугольные матрицы одинаковых размеров.

Утверждение 3. *Все раскраски линиями и блочные раскраски совершенные. Раскраска линиями гиперграфа $G_{n\times m}$, в которой t столбцов окрашены в цвет 0, а $m-t$ столбцов — в цвет 1, имеет полупараметры инцидентности*

$$\begin{bmatrix} (n-1)(t-1) & 0 & (n-1)(m-t) & 0 & 0 \\ 0 & 0 & (n-1)t & 0 & (n-1)(m-t-1) \end{bmatrix}.$$

Блочная раскраска $G_{n\times m}$ существует, только если n и m чётны, и имеет полупараметры инцидентности

$$\begin{bmatrix} \big(\frac{n}{2}-1\big)\big(\frac{m}{2}-1\big) & 0 & \frac{3}{4}nm - \frac{n+m}{2} & 0 & 0 \\ 0 & 0 & \frac{3}{4}nm - \frac{n+m}{2} & 0 & \big(\frac{n}{2}-1\big)\big(\frac{m}{2}-1\big) \end{bmatrix}.$$

Более того, если совершенная 2-раскраска гиперграфа $G_{n\times m}$ имеет полупараметры инцидентности вида

$$\begin{bmatrix} a & 0 & b & 0 & 0 \\ 0 & 0 & c & 0 & d \end{bmatrix},$$

то она либо является раскраской линиями, либо блочной раскраской.

Доказательство. Заметим, что ввиду отсутствия гиперрёбер нечётного цвета любая пара строк (пара столбцов) в раскраске либо совпадает, либо одна строка (столбец) получается инверсией цветов второй строки (столбца).

Таким образом, перестановками строк и столбцов раскраска (транспонированная раскраска) приводится к одному из двух видов:

$$\begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix},$$

где $\mathbf{0}$ и $\mathbf{1}$ — прямоугольные матрицы одинаковых размеров. В первом случае имеем совершенную раскраску линиями. Во втором случае можно проверить, что если блоки из нулей и единиц имеют не одинаковый размер, то раскраска не совершенная. Совершенность данных раскрасок и их полупараметры инцидентности находятся непосредственно. Утверждение 3 доказано.

Следствие 2. *Если в совершенной* 2*-раскраске гиперграфа* $G_{n\times m}$, *где* $n > 2$ *или* $m > 2$, *есть одноцветная строка или столбец, то она является раскраской линиями.*

Доказательство. Без ограничения общности предположим, что все элементы в первой строке раскраски имеют цвет 0. Заметим, что через любой нуль этой строки не проходят гиперрёбра цвета 3, поэтому $v_{03} = v_{13} = 0$, где v_{ij} — элементы матрицы полупараметров V.

Если рассматриваемая раскраска — не раскраска линиями, то найдётся строка, в которой есть вершины как цвета 0, так и цвета 1. Выберем среди них такую строку, что она содержит минимальное число t нулей, $t \neq 0$. Без ограничения общности можно считать, что это вторая строка раскраски. Переставив строки и столбцы, приведём раскраску к виду

$$\begin{bmatrix} 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & \dots & 1 \\ \mathbf{0} & & A & & & B & \\ \mathbf{1} & & C & & & D & \end{bmatrix}.$$

Посчитаем число гиперрёбер v_{00} цвета 0, проходящих через вершину цвета 0. Для элемента, находящегося в первой строке и в первом столбце, это число равно $t - 1 + \alpha + \beta$, где α и β — число нулей в областях A и B соответственно. С другой стороны, для нуля, находящегося во второй строке и первом столбце, число гиперрёбер цвета 0 равно $t - 1 + \alpha$. Отсюда следует, что $\beta = 0$ и подматрица B заполнена единицами.

Так как вторая строка содержит минимальное отличное от нуля число t нулей в строках, то либо $t = 1$ и в подматрице A нет элементов, либо

вся подматрица A заполнена нулями. В первом случае в подматрице C тоже нет элементов, а условие $v_{03} = v_{13} = 0$ влечёт, что подматрица D полностью заполнена нулями. Тогда несложно видеть, что получившаяся раскраска не будет совершенной, так как при $m \geqslant 3$ одни нулевые элементы будут инцидентны гиперрёбрам цвета 0, а другие — нет, а все совершенные раскраски $G_{2\times m}$ и $G_{n\times 2}$ описаны в утверждении 5.

Предположим, что реализуется второй вариант и подматрица A полностью заполнена нулями. Снова из условия $v_{03} = v_{13} = 0$ следует, что подматрица D полностью заполнена нулями. Заметим, что в таком случае число v_{02} гиперрёбер цвета 2, инцидентных элементу в первой строке и в первом столбце, равно числу единиц в подматрице C. С другой стороны, для нуля из второй строки и первого столбца инцидентных ему гиперрёбер цвета 2 будет больше, чем число единиц в подматрице C, так как дополнительно возникнут гиперрёбра, использующие единицы из подматрицы B. Полученное противоречие завершает доказательство. Следствие 2 доказано.

3. Конфигурации линий и конфигурационно однородные раскраски

Определение 2. Будем говорить, что различные строки A_i и A_j в 2-раскраске гиперграфа $G_{n\times m}$ образуют *конфигурацию* $[A_i, A_j] = \big(t_{ij}^{00}, t_{ij}^{01}, t_{ij}^{10}, t_{ij}^{11}\big)$, если матрица, составленная из этой пары строк, содержит из t_{ij}^{00} столбцов $\begin{bmatrix}0\\0\end{bmatrix}$, t_{ij}^{01} столбцов $\begin{bmatrix}0\\1\end{bmatrix}$, t_{ij}^{10} столбцов $\begin{bmatrix}1\\0\end{bmatrix}$ и t_{ij}^{11} столбцов $\begin{bmatrix}1\\1\end{bmatrix}$.

Аналогично определяется конфигурация $[A^i, A^j]$ для пары различных столбцов A^i и A^j раскраски.

Пример 1. Раскраска с конфигурацией строк $[A_1, A_2] = (2, 3, 3, 3)$:

$$\begin{bmatrix}A_1\\A_2\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1\end{bmatrix}.$$

3.1. Необходимое условие существования совершенных 2-раскрасок в терминах конфигураций.

Теорема 2. *Пусть f — совершенная 2-раскраска гиперграфа $G_{n\times m}$, отличная от раскраски линиями, $n, m \geqslant 3$. Тогда для конфигурации двух строк (столбцов) с номерами i и j, $i \neq j$, справедливо равенство $t_{ij}^{01} = t_{ij}^{10}$.*

Доказательство проведём для конфигурации строк, для столбцов доказательство аналогично.

Рассмотрим сначала две строки, для которых $t_{ij}^{00} \neq 0$ (случай $t_{ij}^{11} \neq 0$ рассматривается аналогично).

Так как рассматриваемая раскраска не является раскраской линиями, по следствию 2 она не содержит одноцветных линий, а значит, перестановками строк и столбцов её можно привести к виду

$$\begin{bmatrix} 0 & 0 & \dots & 0 & 1 & \dots & 1 & 0 & \dots & 0 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1 & \dots & 1 & 0 & \dots & 0 \\ \mathbf{0} & & A & & & B & & & C & & & D & \\ \mathbf{1} & & A' & & & B' & & & C' & & & D' & \end{bmatrix}.$$

Заметим, что число столбцов в матрицах A и A' равно t_{12}^{00}, в матрицах B и B' — t_{12}^{11}, в C и C' — t_{12}^{01}, а в D и D' — t_{12}^{10}.

Без ограничения общности докажем, что для конфигураций первых двух строк выполнено $t_{12}^{01} = t_{12}^{10}$.

Обозначим через $\alpha_i, \beta_i, \gamma_i, \delta_i, \alpha_i', \beta_i', \gamma_i', \delta_i'$ число вершин цвета $i \in \{0, 1\}$ в матрицах $A, B, C, D,\ A', B', C', D'$ соответственно.

Так как раскраска совершенная, число гиперрёбер цвета 0, проходящих через нули в первом столбце и первой или второй строке, одинаково:

$$t_{12}^{00} + \alpha_0 + \gamma_0 = t_{12}^{00} + \alpha_0 + \delta_0.$$

Отсюда заключаем, что $\gamma_0 = \delta_0$.

Аналогично число гиперрёбер цвета 3, проходящих через нули в первом столбце и первой или второй строке, совпадает:

$$\beta_1' + \delta_1' = \beta_1' + \gamma_1',$$

откуда $\gamma_1' = \delta_1'$.

Посчитаем и приравняем теперь число гиперрёбер цвета 1, проходящих через нули в первом столбце и первой и второй строке:

$$t_{12}^{01} + t_{12}^{10} + \alpha_1 + \beta_0 + \gamma_1 + \delta_0 + \alpha_0' + \gamma_0' = t_{12}^{01} + t_{12}^{10} + \alpha_1 + \beta_0 + \gamma_0 + \delta_1 + \alpha_0' + \delta_0',$$

откуда

$$\gamma_1 + \delta_0 + \gamma_0' = \gamma_0 + \delta_1 + \delta_0'.$$

Используя соотношения $\gamma_0 = \delta_0$ и $\gamma_1' = \delta_1'$, меняем в этом равенстве γ_0 и δ_0 местами, добавляем γ_1' в левую часть, а δ_1' — в правую:

$$\gamma_1 + \gamma_0 + \gamma_0' + \gamma_1' = \gamma_0 + \gamma_1 + \delta_0' + \delta_1'.$$

Другими словами, суммарное число элементов в матрицах C и C' равно суммарному числу элементов в матрицах D и D', что ввиду равенства числа строк этих матриц возможно лишь при совпадении числа их столбцов, т. е. $t_{12}^{01} = t_{12}^{10}$.

Предположим теперь, что для конфигурации строк i и j выполнено $t_{ij}^{00} = t_{ij}^{11} = 0$ и существует строка $k \notin \{i, j\}$ такая, что подматрица,

образованная этими тремя строками, имеет вид

$$\begin{matrix} i \\ j \\ k \end{matrix} \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1 & \dots & 1 \\ 1 & \dots & 1 & 1 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}.$$

Тогда t_{ik}^{00} или t_{ik}^{11} не равно нулю и t_{jk}^{00} или t_{jk}^{11} не равно нулю. Из предыдущих рассуждений следует, что $t_{ik}^{01} = t_{ik}^{10}$ и $t_{jk}^{01} = t_{jk}^{10}$. Остаётся только заметить, что $t_{ij}^{01} = t_{ik}^{01} + t_{jk}^{10}$ и $t_{ij}^{10} = t_{ik}^{10} + t_{jk}^{01}$, откуда $t_{ij}^{01} = t_{ij}^{10}$.

Наконец, если не существует строки, отличной от строк i или j, то либо $n = 2$, либо все строки совпадают со строками i и j. Первый случай исключается условием $n \geqslant 3$, описание совершенных 2-раскрасок $G_{2\times m}$ будет отдельно выполнено в утверждении 5. Во втором случае получается блочная раскраска, все возможные параметры которой описаны в утверждении 3. Теорема 2 доказана.

Следствие 3. *Любая совершенная* 2*-раскраска гиперграфа* $G_{n\times m}$*, отличная от раскраски линиями, имеет одинаковое число единиц в каждой строке и в каждом столбце.*

Доказательство проведём для строк: для столбцов аналогично.

Так как раскраска отлична от раскраски линиями, по теореме 2 для конфигурации строк с номерами 1 и $j \neq 1$ верно $t_{1j}^{01} = t_{1j}^{10}$. Заметим, что число единиц в строке с номером j равно $t_{1j}^{11} + t_{1j}^{01} = t_{1j}^{11} + t_{1j}^{10}$, что совпадает с числом единиц в первой строке. Таким образом, все строки совершенной раскраски содержат одинаковое число единиц. Следствие 3 доказано.

Замечание 1. Существуют совершенные 2-раскраски $G_{n\times m}$ (пример 3), в которых строки находятся разных конфигурациях, т. е. $t_{ij}^{01} \neq t_{ik}^{10}$ для различных i, j, k.

Замечание 2. В случае k-раскраски, где $k \geqslant 3$, условие вида $t_{ij}^{01} = t_{ij}^{10}$ не является необходимым (пример 2).

3.2. Конфигурационно однородные раскраски.

Определение 3. Раскраску гиперграфа $G_{n\times m}$ назовём *конфигурационно однородной по строкам*, если для любых её конфигураций строк выполнено $[A_i, A_j] = [A_k, A_l]$, где $i, j, k, l \in \{1, 2, \dots, n\}$, $i \neq j$, $k \neq l$.

Аналогично определяется конфигурационная однородность по столбцам.

Назовём раскраску *конфигурационно однородной,* если она конфигурационно однородна и по строкам, и по столбцам, но, возможно, с различными конфигурациями.

Так как по теореме 2 для любых двух строк 2-раскрасок выполнено $t_{ij}^{01} = t_{ij}^{10}$, можно считать, что конфигурация строк совершенных конфигурационно однородных по строкам 2-раскрасок задаётся тройкой чисел, которую для удобства будем записывать как (t_{00}, t_{01}, t_{11}).

Приведём описание конфигурационно однородных по строкам совершенных 2-раскрасок и свяжем их конфигурации строк с полупараметрами инцидентности. Для конфигурационно однородных совершенных раскрасок в большее число цветов утверждение аналогично, но не будет приведено в силу своей громоздкости.

Теорема 3. *Конфигурационно однородная по строкам 2-раскраска гиперграфа $G_{n\times m}$ с числом единиц в каждом столбце, равным r, совершенная. Кроме того, если (t_{00}, t_{01}, t_{11}) — конфигурация строк такой совершенной раскраски, то $t_{01}^2 > t_{00}t_{11}$ и число строк n, величина r и полупараметры инцидентности однозначно выражаются через t_{00}, t_{01} и t_{11}.*

Доказательство. Из конфигурационной однородности и равенства числа единиц в столбцах следует, что каждая вершина цвета i инцидентна одному и тому же числу гиперрёбер цвета j. При этом полупараметры инцидентности задаются следующим образом:

$$v_{00} = (t_{00}-1)(n-r-1), \quad v_{01} = 2t_{01}(n-r-1) + rt_{00},$$
$$v_{02} = (n-r-1)t_{11} + r(2t_{01}-1), \quad v_{03} = rt_{11},$$
$$v_{11} = (n-r)t_{00}, \quad v_{12} = (r-1)t_{00} + (n-r)(2t_{01}-1),$$
$$v_{13} = 2(r-1)t_{01} + (n-r)t_{11}, \quad v_{14} = (r-1)(t_{11}-1).$$

Заметим, что отношение $\frac{n_1}{n_0}$ числа вершин цвета 1 к числу вершин цвета 0 в раскраске равно $\frac{r}{n-r}$. По утверждению 1 имеем

$$\frac{v_{01}}{v_{11}} = \frac{3r}{n-r}, \quad \frac{v_{03}}{v_{13}} = \frac{r}{3(n-r)}.$$

Выражая из этих двух равенств r, находим

$$r = \frac{t_{01}}{t_{00}+t_{01}}(n-1) = \frac{nt_{11}+t_{01}}{t_{11}+t_{01}}.$$

Решая последнее равенство относительно n и используя соотношение $m = t_{00} + 2t_{01} + t_{11}$, получаем

$$n = \frac{t_{01}}{t_{01}^2 - t_{00}t_{11}}m,$$

откуда, в частности, следует, что $t_{01}^2 > t_{00}t_{11}$. Осталось выразить все оставшиеся полупараметры совершенной раскраски через t_{00}, t_{01} и t_{11}:

$$r = \frac{t_{01}^2 + t_{01}t_{11}}{t_{01}^2 - t_{00}t_{11}},$$

$$v_{00} = \frac{t_{00}(t_{00}-1)(t_{01}+t_{11})}{t_{01}^2 - t_{00}t_{11}}, \quad v_{01} = \frac{3t_{00}t_{01}(t_{01}+t_{11})}{t_{01}^2 - t_{00}t_{11}},$$
$$v_{02} = \frac{(t_{01}+t_{11})\big(2t_{01}^2 + t_{00}t_{11} - t_{01}\big)}{t_{01}^2 - t_{00}t_{11}}, \quad v_{03} = \frac{t_{01}t_{11}(t_{01}+t_{11})}{t_{01}^2 - t_{00}t_{11}},$$
$$v_{11} = \frac{t_{00}t_{01}(t_{00}+t_{01})}{t_{01}^2 - t_{00}t_{11}}, \quad v_{12} = \frac{(t_{00}+t_{01})\big(2t_{01}^2 + t_{00}t_{11} - t_{01}\big)}{t_{01}^2 - t_{00}t_{11}},$$
$$v_{13} = \frac{3t_{01}t_{11}(t_{00}+t_{01})}{t_{01}^2 - t_{00}t_{11}}, \quad v_{14} = \frac{t_{11}(t_{11}-1)(t_{00}+t_{01})}{t_{01}^2 - t_{00}t_{11}}.$$

Теорема 3 доказана.

Замечание 3. Целочисленность r и полупараметров инцидентности накладывает дополнительные условия на конфигурации строк совершенных конфигурационно однородных по строкам 2-раскрасок.

Замечание 4. Существуют не конфигурационно однородные совершенные раскраски, как двуцветные, так и многоцветные. Примером таких 2-раскрасок служат блочные раскраски, а многоцветных — раскраски, полученные конструкцией из п. 4.2 и объединением подходящих цветов (под объединением цветов подразумевается перекрашивание вершин этих цветов в новый цвет).

Пример 2. Совершенная, но не конфигурационно однородная 3-раскраска:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Приведём также бесконечную серию не конфигурационно однородных совершенных 2-раскрасок.

Пример 3. Пусть $k \geqslant 2$. Рассмотрим 2-раскраски гиперграфа $G_{2k\times k^2}$ вида

$$\begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 1 & \dots & 1 \\ & I & & & I & & \dots & & I & \end{bmatrix},$$

где каждая группа столбцов имеет ширину k, число таких групп равно k, а I — единичная матрица порядка k.

Заметим, что для различных $1 \leqslant i, j \leqslant 2k$ имеем конфигурацию

$$[A_i, A_j] = \begin{cases} (k(k-2), k, 0), & \text{если } 1 \leqslant i, j \leqslant k \\ & \text{или } k{+}1 \leqslant i, j \leqslant 2k, \\ ((k-1)^2, k-1, 1), & \text{если } 1 \leqslant i \leqslant k,\ k{+}1 \leqslant j \leqslant 2k \\ & \text{или } k{+}1 \leqslant i \leqslant 2k,\ 1 \leqslant j \leqslant k. \end{cases}$$

Полупараметры инцидентности раскраски равны

$$\begin{bmatrix} (k-2)(2k^2-3k-1) & 3(2k^2-4k+1) & 5(k-1) & 1 & 0 \\ 0 & (k-1)(2k^2-4k+1) & 5(k-1)^2 & 3(k-1) & 0 \end{bmatrix}.$$

3.3. Комбинаторные схемы как конфигурационно однородные раскраски.

Определение 4. 2-(v, k, λ)-*Схемой* называется множество блоков B, в котором каждый блок есть k-элементное подмножество множества точек X, $|X| = v$, причём любая пара различных точек содержится ровно в λ блоках. Число блоков в 2-схеме обозначается через b, а число блоков, содержащих фиксированную точку из X, — через r. Схема может содержать кратные блоки, поэтому, говоря общо, B — мультимножество. Тройку чисел (v, k, λ) будем называть *параметрами* 2-схемы.

Для 2-схем с параметрами (v, k, λ) числа r и b определяются соотношениями

$$r = \lambda\frac{v-1}{k-1}, \quad b = \lambda\frac{v(v-1)}{k(k-1)}.$$

Определение 5. *Матрицей инцидентности* 2-(v, k, λ)-схемы называется матрица A размера $v \times b$, строки и столбцы которой занумерованы точками из X и блоками из B соответственно, при этом $a_{ij} = 1$, если точка i принадлежит блоку j, и $a_{ij} = 0$ иначе.

Теорема 4. *Матрица инцидентности* 2-(v, k, λ)-*схемы соответствует совершенной конфигурационно однородной по строкам* 2-*раскраске гиперграфа* $G_{v\times b}$ *с конфигурацией строк*

$$\left(\lambda\frac{(v-k)(v-k-1)}{k(k-1)}, \lambda\frac{v-k}{k-1}, \lambda\right).$$

Более того, совершенные конфигурационно однородные по строкам 2-*раскраски гиперграфов подматриц* (*отличные от раскрасок линиями*) *находятся во взаимно однозначном соответствии с матрицами инцидентности* 2-*схем.*

Доказательство. Пусть имеется совершенная раскраска гиперграфа $G_{n\times m}$, конфигурационно однородная по строкам с конфигурацией строк (t_{00}, t_{01}, t_{11}) и отличная от раскраски линиями. Представим её как матрицу инцидентности точек из некоторого множества X, $|X| = n$,

и блоков из некоторого множества B, $|B| = m$. Пусть строки матрицы занумерованы точками из X, а столбцы — блоками из B. Покажем, что B — это 2-схема.

По следствию 3 все столбцы раскраски содержат одинаковое число единиц, которое обозначим через k. Заметим, что пара различных точек i и j принадлежит t_{ij}^{11} блокам, поэтому из конфигурационной однородности следует, что любая пара различных точек принадлежит ровно t_{11} блокам, а значит, B является 2-схемой.

Докажем соответствие в обратную сторону. По определению матрица A инцидентности 2-схемы имеет размер $v \times b$, каждый её столбец содержит k единиц, а строка — r единиц.

Пусть $[A_i, A_j] = \left(t_{ij}^{00}, t_{ij}^{01}, t_{ij}^{11}\right)$ — конфигурации строк с номерами i и j. Так как любые точки из множества X принадлежат ровно λ блокам, для любых i и j имеем $t_{ij}^{11} = \lambda$. Далее, поскольку в каждой строке матрицы A содержится ровно r единиц, то $t_{ij}^{01} = r - \lambda = \lambda\frac{v-k}{k-1}$ для всех i и j. Отсюда $t_{ij}^{00} = b - 2(r-\lambda) - \lambda = \lambda\frac{(v-k)(v-k-1)}{k(k-1)}$.

Таким образом, матрица инцидентности A является конфигурационно однородной по строкам раскраской с одинаковым числом единиц в каждом столбце, а значит, по теореме 3 она будет совершенной. Теорема 4 доказана.

Из доказательства теоремы 3 также получаем

Следствие 4. *Матрица инцидентности* 2-(v, k, λ)*-схемы является совершенной* 2*-раскраской гиперграфа* $G_{v\times b}$ *с полупараметрами инцидентности*

$$v_{00} = (v-k-1)\left(\lambda\frac{(v-k)(v-k-1)}{k(k-1)} - 1\right),$$

$$v_{01} = 3\lambda\frac{(v-k)(v-k-1)}{k-1},$$

$$v_{02} = \frac{1}{k-1}(3\lambda k(v-k) - \lambda(v-1) - k(k-1)),$$

$$v_{03} = \lambda k, \quad v_{11} = \lambda\frac{(v-k)^2(v-k-1)}{k(k-1)},$$

$$v_{12} = \frac{v-k}{k(k-1)}(3\lambda k(v-k) - \lambda(v-1) - k(k-1)),$$

$$v_{13} = 3\lambda(v-k), \quad v_{14} = (\lambda-1)(k-1).$$

Замечание 5. Матрица инцидентности 2-схемы не обязательно конфигурационно однородна по столбцам, что показывает пример ниже.

Пример 4. Матрица инцидентности 2-схемы с параметрами $(6, 3, 2)$ имеет вид

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Важным частным случаем 2-схем являются *симметричные* 2-схемы, в которых число блоков b совпадает с числом точек v.

Утверждение 4. *Матрица инцидентности симметричной* 2-(v, k, λ)-*схемы является конфигурационно однородной по строкам и столбцам совершенной* 2-*раскраской гиперграфа* $G_{v\times v}$, *строки и столбцы которой имеют конфигурацию* $(v - 2k + \lambda, k - \lambda, \lambda)$, *а полупараметры инцидентности равны*

$$\begin{gathered} v_{00} = (v - k - 1)(v - 2k + \lambda - 1), \\ v_{01} = k(v - 2k + \lambda) + 2(v - k - 1)(k - \lambda), \\ v_{02} = \lambda(v - k - 1) + k(2k - 2\lambda - 1), \quad v_{03} = \lambda k, \\ v_{11} = (v - k)(v - 2k + \lambda), \\ v_{12} = (k - 1)(v - 2k + \lambda) + (v - k)(2k - 2\lambda - 1), \\ v_{13} = 2(k - 1)(k - \lambda) + \lambda(v - k), \quad v_{14} = (\lambda - 1)(k - 1). \end{gathered}$$

Доказательство. Совершенность раскраски и конфигурационная однородность по строкам напрямую следуют из теоремы 4, вид конфигураций линий и конфигурационная однородность по столбцам следуют из того, что в симметричных 2-схемах любая пара блоков имеет одинаковое число λ общих точек. Следствие 4 доказано.

Пример 5. Матрица инцидентности симметричной 2-схемы с параметрами $(7, 3, 1)$ представляет собой совершенную 2-раскраску гиперграфа $G_{7\times 7}$:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

с конфигурациями линий $(2, 2, 1)$. Полупараметры инцидентности такой 2-раскраски имеют вид

$$\begin{bmatrix} 3 & 18 & 12 & 3 & 0 \\ 0 & 8 & 16 & 12 & 0 \end{bmatrix}.$$

Рассмотрим несколько примеров симметричных 2-схем и найдём их параметры как совершенных раскрасок. Начнём с вырожденной схемы, матрица инцидентности которой представляет собой диагональную матрицу.

Пример 6. Единичная матрица I порядка n является конфигурационно однородной совершенной 2-раскраской $G_{n\times n}$ с конфигурациями линий $(n-2, 1, 0)$ и полупараметрами инцидентности

$$\begin{bmatrix} (n-2)(n-3) & 3(n-2) & 1 & 0 & 0 \\ 0 & (n-1)(n-2) & n-1 & 0 & 0 \end{bmatrix}.$$

Определение 6. *Матрицей Адамара* порядка n называется квадратная матрица H размера $n \times n$ с элементами 1 и -1, удовлетворяющая соотношению $HH^\top = nI$, где $H^\top$ — транспонированная матрица H, а I — единичная матрица порядка n.

Приведём несколько хорошо известных свойств матриц Адамара, которые можно найти, например, в [12].

Матрицы Адамара могут существовать только для порядков $n = 4k$. Любую матрицу Адамара перестановками строк и столбцов, а также умножением их на -1 можно привести к стандартной форме, в которой первая строка и первый столбец состоят из 1.

Если из матрицы Адамара порядка $4k$, находящейся в стандартной форме, удалить первую строку и первый столбец и заменить все вхождения -1 на 0, то получится $(0, 1)$-матрица H', которая будет матрицей инцидентности симметричной 2-схемы с параметрами $(4k-1, 2k-1, k-1)$. Такая схема называется называется 2-*схемой Адамара*.

По утверждению 4 2-схема Адамара, полученная из матрицы Адамара порядка $4k$, даёт пример совершенной 2-раскраски $G_{(4k-1)\times(4k-1)}$ с конфигурациями линий $(k, k, k-1)$ и полупараметрами инцидентности

$$\begin{gathered} v_{00} = (2k-1)(k-1), \quad v_{01} = 3k(2k-1), \\ v_{02} = (2k-1)(3k-2), \quad v_{03} = (k-1)(2k-1), \\ v_{11} = 2k^2, \quad v_{12} = 2k(3k-2), \\ v_{13} = 6k(k-1), \quad v_{14} = 2(k-2)(k-1). \end{gathered}$$

4. Конструкции совершенных раскрасок гиперграфа подматриц

4.1. Произведение Кронекера.

Определение 7. *Произведение Кронекера* матрицы A размера $n \times m$ и матрицы B размера $k \times l$ есть матрица $A \otimes B$ размера $nk \times ml$ вида

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & & \vdots \\ a_{n1}B & \dots & a_{nm}B \end{bmatrix}.$$

Умножение Кронекера на матрицу из единиц позволяет получить бесконечную серию совершенных раскрасок из любой матрицы конфигурационно однородной раскраски.

Теорема 5. *Пусть A — совершенная 2-раскраска гиперграфа $G_{n\times m}$, конфигурационно однородная по строкам (столбцам) с конфигурацией строк (столбцов) (t_{00}, t_{01}, t_{11}) и столбцовыми (строчными) суммами, равными k. Тогда матрица $A \otimes E$ есть совершенная 2-раскраска $G_{nl\times mr}$, где E — матрица размера $l \times r$, все элементы которой равны единице.*

Доказательство. Докажем теорему для случая, когда матрица A конфигурационно однородна по строкам. Для случая конфигурационной однородности по столбцам доказательство аналогично.

Поскольку умножение Кронекера ассоциативно, то

$$A \otimes E = A \otimes \left(\begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right) = \left(A \otimes \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \right) \otimes \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

При умножении матрицы A на вектор-строку из единиц длины r

$$B = A \otimes \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$$

достаточно заметить, что столбцовые суммы не изменятся, а получившаяся матрица B будет конфигурационно однородной по строкам с конфигурацией строк $(rt_{00}, rt_{01}, rt_{11})$, а значит, B — совершенная раскраска.

Умножим теперь матрицу B на вектор-столбец высоты $l \geqslant 1$:

$$C = B \otimes \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Получившаяся раскраска C совершенна, что следует непосредственно из подсчёта полупараметров инцидентности v_{ij}, которые будут иметь вид

$$v_{00} = (l-1)(rt_{00} + rt_{01} - 1) + l(n-k-1)(rt_{00} - 1),$$

$$v_{01} = 2lr(n-k-1)t_{01} + lkrt_{00},$$
$$v_{02} = r(l-1)(t_{11}+t_{01}) + lr(n-k-1)t_{11} + lk(2rt_{01}-1),$$
$$v_{03} = lkrt_{11}, \quad v_{11} = lr(n-k)t_{00},$$
$$v_{12} = r(l-1)(t_{00}+t_{01}) + l(n-k)(2rt_{01}-1) + lr(k-1)t_{00},$$
$$v_{13} = lr(n-k)t_{11} + 2lr(k-1)t_{01},$$
$$v_{14} = (l-1)(rt_{11}+rt_{01}-1) + l(k-1)(rt_{11}-1).$$

Таким образом, $C = A \otimes E$ будет совершенной раскраской. Теорема 5 доказана.

Замечание 6. Аналогичная теорема справедлива для конфигурационно однородных (одновременно и по строкам, и по столбцам) совершенных раскрасок гиперграфа $G_{n\times m}$ в $k \geqslant 3$ цветов.

Замечание 7. При умножении Кронекера не конфигурационно однородной совершенной 2-раскраски из примера 3 на матрицу E также получается (не конфигурационная однородная) совершенная раскраска, что проверяется подсчётом полупараметров инцидентности.

4.2. Многоцветная блочная раскраска. Рассмотрим обобщение блочной совершенной 2-раскраски $G_{n\times m}$ на большее число цветов. Оно основано на следующем свойстве.

Лемма 1. *Пусть K_n и K_m — полные графы с матрицами инцидентности B_n и B_m соответственно. Тогда $B_n \otimes B_m = B$ — матрица инцидентности гиперграфа $G_{n\times m}$.*

Доказательство. Из определения произведения Кронекера следует, что строки матрицы B соответствуют парам вершин (x, y), где $x \in V(K_n)$, $y \in V(K_m)$, а столбцы — парам рёбер (e, w), где $e \in E(K_n)$, $w \in E(K_m)$.

В гиперграфе с матрицей инцидентности B вершина (x, y) инцидентна гиперребру (e, w) тогда и только тогда, когда x инцидентна e в K_n и y инцидентна w в K_m, а множество гиперрёбер есть множество всех четвёрок вершин гиперграфа вида (x_1, y_1), (x_1, y_2), (x_2, y_1) и (x_2, y_2). Таким образом, B — матрица инцидентности гиперграфа $G_{n\times m}$. Лемма 1 доказана.

Теорема 6. *Пусть f и g — совершенные раскраски графов K_n и K_m в k и l цветов с матрицами раскраски P' и P'' и параметрами инцидентности (V', W') и (V'', W'') соответственно. Тогда $P = P' \otimes P''$ — совершенная раскраска гиперграфа $G_{n\times m}$ в kl цветов с параметрами инцидентности $(V' \times V'', W' \otimes W'')$.*

Доказательство. Пусть B_n и B_m — матрицы инцидентности графов K_n и K_m соответственно. Так как раскраски f и g совершенные, для графов инцидентности рассматриваемых полных графов справедливы равенства

$$\begin{bmatrix} 0 & B_n \\ B_n^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & P' \\ R' & 0 \end{bmatrix} = \begin{bmatrix} 0 & P' \\ R' & 0 \end{bmatrix} \begin{bmatrix} 0 & W' \\ V' & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & B_m \\ B_m^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & P'' \\ R'' & 0 \end{bmatrix} = \begin{bmatrix} 0 & P'' \\ R'' & 0 \end{bmatrix} \begin{bmatrix} 0 & W'' \\ V'' & 0 \end{bmatrix},$$

где R' и R'' — индуцированные раскраски рёбер. Отсюда получаем

$$B_n R' = P'V', \quad B_m R'' = P''V'',$$
$$B_n^\top P' = R'W', \quad B_m^\top P'' = R''W''.$$

Перемножим произведением Кронекера равенства в каждой строке:

$$(B_n R') \otimes (B_m R'') = (P'V') \otimes (P''V''),$$
$$(B_n^\top P') \otimes (B_m^\top P'') = (R'W') \otimes (R''W'').$$

Используя равенство $AB \otimes CD = (A \otimes B)(C \otimes D)$ для матриц согласованных размеров, получаем

$$(B_n \otimes B_m)(R' \otimes R'') = (P' \otimes P'')(V' \otimes V''),$$
$$(B_n^\top \otimes B_m^\top)(P' \otimes P'') = (R' \otimes R'')(W' \otimes W'').$$

Положим $B = B_n \otimes B_m$, $P = P' \otimes P''$, $R = R' \otimes R''$, $V = V' \otimes V''$ и $W = W' \otimes W''$. Тогда из данной системы равенств имеем

$$\begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & P \\ R & 0 \end{bmatrix} = \begin{bmatrix} 0 & P \\ R & 0 \end{bmatrix} \begin{bmatrix} 0 & W \\ V & 0 \end{bmatrix}.$$

По лемме 1 матрица $B = B_n \otimes B_m$ является матрицей инцидентности гиперграфа $G_{n\times m}$, а значит, последнее уравнение эквивалентно тому, что P — это совершенная раскраска $G_{n\times m}$ с параметрами инцидентности (V, W).

Из размеров матрицы P следует, что раскраска гиперграфа $G_{n\times m}$ является раскраской в kl цветов. Теорема 6 доказана.

Замечание 8. Как известно, любая раскраска полного графа совершенна. Раскраска гиперграфа $G_{n\times m}$, построенная описанным образом из совершенных раскрасок графов K_n и K_m, будет многоцветной блочной раскраской.

Пример 7. Построим совершенную раскраску $G_{2\times 3}$ с помощью совершенных раскрасок графов K_2 и K_3. Рассмотрим совершенные раскраски этих графов, которые индуцируют раскраски их графов инцидентности, представленные на рис. 1.

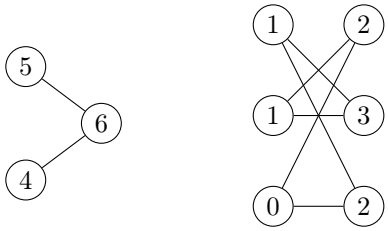


Рис. 1. Индуцированные совершенные раскраски графов инцидентности K_2 и K_3

Для графа K_2 матрица инцидентности равна $B_2 = \begin{bmatrix}1\\1\end{bmatrix}$, матрицы раскрасок вершин и рёбер — $P' = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, $R' = \begin{bmatrix}1\end{bmatrix}$, параметры инцидентности — $V' = \begin{bmatrix}1\\1\end{bmatrix}$, $W' = \begin{bmatrix}1 & 1\end{bmatrix}$. Для графа K_3 матрица инцидентности равна $B_3 = \begin{bmatrix}1 & 0 & 1\\1 & 1 & 0\\0 & 1 & 1\end{bmatrix}$, матрицы раскрасок вершин и рёбер — $P'' = \begin{bmatrix}1 & 0\\1 & 0\\0 & 1\end{bmatrix}$, $R'' = \begin{bmatrix}1 & 0\\0 & 1\\0 & 1\end{bmatrix}$, параметры инцидентности — $V'' = \begin{bmatrix}1 & 1\\0 & 2\end{bmatrix}$, $W'' = \begin{bmatrix}1 & 1\\0 & 2\end{bmatrix}$. Тогда матрица $\begin{bmatrix}0 & 2 & 2\\1 & 3 & 3\end{bmatrix}$ представляет совершенную раскраску гиперграфа $G_{2\times 3}$ с параметрами инцидентности $(V' \otimes V'', W' \otimes W'')$.

5. Совершенные 2-раскраски $G_{2\times m}$ и $G_{3\times m}$

В справедливости утверждения 5 нетрудно убедиться прямым перебором, полупараметры инцидентности вычисляются непосредственно.

Утверждение 5. *Любая раскраска гиперграфа $G_{2\times 2}$ совершенна. При $m > 2$ существуют два следующих типа раскрасок $G_{2\times m}$.*

1) *Раскраска линиями* (*столбцами*)

$$\begin{bmatrix}0 & \dots & 0 & 1 & \dots & 1\\0 & \dots & 0 & 1 & \dots & 1\end{bmatrix}.$$

Если число нулевых столбцов равно t, то полупараметры инцидентности такой совершенной раскраски имеют вид

$$\begin{bmatrix}t-1 & 0 & m-t & 0 & 0\\0 & 0 & t & 0 & m-t-1\end{bmatrix}.$$

2) *Раскраска без одноцветных столбцов:*

$$\begin{bmatrix} 0 & \dots & 0 & 1 & \dots & 1 \\ 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}.$$

Такая раскраска определяется с точностью до инверсии цветов в произвольном наборе столбцов. Полупараметры инцидентности такой совершенной раскраски имеют вид

$$\begin{bmatrix} 0 & 0 & m-1 & 0 & 0 \\ 0 & 0 & m-1 & 0 & 0 \end{bmatrix}.$$

Утверждение 6. *Любая совершенная раскраска гиперграфа $G_{3\times m}$, где $m \geqslant 3$, является одной из приведённых ниже раскрасок с точностью до инверсии цветов и перестановок строк и столбцов.*

1) *Раскраска линиями (столбцами)*

$$\begin{bmatrix} 0 & \dots & 0 & 1 & \dots & 1 \\ 0 & \dots & 0 & 1 & \dots & 1 \\ 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix}.$$

Если число нулевых столбцов равно t, то полупараметры инцидентности такой совершенной раскраски имеют вид

$$\begin{bmatrix} 2(t-1) & 0 & 2(m-t) & 0 & 0 \\ 0 & 0 & 2t & 0 & 2(m-t-1) \end{bmatrix}.$$

2) *Раскраска линиями (строками)*

$$\begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 1 & \dots & 1 \end{bmatrix}.$$

Полупараметры инцидентности такой совершенной раскраски равны

$$\begin{bmatrix} m-1 & 0 & m-1 & 0 & 0 \\ 0 & 0 & 2(m-1) & 0 & 0 \end{bmatrix}.$$

3) *Раскраска, представимая в виде произведения Кронекера*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \dots & 1 \end{bmatrix},$$

где вектор-строка имеет длину $t \geqslant 1$. Полупараметры инцидентности такой совершенной раскраски

$$\begin{bmatrix} t-1 & 3t & 2t-1 & 0 & 0 \\ 0 & 2t & 4t-2 & 0 & 0 \end{bmatrix}.$$

Доказательство. Если раскраска отличается от раскраски линиями, то по следствию 3 во всех её столбцах (и во всех строках) содержится одинаковое число единиц. Отсюда следует, что единственным возможным остаётся только третий тип раскрасок.

Полупараметры инцидентности приведённых раскрасок вычисляются непосредственным образом. Утверждение 6 доказано.

Финансирование работы

Исследование выполнено за счёт Российского научного фонда (проект № 22–11–00266, rscf.ru/project/22-11-00266). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Дельсарт Ф.** Алгебраический подход к схемам отношений теории кодирования. М.: Мир, 1976. 134 с.
2. **Godsil C.** Compact graphs and equitable partitions // Linear Algebra Appl. 1997. V. 255, No. 1–3. P. 259–266. DOI: 10.1016/S0024-3795(97)83595-1.
3. **Визинг В. Г.** Дистрибутивная раскраска вершин графа // Дискрет. анализ и исслед. операций. 1995. Т. 2, № 4. С. 3–12.
4. **Пузынина С. А.** Периодичность совершенных раскрасок бесконечной прямоугольной решётки // Дискрет. анализ и исслед. операций. Сер. 1. 2004. Т. 11, № 1. С. 79–92.
5. **Krotov D. S.** Perfect colorings of the infinite square grid: Coverings and twin colors // Electron. J. Comb. 2023. V. 30, No. 2. Paper ID P2.4. 59 p. DOI: 10.37236/10005.
6. **Axenovich M. A.** On multiple coverings of the infinite rectangular grid with balls of constant radius // Discrete Math. 2003. V. 268, No. 1–3. P. 31–49. DOI: 10.1016/S0012-365X(02)00744-6.
7. **Августинович С. В., Могильных И. Ю.** Совершенные раскраски графов Джонсона $J(8,3)$ и $J(8,4)$ в два цвета // Дискрет. анализ и исслед. операций. 2010. Т. 17, № 2. С. 3–19.
8. **Хорошилова Д. Б.** О циркулярных совершенных раскрасках в два цвета // Дискрет. анализ и исслед. операций. 2009. Т. 16, № 1. С. 80–92.
9. Handbook of combinatorics. V. 1. Amsterdam: Elsevier, 1995. 2198 p.
10. **Потапов В. Н., Августинович С. В.** Комбинаторные дизайны, разностные множества и бент-функции как совершенные раскраски графов и мультиграфов // Сиб. мат. журн. 2020. Т. 61, № 5. С. 1087–1100.
11. **Taranenko A. A.** Perfect colorings of hypergraphs. Ithaca, NY: Cornell Univ., 2022. 20 p. (Cornell Univ. Libr. e-Print Archive; arXiv:2208.03447). DOI: 10.48550/arXiv.2208.03447.

12. Handbook of combinatorial designs. Boca Raton: Chapman & Hall/CRC, 2007. 1018 p.

Бородин Семён Олегович
Тараненко Анна Александровна

Статья поступила
4 сентября 2023 г.
После доработки —
7 февраля 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 54–78

UDC 519.179.1+519.174.7 DOI: 10.33048/daio.2024.31.784

PERFECT COLORINGS OF SUBMATRIX HYPERGRAPHS

S. O. Borodin[a] and *A. A. Taranenko*[b]

Sobolev Institute of Mathematics,
4 Koptyug Avenue, 4, 630090, Novosibirsk, Russia
E-mail: [a] s.borodin@g.nsu.ru, [b] taa@math.nsc.ru

Abstract. A submatrix hypergraph $G_{n\times m}$ is a hypergraph whose vertices are entries of an $n \times m$ matrix and hyperedges are submatrices of order 2. In this paper, we consider perfect colorings of submatrix hypergraphs and study their parameters. We provide several constructions of perfect colorings of $G_{n\times m}$ and prove that the incidence matrices of 2-designs are perfect colorings of the submatrix hypergraph. Moreover, we describe all perfect 2-colorings of hypergraphs $G_{2\times m}$ and $G_{3\times m}$. Illustr. 1, bibliogr. 12.

Keywords: hypergraph, symmetric 2-design, perfect coloring.

References

1. **P. Delsarte,** *An Algebraic Approach to the Association Schemes of Coding Theory* (N. V. Philips' Gloeilampenfabrieken, Eindhoven, 1973; Mir, Moscow, 1976 [Russian]).
2. **C. Godsil,** Compact graphs and equitable partitions, *Linear Algebra Appl.* **255** (1–3), 259–266 (1997), DOI: 10.1016/S0024-3795(97)83595-1.
3. **V. G. Vizing,** Distributive coloring of graph vertices, *Diskretn. Anal. Issled. Oper.* **2** (4), 3–12 (1995) [Russian].
4. **S. A. Puzynina,** Periodicity of perfect colorings of an infinite rectangular grid, *Diskretn. Anal. Issled. Oper., Ser. 1,* **11** (1), 79–92 (2004) [Russian].
5. **D. S. Krotov,** Perfect colorings of the infinite square grid: Coverings and twin colors, *Electron. J. Comb.* **30** (2), ID P2.4 (2023), DOI: 10.37236/10005.
6. **M. A. Axenovich,** On multiple coverings of the infinite rectangular grid with balls of constant radius, *Discrete Math.* **268** (1–3), 31–49 (2003), DOI: 10.1016/S0012-365X(02)00744-6.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 424–440 (2024), DOI: 10.1134/S1990478924030050.

7. **S. V. Avgustinovich** and **I. Yu. Mogilnykh,** Perfect colorings of the Johnson graphs $J(8,3)$ and $J(8,4)$ with two colors, *Diskretn. Anal. Issled. Oper.* **17** (2), 3–19 (2010) [Russian] [*J. Appl. Ind. Math.* **5** (1), 19–30 (2011)].
8. **D. B. Khoroshilova,** On two-colour perfect colourings of circular graphs, *Diskretn. Anal. Issled. Oper.* **16** (1), 80–92 (2009) [Russian].
9. *Handbook of Combinatorics,* Vol. 1 (Elsevier, Amsterdam, 1995).
10. **V. N. Potapov** and **S. V. Avgustinovich,** Combinatorial designs, difference sets, and bent functions as perfect colorings of graphs and multigraphs, *Sib. Mat. Zh.* **61** (5), 1087–1100 (2020) [Russian] [*Sib. Math. J.* **61** (5), 867–877 (2020)].
11. **A. A. Taranenko,** Perfect colorings of hypergraphs (Cornell Univ., Ithaca, NY, 2022) (Cornell Univ. Libr. e-Print Archive, arXiv:2208.03447), DOI: `10.48550/arXiv.2208.03447`.
12. *Handbook of Combinatorial Designs* (Chapman & Hall/CRC, Boca Raton, 2007).

Semyon O. Borodin
Anna A. Taranenko

Received September 4, 2023
Revised February 7, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 79–104

УДК 519.8+518.25 DOI: 10.33048/daio.2024.31.788

ИССЛЕДОВАНИЕ ПОРОГОВОЙ УСТОЙЧИВОСТИ ДВУХУРОВНЕВОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРОИЗВОДСТВА И ДИСКРИМИНАЦИОННОГО ЦЕНООБРАЗОВАНИЯ

М. Е. Водян[1, a], *А. А. Панин*[2, b], *А. В. Плясунов*[2, c]

[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] m.vodyan@g.nsu.ru,
[b] aapanin1988@gmail.com, [c] apljas@math.nsc.ru

Аннотация. Рассматривается задача пороговой устойчивости для двухуровневой задачи с медианным типом размещения предприятий и дискриминационным ценообразованием. При решении такой задачи необходимо найти радиус пороговой устойчивости и такое полудопустимое решение исходной двухуровневой задачи, для которого выручка лидера не меньше заранее заданного значения (порога) при любом отклонении бюджетов, не превышающем порогового радиуса устойчивости, и которое сохраняет свою полудопустимость. Таким образом, пороговый радиус устойчивости определяет предел возмущений бюджетов потребителей, при котором выполняются эти условия.

Разработаны два приближённых алгоритма решения задачи пороговой устойчивости на основе эвристики спуска с чередующимися окрестностями. Эти алгоритмы основываются на поиске хорошего приближённого размещения предприятий, а также на вычислении оптимального набора цен для найденного размещения предприятий. Алгоритмы отличаются способом сравнения различных размещений предприятий, что в конечном итоге приводит к различным оценкам радиуса пороговой устойчивости. Численный эксперимент показал эффективность выбранного подхода как с точки зрения времени работы алгоритмов, так и качества получаемых решений. Табл. 4, ил. 2, библиогр. 24.

Ключевые слова: двухуровневая оптимизация, задача пороговой устойчивости, радиус пороговой устойчивости, размещение предприятий, дискриминационное ценообразование, спуск с чередующимися окрестностями.

Введение

При решении прикладных оптимизационных задач часто необходимо выбрать такое оптимальное или просто допустимое решение, которое приемлемо не только для текущих исходных данных, но и остаётся приемлемым при изменении этих данных в достаточно широком диапазоне. Например, не все исходные данные задачи могут быть определены точно; на качество решения могут влиять ошибки округления на этапе численного решения задачи; если приходится часто решать NP-трудную задачу большой размерности, то естественно использовать ранее полученное оптимальное или приемлемое допустимое решение для той же задачи, но уже с возмущёнными исходными данными.

В зависимости от типа доступной информации такие проблемы исследуются в ряде направлений: стохастическое программирование, оптимизация на основе нечёткого представления данных, робастная оптимизация, постоптимальный анализ чувствительности и устойчивости решений задач линейного и целочисленного программирования [1–7]. В [1] содержатся основные работы до 2000 г., связанные с устойчивостью конечномерных задач, включая идею радиуса устойчивости, идущую от В. К. Леонтьева. В [7] демонстрируется важность исследования проблем, связанных с устойчивостью, для приложений на основе многопараметрического программирования.

С каждым из классических подходов к анализу надёжности решений при различных возмущениях исходных данных связаны определённые недостатки. В основе моделей стохастического программирования лежит информация о вероятностном распределении случайных параметров, которая на практике зачастую недоступна. Разработка моделей на основе нечётких данных — существенно более сложное занятие, чем классическое математическое моделирование. Качество получаемых моделей существенно зависит от качества используемых экспертных оценок. Узкое место робастной оптимизации в том, что её применение ориентируется на учёт худших сценариев. С вычислительной точки зрения это приводит к решению значительно более сложных оптимизационных задач, чем исходная постановка. Некоторые полиномиально разрешимые задачи становятся NP-трудными в робастной постановке. Результаты, полученные в [4], демонстрируют очень высокую сложность многоэтапных задач стохастического программирования, которые оказываются PSPACE-трудными. Таким образом, есть проблемы с получением информации,

которая требуется в том или ином подходе при анализе устойчивости задачи [7].

Относительно недавно в этой области возникло новое направление исследований под названием пороговая устойчивость, свободное от недостатков перечисленных выше подходов, которое использует преобразование оптимизационной задачи на основе понятия радиуса устойчивости [8–12]. В работах [13–16] представлены результаты исследований, связанные с понятием радиуса устойчивости. Это направление исследований в конечном итоге привело к формулировкам, характерным для задач анализа пороговой устойчивости. Однако, впервые и независимо в явном виде эти формулировки появились при исследовании двухкритериальных задач размещения предприятий [8, 9], когда робастный подход к решению задач дискретной оптимизации [2] был реализован на основе пороговой модели, введённой в [9]. В этой работе при решении двухкритериальных задач размещения применялся один из известных методов — метод изменения ограничений (the ε-constraint method). В этом методе выбирается одна из целевых функций, оптимум которой ищется на множестве допустимых решений с учётом дополнительного порогового ограничения, образованного второй целевой функцией, ограниченной параметром $\varepsilon \geqslant 0$. Таким образом, в задаче пороговой устойчивости для заданного набора входных данных задачи размещения на максимум вместо максимизации дохода в новой формулировке будем максимизировать область входных данных (выбранный критерий), близких к начальному набору данных задачи, для которой ищется решение, приводящее к доходу не меньше заданного порога (второй критерий).

Итак, теперь с каждой оптимизационной задачей можно связать задачу пороговой устойчивости, в которой ищется максимальное значение параметра (радиус устойчивости), ограничивающего нормы вариаций, возмущающих исходные данные исследуемой задачи, и допустимое решение, которое при любых вариациях исходных данных допустимо в базовой постановке и удовлетворяет пороговому ограничению.

В настоящей работе продолжается исследование пороговой устойчивости двухуровневых задач, начатое в [17, 18]. Задачи такого типа образуют новый класс двухуровневых задач, для которых не известны ни точные, ни приближённые методы решения. Исследование такого класса задач с алгоритмической точки зрения несомненно является важной теоретической проблемой, поскольку при этом разрабатываются новые точные и приближённые методы для решения оптимизационных задач. Описание современного состояния дел в области двухуровневой оптимизации можно найти в обзорах [19–21].

В этой статье впервые исследована пороговая устойчивость решения двухуровневой задачи с медианным типом размещения предприятий

и дискриминационным ценообразованием. Основным вкладом является подход к разработке алгоритмов решения задачи пороговой устойчивости на основе методов решения исходной задачи и её подзадач и два быстрых приближённых алгоритма, полученных с помощью данного подхода. Оба алгоритма объединяет общая идея генерации хорошего приближения к радиусу устойчивости. С помощью VND-эвристики находится в определённом смысле наилучшее размещение предприятий, которое и используется для построения приближения к радиусу устойчивости [21–23]. Алгоритмы отличаются характеристиками используемых окрестностей. В первом из алгоритмов используется окрестность, в которой текущее размещение предприятий лучше соседнего размещения, если доход, полученный в задаче ценообразования, связанного с текущим размещением, больше, чем в задаче ценообразования соседа. В другом алгоритме используется окрестность, в которой текущее размещение предприятий лучше соседнего размещения, если оценка радиуса пороговой устойчивости, полученного с помощью вектора цен, соответствующего текущему размещению, больше оценки радиуса пороговой устойчивости, полученной с помощью задачи ценообразования соседа.

В разд. 1 вводятся основные определения и формулируется задача пороговой устойчивости для двухуровневой задачи размещения производства и дискриминационного ценообразования в виде двухуровневой модели нелинейного программирования. Здесь же содержится эквивалентное представление задачи пороговой устойчивости в виде одноуровневой линейной задачи смешанного целочисленного программирования. В разд. 2 содержатся результаты о вычислительной сложности задачи пороговой устойчивости. Разд. 3 содержит описание вспомогательных алгоритмов, с помощью которых разрабатываются два приближённых алгоритма на основе VND-эвристики. Разд. 4 содержит результаты вычислительных экспериментов на исходных данных из библиотеки тестовых задач «Дискретные задачи размещения», а также на случайных входных данных. Предлагаемые алгоритмы сравниваются между собой и точным методом из библиотеки Gurobi. В заключении обсуждаются полученные результаты и направление дальнейших исследований.

1. Постановка задачи пороговой устойчивости

Прежде чем сформулировать определение пороговой устойчивости для двухуровневых задач, приведём определение устойчивости, восходящее к работам Леонтьева, Гордеева и ряда других исследователей. Имеется оптимизационная задача P (с критерием максимизации) и некоторый её вход X. Обозначим через $F^*(X)$ $(F(X))$ множество оптимальных решений (множество допустимых решений) задачи P для входа X.

Пусть $\Delta(\rho) = \{\delta \mid \|\delta\| \leqslant \rho\}$ ($\Delta^{=}(\rho) = \{\delta \mid \|\delta\| = \rho\}$) — множество вариаций входа X, где $\rho > 0$. Оптимальное решение $Y^* \in F^*(X)$ называется *устойчивым,* если непусто множество

$$\Gamma^P(X, Y^*) = \{\rho > 0 \mid Y^* \in F^*(X + \delta) \text{ для любого } \delta \in \Delta(\rho)\}.$$

Величина $\sup \Gamma^P(X, Y^*)$ называется *радиусом устойчивости* оптимального решения $Y^* \in F^*(X)$.

Понятие устойчивости, которое исследуется в данной работе, получается путём релаксации условия, что решение $Y^* \in F^*(X)$ остаётся оптимальным при изменении входа X, и замены его условием, что решение остаётся допустимым при изменении входа X и значение целевой функции на нём не меньше заданного порога V. Пусть $f_P(X, Y)$ — целевая функция задачи P с входом X, где Y — произвольное допустимое решение. Решение $Y \in F(X)$ называется *устойчивым относительно порога* V, если непусто множество

$$\Gamma^P(X, Y, V) = \{\rho \geqslant 0 \mid Y \in F(X + \delta), \\ f_P(X + \delta, Y) \geqslant V \text{ для любого } \delta \in \Delta(\rho)\}.$$

Величина $\rho(X, Y, V) = \sup \Gamma^P(X, Y, V)$ называется *радиусом устойчивости* допустимого решения $Y \in F(X)$ *относительно порога* V.

В общем случае задача пороговой устойчивости для входа X и порога V формулируется следующим образом.

Задача 1. *Необходимо найти радиус устойчивости* $\rho(X, \widetilde{Y}, V)$ *и допустимое решение* $\widetilde{Y}$ *исходной задачи, устойчивое относительно порога* V:

$$\rho(X, Y, V) \to \max_{Y \in F(X)}.$$

В двухуровневых задачах в определении как допустимого, так и оптимального решений часть переменных является оптимальным решением задачи нижнего уровня, поэтому в дополнение к задаче 1 сформулируем ещё одну постановку, которая учитывает структуру допустимых и оптимальных решений двухуровневых задач. Разделим переменные двухуровневой задачи на две группы (Y_l, Y_f), где Y_l — переменные верхнего уровня, Y_f — переменные нижнего уровня. Пусть $F_f^*(Y_l)$ — множество оптимальных решений задачи нижнего уровня, а $F(X)|_l = \{Y_l \mid \exists\, Y_f \in F_f^*(Y_l)\colon (Y_l, Y_f) \in F(X)\}$ — проекция множества допустимых решений $F(X)$ двухуровневой задачи P на пространство переменных верхнего уровня. Полудопустимое решение $Y_l \in F(X)|_l$ называется *устойчивым относительно порога* V, если непусто множество

$$\Gamma(X, Y_l, V) = \{\rho \geqslant 0 \mid \text{ для любого } \delta \in \Delta(\rho) \\ \exists\, Y_f(\delta) \in F_f^*(Y_l)\colon (Y_l, Y_f(\delta)) \in F(X + \delta),\ f(X + \delta, Y) \geqslant V\}.$$

Величина $\rho(X, Y_l, V) = \sup \Gamma(X, Y_l, V)$ называется *радиусом устойчивости относительно порога* V или *пороговым радиусом устойчивости полудопустимого решения* $Y_l \in F(X)|_l$. В определении радиуса устойчивости заменяем требование допустимости решения при вариации δ исходных данных X условием существования оптимального решения $Y_f(\delta)$ задачи нижнего уровня с параметрами Y_l такого, что пара $(Y_l, Y_f(\delta))$ является допустимым решением двухуровневой задачи, возмущённой вариацией δ, и выполняется соответствующее пороговое ограничение.

В общем случае задача пороговой устойчивости двухуровневой задачи для входа X и порога V формулируется следующим образом.

Задача 2. *Найти радиус устойчивости относительно входа* X *и вектор переменных верхнего уровня* Y_l, *устойчивый относительно порога* V:

$$\rho(X, Y_l, V) \to \max_{Y_l \in F(X)|_u}.$$

Приведём содержательную постановку базовой задачи с медианным типом размещения предприятий и дискриминационным ценообразованием, пороговая устойчивость которой исследуется далее. Сформулируем её в виде игры Штакельберга «лидер — последователи». В качестве лидера выступает производитель, который размещает r предприятий и формирует цены на каждом из них. В качестве последователей — потребители, каждый из которых выбирает то предприятие, на котором его суммарные затраты на покупку и транспортировку товара минимальны, и совершает покупку только в том случае, когда эти затраты не превышают его бюджета. Требуется выбрать такое размещение предприятий и такие цены, при которых доход производителя максимален. Далее рассматривается оптимистическая постановка двухуровневой задачи. Для этого необходимо ввести следующее соглашение. Если у потребителя есть несколько предприятий с одинаковой минимальной суммой платежей, то он выберет предприятие с минимальными транспортными затратами. В статье рассматривается дискриминационное ценообразование (discriminatory pricing), когда на каждом предприятии для каждого потребителя устанавливается своя цена.

Задача пороговой устойчивости отличается от базовой постановки тем, что заранее задан доход производителя, определяющий пороговое ограничение, и наличием неопределённости в бюджетах потребителей, максимизируя которые сможем получить максимально возможное отклонение от ожидаемых (данных) бюджетов.

Для того чтобы сформулировать математическую модель задачи пороговой устойчивости, введём следующие обозначения и переменные. Обозначения:

- $I = \{1, \ldots, n\}$ — множество возможных мест открытия предприятий;
- $J = \{1, \ldots, m\}$ — множество потребителей;
- $r \in Z^+$ — число размещаемых предприятий;
- $b_j \in \mathbb{Z}^+ \cup \{0\}$ — бюджет потребителя j;
- $c_{ij} \in \mathbb{Z}^+ \cup \{0\}$ — транспортные затраты потребителя j, если он обслуживается на предприятии i;
- $V \in \mathbb{Z}^+$ — доход производителя.

Переменные:

- $\rho \in \mathbb{Q}^+ \cup \{0\}$ — радиус пороговой устойчивости;
- $p_{ij} \in \mathbb{Q}^+ \cup \{0\}$ — цена товара на предприятии i для потребителя j;
- $x_{ij} = \begin{cases} 1, & \text{если потребитель } j \text{ обслуживается на предприятии } i, \\ 0 & \text{иначе;} \end{cases}$
- $y_i = \begin{cases} 1, & \text{если предприятие } i \text{ открыто,} \\ 0 & \text{иначе.} \end{cases}$

Двухуровневая смешанно целочисленная квадратичная математическая модель задачи пороговой устойчивости имеет вид

$$\rho \to \max_{p,y,x,\rho}, \tag{1}$$

$$\sum_{i\in I}\sum_{j\in J} p_{ij}x_{ij} \geqslant V, \tag{2}$$

$$\sum_{i\in I} y_i = r, \tag{3}$$

$$y_i \in \{0,1\}, \quad p_{ij}, \rho \in \mathbb{Q}^+ \cup \{0\}, \quad x \in \mathcal{F}^*(p,y,\rho), \quad i \in I,\ j \in J, \tag{4}$$

$\mathcal{F}^*(p,y,\rho)$ — множество оптимальных решений задачи нижнего уровня:

$$\sum_{i\in I}\sum_{j\in J}(b_j - c_{ij} - \rho - p_{ij})x_{ij} \to \max_x, \tag{5}$$

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j \in J, \tag{6}$$

$$x_{ij} \leqslant y_i, \quad i \in I,\ j \in J, \tag{7}$$

$$x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J. \tag{8}$$

Максимизируя целевую функцию (1) на верхнем уровне, получим максимально возможное отклонение от ожидаемых (данных) бюджетов. Ограничение (2) — пороговое ограничение, гарантирующее, что доход производителя не меньше заданного. Условие (3) требует, чтобы было открыто ровно r предприятий. Ограничения (4) определяют тип переменных верхнего уровня и фиксируют фундаментальное свойство двухуровневых задач: переменные нижнего уровня x являются его оптимальным

решением. Целевая функция нижнего уровня (5) описывает стратегию каждого потребителя — в максимальной степени экономить свой бюджет, а ограничения (6)–(8) гарантируют, что каждый потребитель обслуживается не более чем одним предприятием производителя, которое должно быть открыто. Также из этих ограничений и определения целевой функции следует, что покупка совершается в том случае, когда это позволяет бюджет потребителя. В базовой постановке отсутствует целевая функция (1), а максимизируется доход производителя $\sum\limits_{i\in I}\sum\limits_{j\in J} p_{ij}x_{ij}$.

Задачу (1)–(8) можно свести к одноуровневой задаче и линеаризовать, введя дополнительные переменные $z_{ij} = p_{ij}x_{ij}$, $r_{ij} = \rho x_{ij}$ и ограничения:

$$\rho \to \max_{x,p,y,\rho,z,r},$$

$$\sum_{i\in I}\sum_{j\in J} z_{ij} \geqslant V,$$

$$\sum_{i\in I} y_i = r,$$

$$x_{ij} \leqslant y_i, \quad i \in I,\ j \in J,$$

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j \in J,$$

$$\sum_{i\in I}((b_j - c_{ij})x_{ij} - r_{ij} - z_{ij}) \geqslant 0, \quad j \in J, \tag{9}$$

$$\sum_{i\in I}(c_{ij}x_{ij} + z_{ij}) \leqslant c_{kj} + p_{kj}, \quad k \in I,\ j \in J, \tag{10}$$

$$x_{ij}, y_i \in \{0,1\}, \quad \rho, p_{ij} \in \mathbb{Q}^+ \cup \{0\}, \quad i \in I,\ j \in J,$$

$$(1 - x_{ij})W + z_{ij} \geqslant p_{ij}, \tag{11}$$

$$(1 - x_{ij})W + p_{ij} \geqslant z_{ij}, \tag{12}$$

$$z_{ij} \leqslant x_{ij}W, \tag{13}$$

$$z_{ij} \geqslant 0, \tag{14}$$

$$(1 - x_{ij})W + r_{ij} \geqslant \rho, \tag{15}$$

$$(1 - x_{ij})W + \rho \geqslant r_{ij}, \tag{16}$$

$$r_{ij} \leqslant x_{ij}W, \tag{17}$$

$$r_{ij} \geqslant 0, \tag{18}$$

где W — положительная константа и $W \geqslant \max\limits_{i\in I,\, j\in J}\{b_j - c_{ij}\}$.

Условия (9) и (10) гарантируют, что любой потребитель обслуживается только в том случае, если ему позволяет бюджет и его суммарные затраты минимальны. Группа ограничений (11)–(14) гарантирует, что если

j-й потребитель обслуживается на i-м предприятии, то цена на продукт для него равна p_{ij}, а если он не обслуживается, то индикатором этого будет значение z_{ij}, равное нулю. Таким образом, если $x_{ij} = 1$, то $z_{ij} = p_{ij}$, а если $x_{ij} = 0$, то $z_{ij} = 0$. Аналогично для группы ограничений (15)–(18).

2. Вычислительная сложность задачи пороговой устойчивости (1)–(8)

Напомним обозначения, используемые в теории сложности для описания полиномиальной иерархии классов сложности [20]. Первые два основных класса задач распознавания (P и NP) определяются с помощью детерминированных и недетерминированных машин Тьюринга [20]. Класс P содержит задачи распознавания, решаемые за полиномиальное время на детерминированных машинах Тьюринга, а класс NP — это класс задач распознавания, решаемых за полиномиальное время на недетерминированных машинах Тьюринга. Третий основной класс co-NP состоит из задач распознавания, дополнения которых принадлежат NP. Эти классы образуют первый уровень полиномиальной иерархии. В [18] было показано, что базовая задача с медианным типом размещения предприятий и дискриминационным ценообразованием NP-трудна в сильном смысле.

Обозначим через D_ρ и D стандартную задачу распознавания задачи (1)–(8) и стандартную задачу распознавания базовой задачи соответственно.

Теорема 1. *Задача D_ρ* NP*-полна в сильном смысле.*

Доказательство. Покажем, что D_ρ принадлежит классу NP. Предположим, что для некоторого целого числа $\widehat{\rho}$ существует такое допустимое решение (ρ, y, p, x), что $\rho \geqslant \widehat{\rho}$. Можно считать, что $\rho = \widehat{\rho}$. Действительно, если $\rho > \widehat{\rho}$, то при уменьшении ρ до $\widehat{\rho}$ при фиксированных (y, p) будет изменяться только оптимальное решение x задачи нижнего уровня, так как могут появиться клиенты, бюджеты которых увеличатся, и они будут обслужены на открытых предприятиях. Доход лидера при этом только возрастёт, т. е. пороговое ограничение не будет нарушено. Учитывая, что при заданных y и $\widehat{\rho}$ базовая задача полиномиально разрешима, также можно считать, что вектор цен p оптимален. Таким образом, существование для некоторого целого числа $\widehat{\rho}$ такого допустимого решения (ρ, y, p, x), что $\rho \geqslant \widehat{\rho}$, эквивалентно существованию такого размещения предприятий y, при котором в базовой задаче с бюджетами $b_j - \widehat{\rho}$, $j \in J$, и множеством открытых предприятий $\{i \mid y_i = 1\}$ доход лидера больше заданного порога V. Теперь заметим, что для заданного

целого числа $\widehat{\rho}$ требуемый вектор y может быть найден за недетерминированное полиномиальное время, если в задаче D_ρ ответ «да». Отсюда следует, что задача D_ρ принадлежит классу NP.

Покажем, что задача D_ρ полна в NP. Этот результат следует из полиномиальной сводимости задачи D к задаче D_ρ. Действительно, в задаче D для заданного порога V надо найти допустимое решение (y, p, x), которое приносит лидеру доход, не меньший порога. В качестве исходных данных задачи D_ρ возьмём $\widehat{\rho} = 0$ и исходные данные задачи D. Из результатов, полученных в [3], следует, что задача D NP-полна в сильном смысле. Теорема 1 доказана.

Теорема 2. *Для задачи* (1)–(8) *не существует детерминированных полиномиальных приближённых алгоритмов с абсолютной или относительной оценкой уклонения от оптимального решения при условии, что* $\mathrm{P} \neq \mathrm{NP}$.

Доказательство. Предположим, что существует детерминированный приближённый полиномиальный алгоритм для задачи (1)–(8). Покажем, что тогда задача D полиномиально разрешима. Рассмотрим произвольный вход данной задачи с порогом V. Применим приближённый алгоритм к входу задачи (1)–(8), который получается из входа задачи D. Если в задаче D для порога V ответ «да», то алгоритм выдаст некоторое приближённое допустимое решение (ρ, y, p, x) задачи (1)–(8). Если $\rho > 0$, то, рассуждая как в доказательстве теоремы 1, получим её допустимое решение $(0, y, p, \widehat{x})$, которое является подтверждением, что в задаче D для текущего входа ответ «да». Таким образом, получен полиномиальный алгоритм для задачи D, что противоречит условию $\mathrm{P} \neq \mathrm{NP}$. Теорема 2 доказана.

3. Алгоритмы

Для нахождения оптимального размещения и радиуса устойчивости предлагаются два алгоритма, основанных на методе спуска с чередующимися окрестностями (метаэвристика VND), который выполняет некоторое количество итераций с разными окрестностями до тех пор, пока не будет получен локальный оптимум относительно всех используемых окрестностей. Алгоритмы различаются выбором целевой функции и критерием оптимизации при сравнении двух размещений предприятий. Далее приводится описание нескольких вспомогательных алгоритмов.

3.1. Вспомогательные алгоритмы. В этом пункте оптимальную цену для j-го потребителя будем обозначать через p_j. Она всегда достигается на предприятии с минимальными транспортными затратами, поэтому можем убрать индекс i из обозначения p_{ij}.

Для заданного размещения y оптимальную цену для каждого потребителя можно посчитать при помощи алгоритма PC (price calculation).

Алгоритм 1. PC(y)

Вход: размещение y.
Выход: вектор цен $p = (p_1, \ldots, p_m)$, где p_j — цена для потребителя j.

1: $I(y) = \{i \mid y_i = 1\}$; $j \leftarrow 1$;
2: **if** $j > m$ **then** STOP;
3: **if** $b_j > \min\limits_{i \in I(y)} \{c_{ij}\}$ **then** $p_j \leftarrow b_j - \min\limits_{i \in I(y)} \{c_{ij}\}$;
4: **else** $p_j \leftarrow 0$; $j \leftarrow j + 1$; **goto** 1.

Таким образом, для каждого потребителя находится то предприятие, на котором затраты на транспортировку минимальны, а если бюджет потребителя превосходит эти затраты, то цена устанавливается как разность между бюджетом потребителя и минимальными транспортными затратами, тем самым принося максимальный доход производителю. Если бюджет j-го потребителя меньше минимальных транспортных затрат, то ни на одном из открытых предприятий клиент не может быть обслужен, принося положительную прибыль; индикатором этого является значение переменной p_j, равное нулю. Данный алгоритм решает задачу ценообразования для заданного размещения за полиномиальное время. Временна́я сложность алгоритма $O(mr)$.

Следующий алгоритм PR (price recalculation) понадобится для пересчёта вектора цен при вычислении радиуса пороговой устойчивости.

Алгоритм 2. PR(p, Δ)

Вход: вектор цен $p = (p_1, \ldots, p_m)$, некоторое значение Δ.
Выход: вектор цен $\overline{p} = (\overline{p}_1, \ldots, \overline{p}_m)$.

1: $j \leftarrow 1$;
2: **if** $j > m$ **then** STOP;
3: **if** $p_j > \Delta$ **then** $\overline{p}_j \leftarrow p_j - \Delta$;
4: **else** $\overline{p}_j \leftarrow 0$; $j \leftarrow j + 1$; **goto** 2.

Данный алгоритм пересчитывает цену для каждого потребителя. Если можно уменьшить цену на заданную величину и прибыль от клиента останется положительной, то уменьшаем цену, иначе перестаём обслуживать клиента. Временна́я сложность алгоритма $O(m)$.

Радиус пороговой устойчивости для заданного размещения y можно найти при помощи алгоритма RC (radius calculation).

Алгоритм 3. RC(y)

Вход: размещение y.
Выход: радиус устойчивости ρ.

1: $\rho \leftarrow 0$; $p \leftarrow \mathrm{PC}(y)$;
2: **if** $\sum\limits_{j \in J} p_j < V$ **then** STOP;
3: $d(p) \leftarrow \sum\limits_{j \in J} p_j - V$;
4: **if** $d(p) = 0$ **then** STOP;
5: $c(p) = |\{j \in J \mid p_j \neq 0\}|$; $\rho \leftarrow \rho + d(p)/c(p)$; $p \leftarrow \mathrm{PR}(p, d(p)/c(p))$;
6: **goto** 3.

Если алгоритм остановился на шаге 2, т. е. максимальный доход производителя на размещении y меньше порогового ограничения (ожидаемого дохода), то на данном размещении невозможно найти допустимого значения радиуса пороговой устойчивости. На шаге 3 расчитываем сверхприбыль $d(p)$. Если она положительная, то можем увеличить радиус пороговой устойчивости. На шаге 5 вычисляем число обслуживаемых клиентов и увеличиваем радиус устойчивости на величину, полученную делением сверхприбыли на число всех обслуживаемых клиентов. Таким образом, если на данном размещении есть допустимое решение, то алгоритм останавливается только тогда, когда сверхприбыль $d(p)$ будет нулевая. Временна́я сложность алгоритма RC в худшем случае $O(mr + m^2)$.

Утверждение 1. *Если для размещения предприятий y оптимальный набор цен и оптимальное назначение потребителей удовлетворяют пороговому ограничению, то радиус устойчивости* RC(y) *максимальный для данного размещения y.*

Доказательство. Предположим, напротив, что существует допустимое решение (ρ, y, p) и $\rho > \overline{\rho} = \mathrm{RC}(y)$. Каждая компонента $\overline{p}_j$ вектора цен $\overline{p}$, полученного алгоритмом RC, является максимально возможной ценой обслуживания j-го потребителя при радиусе пороговой устойчивости $\overline{\rho}$. Из шага 1 алгоритма RC следует, что $\sum\limits_{j \in J} \overline{p}_j = V$. Так как $\rho > \overline{\rho}$, имеем $\overline{p}_j > p_j$. В результате получаем $V = \sum\limits_{j \in J} \overline{p}_j < \sum\limits_{j \in J} p_j$. Значит, решение (ρ, y, p, x) не допустимо; противоречие. Утверждение 1 доказано.

3.2. Критерии выбора. Используем два критерия для попарного сравнения мест расположения объектов. Первый критерий использует целевую функцию исходной задачи. Второй критерий при оценке двух вариантов размещения предприятий основывается на сравнении соответствующих радиусов пороговой устойчивости.

Пусть имеется два допустимых размещения y и $\widetilde{y}$ и соответствующие им векторы цен p и $\widetilde{p}$, полученные при помощи алгоритма PC.

Первый критерий. Если $\sum\limits_{j\in J} p_j > \sum\limits_{j\in J} \widetilde{p}_j$, то y и соответствующий вектор цен p лучше, чем $\widetilde{y}$ и $\widetilde{p}$.

Второй критерий. Пусть $\rho = \mathrm{RC}(y)$, $\widetilde{\rho} = \mathrm{RC}(\widetilde{y})$. Если $\rho > \widetilde{\rho}$, то y и соответствующий вектор цен p лучше, чем $\widetilde{y}$ и $\widetilde{p}$.

Следует отметить, что не всегда там, где доход больше, больше и радиус пороговой устойчивости. В доказательство этого рассмотрим

Пример 1. Пусть общее число предприятий, число клиентов и число открываемых предприятий равны n, $m > 2$ и r соответственно. Задано пороговое ограничение $V = k \in Z^+$, и есть два допустимых размещения y_f и y_s таких, что открытые объекты из y_f не входят в y_s. Пусть на первом размещении алгоритм PC возвращает вектор цен $p^f = \mathrm{PC}(y_f) = (k, k, \ldots, k)$, $\left|\left\{j \mid p_j^f \neq 0\right\}\right| = m$, а на втором — $p^s = \mathrm{PC}(y_s) = (k(m-1), 0, \ldots, 0)$, $\left|\left\{j \mid p_j^s \neq 0\right\}\right| = 1$. Тогда доход производителя на размещении y_f равен km, а радиус пороговой устойчивости равен $k\frac{m-1}{m} < k$. На размещении y_s доход производителя равен $k(m-1) < km$, а радиус пороговой устойчивости равен $k(m-2) \geqslant k$.

Утверждение 2. *Пусть (y^*, p^*, x^*) — оптимальное решение исходной задачи, $\rho^* = \mathrm{RC}(y^*)$, $\widetilde{p}^* = \mathrm{PR}(p^*, \rho^*)$. Решение $(\rho^*, y^*, \widetilde{p}^*)$ будет оптимальным в задаче пороговой устойчивости тогда и только тогда, когда $\sum\limits_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) \geqslant 0$ для любого допустимого решения (y, p, x) исходной задачи и $\widetilde{p} = \mathrm{PR}(p, \rho^*)$.*

Доказательство. Предположим, напротив, что $\sum\limits_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) < 0$ для некоторого решения (y, p, x) исходной задачи. Тогда

$$\sum_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) = \sum_{j=1}^{m} \widetilde{p}_j^* - \sum_{j=1}^{m} \widetilde{p}_j = V - \sum_{j=1}^{m} \widetilde{p}_j < 0,$$

откуда $\sum\limits_{j=1}^{m} \widetilde{p}_j - V > 0$. Следовательно, на шаге 2 алгоритма RC можем увеличить радиус пороговой устойчивости. Утверждение 2 доказано.

Максимальный радиус пороговой устойчивости будет меньше при неоптимальном наборе цен. Тогда имеет место

Следствие 1. *Пусть (y^*, p^*, x^*) — оптимальное решение исходной задачи, $\rho^* = \mathrm{RC}(y^*)$, $\widetilde{p}^* = \mathrm{PR}(p^*, \rho^*)$. Решение $(\rho^*, y^*, \widetilde{p}^*)$ будет оптимальным в задаче пороговой устойчивости тогда и только тогда, когда*

$\sum\limits_{j=1}^{m}(\widetilde{p}_j^* - \widetilde{p}_j) \geqslant 0$ *для любого допустимого размещения предприятий* y, $\widetilde{p} = \mathrm{PR}(p, \rho^*)$ *и набора цен* p, *оптимального в исходной задаче для размещения* y.

3.3. Основной алгоритм. В качестве основного алгоритма использована метаэвристика VND (variable neighborhood descent). Приведём описание окрестностей, по которым будет производиться спуск. Обозначим через $d(x, y)$ расстояние Хэмминга между векторами x и y, а через $wt(x)$ — вес Хэмминга вектора x. Определим окрестность k-Swap(y) как множество, содержащее все допустимые размещения такие, что относительно размещения y было закрыто k предприятий и открыто k новых: k-Swap$(y) = \{\widetilde{y} \mid wt(\widetilde{y}) = r,\ d(y, \widetilde{y}) = 2k\}$. Вес Хэмминга размещения $\widetilde{y} \in k$-Swap(y), равный r, гарантирует, что число открываемых предприятий будет равно r.

Определим процедуру встряски k-Shake(y): последовательно просматриваем окрестности 2-Swap(y), 3-Swap(y), ..., k-Swap(y), пока не будет найдено размещение лучше y. Если такое размещение найдено, то на выходе имеем новое размещение, лучшее y, иначе сохраняем размещение y.

Алгоритм 4. VND

Вход: $I_{\max}$ — максимальное число итераций алгоритма, k — параметр процедуры встряски k-Shake.

Выход: Наилучшее размещение y.

1: $I \leftarrow 0$; $y = \mathrm{rand}\{0,1\}^n$ — случайный булев вектор;
2: Применить локальный поиск относительно 1-Swap(y), y^* — локальный оптимум;
3: $I \leftarrow I + 1$; $y \leftarrow k$-Shake(y^*);
4: **if** $y = y^*$ или $I > I_{\max}$ **then** STOP;
5: **else goto** 1.

В результате работы алгоритма VND получим некоторое размещение y^*. При помощи алгоритма RC найдём радиус пороговой устойчивости для размещения y^*. В зависимости от критерия выбора получаются различные размещения предприятий на выходе. Следовательно, возникает два алгоритма, поведение которых исследуется в ходе численного эксперимента.

4. Численный эксперимент

Тестирование алгоритмов проведено на компьютере с процессором Intel(R) Core(TM) i7-8750H и 16 ГБ оперативной памяти. Алгоритм VND сравнивался с решателем Gurobi. Для сравнения использованы входные

данные из библиотеки «Discrete Location Problems» (табл. 1, 2) и входные данные, порождённые случайным образом с равномерным распределением (табл. 3, 5).

Если I — вход для исходной задачи, то $I \cup \{V\}$ — вход для задачи пороговой устойчивости с пороговым ограничением V (минимальным ожидаемым доходом). Для каждого примера решаем исходную задачу и определяем максимальный доход производителя. В качестве порогового ограничения V выбираем некоторую часть найденного максимального дохода, каждую из которых назовём уровнем со своим порядковым номером. Таким образом для одного набора входных данных исходной задачи имеем несколько наборов входных данных для задачи пороговой устойчивости, которые отличаются друг от друга пороговым ограничением.

Например, в табл. 1, 2 максимальный доход производителя разбит на 14 равных частей, и в качестве порогового ограничения V взяты 1, 4, 7, 10 и 13 частей, т. е. определено пять уровней. В табл. 1, 2 первый столбец означает номер примера, столбцы level, % — номер уровня и рассматриваемую часть максимального дохода производителя. Столбцы Gurobi, VND_1 и VND_2 содержат характеристики решений Gurobi и алгоритма VND с первым и вторым критерием лучшего размещения соответственно: time — время поиска решения; opt — оптимальное решение Gurobi; best, GAP — лучшее решение алгоритма VND и его относительное отклонение от оптимума, рассчитанное по формуле $\text{GAP} = \frac{\text{opt}-\text{best}}{\text{opt}}$. В столбце $\text{RC}(\overline{y})$ указан радиус пороговой устойчивости, полученный на оптимальном размещении $\overline{y}$ исходной задачи. В табл. 5 столбец dim содержит размеры входа n, m и r — число предприятий, число клиентов и число открываемых предприятий соответственно.

Таблица 1

Результаты численного эксперимента $n = 40$, $m = 100$, $r = 5$

№	V		Gurobi		VND_1			VND_2			$\text{RC}(\overline{y})$
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	7,12	8,13	64,05	0,02	64,05	0	0,02	64,05	0	63,33
	2	28,56	16,91	38,44	0,03	38,44	0	0,03	38,44	0	38,29
	3	49,97	31,85	24,58	0,03	24,55	0,0012	0,03	24,58	0	24,55
	4	71,41	20,23	13,08	0,03	13,08	0	0,03	13,08	0	13,08
	5	92,85	7,89	3,16	0,04	3,16	0	0,03	3,16	0	3,16
2	1	7,14	9,50	65,17	0,02	64,85	0,0049	0,02	65,17	0	64,0
	2	28,56	18,33	40,7	0,02	40,7	0	0,02	40,7	0	40,33
	3	50,00	36,23	26,1	0,04	26,1	0	0,03	26,1	0	25,71
	4	71,42	18,24	13,65	0,03	13,52	0,0095	0,03	13,65	0	13,52
	5	92,83	9,16	3,21	0,03	3,21	0	0,03	3,21	0	3,21
3	1	7,13	6,09	63,21	0,02	63,0	0,0033	0,02	63,21	0	62,69
	2	28,57	17,77	36,79	0,03	36,71	0,0021	0,03	36,79	0	36,31
	3	50,00	22,92	23,54	0,03	23,44	0,0042	0,03	23,54	0	23,44
	4	71,40	18,68	12,58	0,04	12,58	0	0,04	12,58	0	12,58
	5	92,84	9,18	3,0	0,04	3,0	0	0,03	3,0	0	3,0

4	1	7,12	5,70	61,43	0,03	60,65	0,0126	0,02	61,43	0	58,44
	2	28,55	12,69	36,64	0,03	36,57	0,0019	0,02	36,64	0	34,42
	3	49,98	15,43	22,04	0,03	22,04	0	0,02	22,04	0	20,98
	4	71,41	10,01	11,49	0,03	11,48	0,0008	0,03	11,48	0,0008	11,3
	5	92,84	6,68	2,68	0,03	2,63	0,0186	0,03	2,68	0	2,68
5	1	7,13	10,05	68,67	0,03	68,56	0,0016	0,03	68,67	0	64,41
	2	28,56	19,82	43,58	0,04	43,58	0	0,03	43,58	0	41,56
	3	49,99	41,12	27,25	0,03	26,89	0,0132	0,03	27,25	0	26,81
	4	71,42	30,46	14,51	0,03	14,51	0	0,03	14,51	0	14,45
	5	92,85	13,31	3,44	0,04	3,42	0,0058	0,05	3,44	0	3,42
6	1	7,12	6,33	64,47	0,02	63,47	0,0155	0,02	64,47	0	61,94
	2	28,56	17,34	40,11	0,04	40,11	0	0,03	40,11	0	38,76
	3	50,00	25,45	24,63	0,04	24,48	0,006090	0,03	24,63	0	24,48
	4	71,41	14,88	13,29	0,05	13,29	0	0,03	13,29	0	13,29
	5	92,84	7,99	3,2	0,06	3,2	0	0,04	3,2	0	3,2
7	1	7,12	8,98	65,58	0,02	65,58	0	0,02	65,58	0	63,16
	2	28,56	19,27	42,47	0,04	42,42	0,0011	0,03	42,47	0	41,69
	3	50,00	31,80	27,17	0,03	27,17	0	0,02	27,17	0	26,6
	4	71,41	19,72	13,95	0,03	13,95	0	0,03	13,95	0	13,85
	5	92,85	9,13	3,23	0,04	3,23	0	0,04	3,23	0	3,23
8	1	7,13	9,71	65,35	0,03	65,33	0,0003	0,03	65,35	0	62,45
	2	28,57	16,46	40,34	0,03	40,34	0	0,02	40,34	0	39,77
	3	50,00	41,41	25,11	0,05	25,11	0	0,03	25,11	0	25,0
	4	71,40	25,25	13,28	0,03	13,28	0	0,03	13,28	0	13,2
	5	92,84	11,26	3,26	0,03	3,19	0,0214	0,03	3,26	0	3,19
9	1	7,12	7,72	65,53	0,03	65,47	0,0009	0,02	65,53	0	62,58
	2	28,55	12,79	41,23	0,03	41,23	0	0,03	41,23	0	40,59
	3	49,98	40,21	25,45	0,03	25,45	0	0,03	25,45	0	25,35
	4	71,41	17,47	12,73	0,04	12,73	0	0,03	12,73	0	12,73
	5	92,85	7,63	2,96	0,06	2,96	0	0,06	2,96	0	2,96
10	1	7,14	8,76	66,56	0,03	65,76	0,0120	0,02	66,56	0	63,19
	2	28,55	13,82	41,83	0,03	41,83	0	0,02	41,83	0	39,93
	3	49,99	29,60	25,42	0,03	25,31	0,0043	0,03	25,42	0	25,0
	4	71,42	15,18	13,29	0,03	13,2	0,0067	0,03	13,29	0	13,2
	5	92,84	9,10	3,15	0,04	3,15	0	0,03	3,15	0	3,15

Результат, отражённый в табл. 1, получен на небольшой размерности $n = 40$. VND_1 находит оптимальное решение в 28 из 50 случаев, VND_2 выдаёт оптимум в 49 из 50 примеров. Время работы обоих алгоритмов примерно одинаково и в среднем более чем в 500 раз меньше времени работы решателя.

Увеличим число открываемых предприятий на 60 и посмотрим на следующий результат.

Таблица 2

Результаты численного эксперимента $n = 100$, $m = 100$, $r = 5$

№	V		Gurobi		VND_1			VND_2			$\mathrm{RC}(\overline{y})$
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	7,14	35,07	66,89	0,14	66,48	0,0061	0,14	66,89	0	63,19
	2	28,55	81,63	42,18	0,16	42,18	0	0,15	42,18	0	39,93
	3	49,99	143,16	25,51	0,27	25,51	0	0,16	25,51	0	25,0
	4	71,42	115,21	13,29	0,19	13,26	0,0022	0,31	13,29	0	13,2
	5	92,84	68,12	3,18	0,19	3,15	0,0063	0,17	3,18	0	3,15

2	1	7,14	37,77	64,24	0,13	63,19	0,0163	0,12	64,24	0	60,22
	2	28,55	129,49	39,11	0,16	39,02	0,0023	0,15	39,11	0	38,74
	3	49,99	275,46	24,55	0,19	24,31	0,0097	0,23	24,55	0	24,28
	4	71,42	109,75	13,12	0,19	13,0	0,0091	0,28	13,12	0	13,0
	5	92,84	226,93	3,11	0,20	3,11	0	0,21	3,06	0,0160	3,11
3	1	7,14	34,08	69,18	0,14	68,85	0,0047	0,12	69,18	0	64,21
	2	28,57	77,52	41,75	0,15	40,82	0,0222	0,14	41,75	0	40,1
	3	50,00	212,27	26,09	0,17	26,09	0	0,17	26,09	0	25,38
	4	71,43	80,97	14,13	0,18	13,48	0,0460	0,20	14,13	0	13,66
	5	92,86	66,34	3,19	0,19	3,13	0,0188	0,20	3,14	0,0156	3,13
4	1	7,13	36,20	66,79	0,14	65,76	0,0154	0,14	66,79	0	63,84
	2	28,57	88,10	41,55	0,17	40,4	0,0276	0,29	41,55	0	39,81
	3	49,99	220,34	25,63	0,17	25,25	0,0148	0,34	25,63	0	25,51
	4	71,43	111,33	13,45	0,34	13,45	0	0,18	13,45	0	13,37
	5	92,84	208,84	3,17	0,34	3,17	0	0,21	3,17	0	3,17
5	1	7,13	37,07	66,88	0,13	66,39	0,0073	0,21	66,88	0	62,11
	2	28,56	60,72	41,43	0,15	41,05	0,0091	0,15	41,43	0	39,37
	3	50,00	144,30	24,68	0,21	24,68	0	0,18	24,68	0	23,91
	4	71,41	155,15	12,93	0,21	12,93	0	0,20	12,91	0,0015	12,68
	5	92,84	73,71	3,09	0,21	3,08	0,0032	0,33	3,09	0	3,08
6	1	7,13	34,45	67,26	0,13	66,68	0,0086	0,12	67,26	0	66,05
	2	28,57	66,79	41,74	0,17	41,66	0,0019	0,20	41,74	0	41,63
	3	50,00	124,89	26,33	0,19	26,33	0	0,17	26,33	0	25,85
	4	71,40	97,23	14,2	0,19	14,2	0	0,22	14,2	0	13,87
	5	92,84	201,63	3,39	0,21	3,28	0,0324	0,30	3,39	0	3,28
7	1	7,14	36,30	59,57	0,14	59,57	0	0,12	59,57	0	53,25
	2	28,55	74,21	36,0	0,19	36,0	0	0,15	36,0	0	33,84
	3	50,00	143,13	22,31	0,39	22,25	0,0026	0,16	22,25	0,0026	21,62
	4	71,41	150,51	11,74	0,19	11,31	0,0366	0,18	11,74	0	11,54
	5	92,83	72,89	2,78	0,19	2,26	0,1870	0,19	2,78	0	2,77
8	1	7,12	36,14	65,12	0,14	64,26	0,0132	0,18	65,12	0	62,84
	2	28,55	91,18	38,73	0,15	38,62	0,0028	0,16	38,73	0	37,63
	3	49,98	217,38	23,19	0,20	23,13	0,0025	0,19	23,19	0	22,88
	4	71,41	170,15	11,92	0,21	11,84	0,0067	0,20	11,84	0,0067	11,85
	5	92,85	231,70	2,86	0,22	2,86	0	0,18	2,86	0	2,84
9	1	7,14	36,27	65,73	0,14	65,06	0,0101	0,20	65,73	0	62,84
	2	28,56	80,06	37,7	0,17	37,7	0	0,17	37,7	0	37,65
	3	49,98	167,22	23,17	0,27	23,17	0	0,24	23,17	0	23,07
	4	71,41	155,83	12,0	0,22	12,0	0	0,23	12,0	0	12,0
	5	92,83	51,49	2,76	0,21	2,76	0	0,21	2,76	0	2,76
10	1	7,12	39,72	65,33	0,14	65,1	0,0035	0,14	65,33	0	62,35
	2	28,55	111,89	40,79	0,16	40,71	0,0019	0,17	40,79	0	40,71
	3	50,00	255,10	25,81	0,18	25,81	0	0,21	25,81	0	25,81
	4	71,42	202,17	13,53	0,20	13,53	0	0,21	13,53	0	13,53
	5	92,85	752,71	3,22	0,21	3,22	0	0,19	3,22	0	3,22

Этот результат очень похож на предыдущий. Алгоритм VND_2 находит оптимум в 45 из 50 случаев, что на 25 превосходит число оптимумов, найденных алгоритмом VND_1. Время работы решателя Gurobi в среднем, более чем в 500 раз больше времени алгоритмов VND.

Диаграмма на рис. 1 отражает усреднённые значения результатов из табл. 1, 2 по каждому уровню разбиения. По горизонтали отмечены номера уровней, по вертикали — усреднённые значения относительного отклонения. Сплошная линия относится к алгоритму VND_1, штриховая — VND_2. Штрихпунктирная линия с двумя точками показывает целевую функцию на оптимальном размещении исходной задачи.

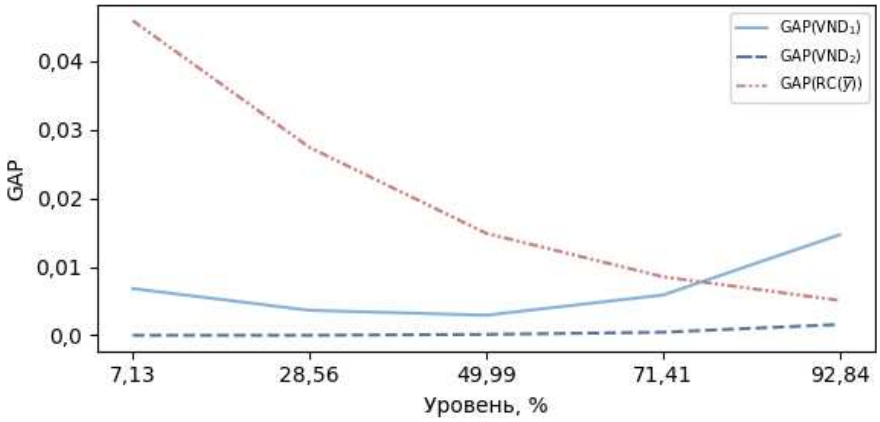


Рис. 1. Усреднённые значения по каждому уровню (табл. 1, 2)

Решения, найденные алгоритмами VND_1 и VND_2, близки к оптимальному. Значение целевой функции, полученное на оптимальном размещении исходной задачи, приближается к оптимальному при приближении порогового ограничения к максимальному доходу производителя.

Следующий результат получен на случайных входных данных с равномерным распределением.

Таблица 3

Результаты численного эксперимента на случайных входных данных

№	V		Gurobi		VND_1			VND_2			RC($\overline{y}$)
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	32,54	1,20	45,2	0,01	45,2	0	0,01	45,2	0	45,2
	2	65,90	1,08	22,85	0,01	22,85	0	0,01	22,85	0	22,85
	3	99,25	0,30	0,5	0,01	0,5	0	0,01	0,5	0	0,5
2	1	32,66	1,18	52,05	0,00	52,05	0	0,00	52,05	0	52,05
	2	65,98	0,62	26,3	0,00	26,3	0	0,00	26,3	0	26,3
	3	99,35	0,26	0,5	0,00	0,5	0	0,00	0,5	0	0,5
3	1	33,01	25,00	53,98	0,09	53,95	0,0005	0,07	53,95	0,0005	53,98
	2	66,34	23,33	27,13	0,08	27,1	0,0011	0,07	27,1	0,0011	27,12
	3	99,69	4,00	0,25	0,10	0,23	0,0800	0,07	0,0	1	0,25
4	1	32,93	18,14	44,65	0,16	44,65	0	0,15	44,65	0	44,65
	2	66,28	15,46	22,45	0,16	22,45	0	0,15	22,45	0	22,45
	3	99,62	2,51	0,25	0,18	0,25	0	0,16	0,25	0	0,25
5	1	33,13	140,46	56,98	0,26	56,98	0	0,39	56,95	0,0005	56,98
	2	66,46	184,77	28,58	0,27	28,58	0	0,39	28,55	0,0010	28,58
	3	99,80	58,76	0,17	0,26	0,17	0	0,20	0,0	1	0,17
6	1	33,11	123,96	53,43	0,58	53,42	0,0001	0,51	53,4	0,0005	53,43
	2	66,45	117,26	26,8	0,56	26,78	0,0007	0,54	26,77	0,0011	26,8
	3	99,79	18,99	0,17	0,61	0,15	0,1176	0,54	0,0	1	0,17
7	1	33,18	17801,16	48,13	1,45	48,13	0	0,76	48,08	0,0010	48,13
	2	66,51	23961,22	24,12	1,56	24,12	0	0,82	24,07	0,0020	24,12
8	1	33,20	168378,12	53,38	1,93	53,38	0	1,8	53,38	0	53,20

В табл. 4 указаны размеры входа для примеров из табл. 3.

Таблица 4

Размеры входа для табл. 3

№	n, m, r	№	n, m, r	№	n, m, r	№	n, m, r
1	$20, 20, 10$	3	$40, 40, 10$	5	$60, 60, 10$	7	$90, 90, 10$
2	$20, 20, 15$	4	$40, 40, 15$	6	$60, 60, 15$	8	$100, 100, 10$

На примерах в табл. 3 алгоритм VND_1 находит оптимум чаще, чем алгоритм VND_2. Значение целевой функции, полученное на оптимальном решении исходной задачи, в 19 случаях из 21 оказывается оптимальным. В примере 7 на первом и втором уровнях решатель находит оптимум за 5–6 часов, в то время как алгоритму VND требуется менее двух секунд. В примере 8 решатель потратил на поиск оптимального решения около двух суток, а алгоритм VND — около двух секунд.

На рис. 2 представлены усреднённые значения результатов из табл. 3 по каждому уровню разбиения.

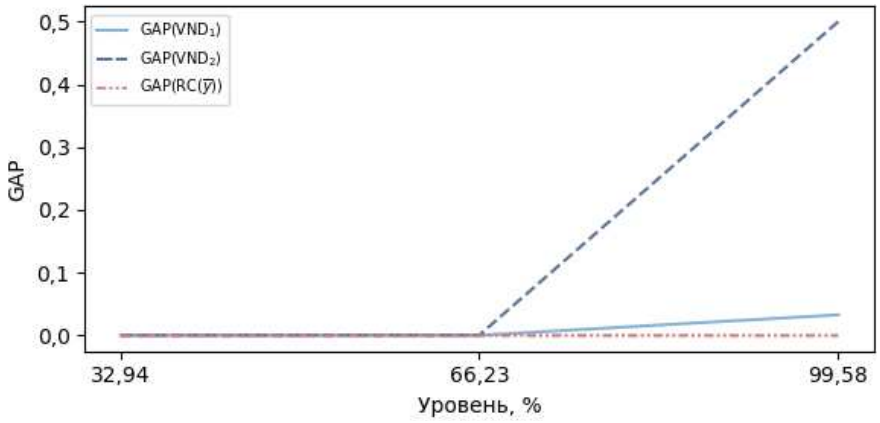


Рис. 2. Усреднённые значения по каждому уровню (табл. 3)

Таким образом, на небольшой с точки зрения времени вычисления размерности при наличии оптимального размещения $\overline{y}$ исходной задачи, имеет смысл применить алгоритм $\mathrm{RC}(\overline{y})$ для получения нижней оценки радиуса устойчивости.

Следующий эксперимент проводился на случайных данных большой с точки зрения времени вычисления размерности. Каждый пример разбит на два уровня. В столбце time для Gurobi указан прочерк, если решателю не удалось найти оптимального решения за три часа работы.

Лучшее найденное решение отмечено в столбце best. Синим цветом отмечено лучшее из решений решателя и алгоритмов, если оптимальное неизвестно.

Таблица 5

Результаты численного эксперимента на случайных входных данных

№	dim	V		Gurobi		VND_1		VND_2	
	n, m, r	level	%	time	best	time	best	time	best
1	100, 100, 10	1	33,32	—	53,26	4,59	53,28	4,33	53,28
	100, 100, 10	2	66,66	8616	26,63	3,42	26,63	3,05	26,63
2	100, 100, 15	1	33,33	3531	46,89	8,57	46,8	8,07	46,79
	100, 100, 15	2	66,66	2995	23,45	6,36	23,36	5,56	23,35
3	200, 200, 10	1	33,33	—	43,91	20,49	44,22	20,86	44,27
	200, 200, 10	2	66,67	—	22,69	25,98	23,3	23,71	23,35
4	200, 200, 15	1	33,33	—	50,55	154,51	51,65	56,69	51,54
	200, 200, 15	2	66,67	—	26,24	105,94	26,98	40,09	26,87
5	300, 300, 10	1	33,33	—	57,12	53,34	57,68	62,43	57,62
	300, 300, 10	2	66,66	—	27,69	76,90	30,24	83,31	30,19
6	300, 300, 15	1	33,33	—	31,70	274,46	52,14	195,93	52,16
	300, 300, 15	2	66,67	—	26,51	151,43	27,07	186,05	27,09
7	400, 400, 10	1	33,33	—	47,30	240,62	55,7	111,34	55,65
	400, 400, 10	2	66,67	—	21,20	443,30	29,61	210,62	29,56
8	400, 400, 15	1	33,33	—	43,33	611,13	49,58	553,10	49,58
	400, 400, 15	2	66,67	—	0,13	1009,20	25,9	910,73	25,9

В этих примерах решения, найденные алгоритмами VND_1 и VND_2, близки друг к другу. В большинстве случаев оба алгоритма находят решение лучше и за гораздо меньшее относительно решателя Gurobi время.

Заключение

Изучение пороговой устойчивости комбинаторных оптимизационных задач является новым направлением исследований в современной теории экстремальных задач. В ходе таких исследований возникают новые классы задач со своими структурными особенностями. В связи с этим возникает необходимость разработки точных и приближённых алгоритмов их решения. Основная доля работ в этой области связана с исследованием пороговой устойчивости одноуровневых оптимизационных задач [8–12]. В работах [8, 9] начато исследование пороговой устойчивости бикритериальных задач размещения объектов. Исследование пороговой

устойчивости двух- и трёхуровневых задач размещения объектов и ценообразования начато в [17, 18].

В настоящей работе предлагается подход к разработке алгоритмов решения задач пороговой устойчивости, основанный на методах решения исходной задачи и её подзадач. В статье показано, что используя решение исходной задачи, можно эффективно найти решение задачи пороговой устойчивости для двухуровневой задачи размещения производства и дискриминационного ценообразования при помощи предложенного полиномиального алгоритма поиска оптимальной цены и гибридной эвристики, которая основана на эвристике VND и локальном поиске. Приводится экспериментальное сравнение двух версий алгоритма и решателя Gurobi. Аналогичные результаты содержатся в работе [24], в которой показана эффективность предлагаемого подхода при разработке приближённых алгоритмов решения для двухуровневой задачи с медианным типом размещения предприятий и равномерным ценообразованием как с точки зрения времени работы алгоритмов, так и качества получаемых решений. Однако наибольший интерес представляет исследование пороговой устойчивости двухуровневых задач размещения предприятий с фабричным ценообразованием, поскольку в этом случае задача ценообразования NP-трудна для фиксированного размещения объектов. По этой причине важно оценить, насколько эффективными окажутся приближённые алгоритмы, разработанные на основе предложенного подхода для данной постановки.

Теоремы 1 и 2 приводят к следующей гипотезе: возможно, задачи пороговой устойчивости, исследованные в данной работе, полны в классе NPO относительно подходящей сводимости, сохраняющей аппроксимируемость.

Финансирование работы

Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 23–21–00424).

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Greenberg H. J.** An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization // Advances in computational and stochastic optimization, logic programming, and heuristic search. New York: Springer, 1998. P. 97–147. DOI: 10.1007/978-1-4757-2807-1_4.

2. **Ben-Tal A., Nemirovski A.** Robust optimization: Methodology and applications // Math. Program. 2002. V. 92. P. 453–480.
3. **Snyder L. V.** Facility location under uncertainty: A review // IIE Trans. 2006. V. 38. P. 537–554. DOI: 10.1080/07408170500216480.
4. **Dyer M., Stougie L.** Computational complexity of stochastic programming problems // Math. Program. Ser. A. 2006. V. 106. P. 423–432. DOI: 10.1007/s10107-005-0597-0.
5. **Кибзун А. И., Кан Ю. С.** Задачи стохастического программирования с вероятностными критериями. М.: Физматлит, 2009. 371 с.
6. **Correia I., da Gama F. S.** Facility location under uncertainty // Location science. Cham: Springer, 2015. P. 177–203.
7. **Charitopoulos V. M., Papageorgiou L. G., Dua V.** Multiparametric mixed integer linear programming under global uncertainty // Comput. Chem. Eng. 2018. V. 116. P. 279–295.
8. **Carrizosa E., Nickel S.** Robust facility location // Math. Methods Oper. Res. 2003. V. 58. P. 331–349.
9. **Carrizosa E., Ushakov A., Vasilyev I.** Threshold robustness in discrete facility location problems: A bi-objective approach // Optim. Lett. 2015. V. 9. P. 1297–1314.
10. **Rossi A., Gurevsky E., Battaïa O., Dolgui A.** Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations // Discrete Appl. Math. 2016. V. 208. P. 123–136.
11. **Pirogov A., Gurevsky E., Rossi A., Dolgui A.** Robust balancing of transfer lines with blocks of uncertain parallel tasks under fixed cycle time and space restrictions // Eur. J. Oper. Res. 2021. V. 290. P. 946–955.
12. **Sotskov Yu. N.** Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives // Algorithms. 2023. V. 16, No. 2. Paper ID 100. 43 p.
13. **Леонтьев В. К.** Устойчивость задачи коммивояжера // Вычисл. математика и мат. физика. 1975. Т. 15, № 5. С. 1298–1309.
14. **Леонтьев В. К., Гордеев Э. Н.** Качественное исследование траекторных задач // Кибернетика. 1986. № 5. С. 82–89.
15. **Sotskov Yu. N., Leontiev V. K., Gordeev Eh. N.** Some concepts of stability analysis in combinatorial optimization // Discrete Appl. Math. 1995. V. 58, No. 2. P. 169–190.
16. **Кузьмин К. Г.** Единый подход к нахождению радиусов устойчивости в многокритериальной задаче о максимальном разрезе графа // Дискрет. анализ и исслед. операций. 2015. Т. 22, № 5. С. 30–51.
17. **Panin A. A., Plyasunov A. V.** Stability analysis for pricing // Mathematical optimization theory and operations research. Rev. Sel. Pap. 19th Int. Conf. (Novosibirsk, Russia, July 6–10, 2020). Cham: Springer, 2020. P. 57–69. (Commun. Comput. Inf. Sci.; V. 1275). DOI: 10.1007/978-3-030-58657-7_7.
18. **Panin A. A., Plyasunov A. V.** The multilevel facility location and pricing problems: the computational complexity and the stability analysis // Optim. Lett. 2023. V. 17. P. 1295–1315.

19. Dempe S., Zemkoho A. Bilevel optimization. Advances and next challenges. Cham: Springer, 2020. 672 p. (Springer Optim. Its Appl.; V. 161). DOI: 10.1007/978-3-030-52119-6.

20. Kochetov Yu. A., Plyasunov A. V., Panin A. A. Bilevel discrete optimisation: Computational complexity and applications // The Palgrave handbook of operations research. Cham: Palgrave Macmillan, 2022. P. 3–42. DOI: 10.1007/978-3-030-96935-6_1.

21. Talbi E.-G. Metaheuristics: From design to implementation. Berlin: Wiley, 2009. 624 p.

22. Mladenovic N., Hansen P. Variable neighbourhood search // Comput. Oper. Res. 1997. V. 24. P. 1097–1100.

23. Кочетов Ю. А., Панин А. А., Плясунов А. В. Сравнение метаэвристик для решения двухуровневой задачи размещения предприятий и фабричного ценообразования // Дискрет. анализ и исслед. операций. 2015. Т. 22, № 3. С. 36–54.

24. Vodyan M. E., Panin A. A., Plyasunov A. V. Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem // 2023 19th Int. Asian School-Seminar Optimization Problems of Complex Systems (Novosibirsk, Russia, Aug. 14–22, 2023). Piscataway: IEEE, 2023. P. 130–135. DOI: 10.1109/OPCS59592.2023.10275325.

Водян Максим Евгеньевич
Панин Артём Александрович
Плясунов Александр Владимирович

Статья поступила
10 ноября 2023 г.
После доработки —
23 января 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 79–104

UDC 519.8+518.25 DOI: 10.33048/daio.2024.31.788

A STUDY OF THE THRESHOLD STABILITY OF THE BILEVEL PROBLEM OF FACILITY LOCATION AND DISCRIMINATORY PRICING

M. E. Vodyan[1, a], *A. A. Panin*[2, b], and *A. V. Plyasunov*[2, c]

[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] m.vodyan@g.nsu.ru,
[b] aapanin1988@gmail.com, [c] apljas@math.nsc.ru

Abstract. The problem of threshold stability for a bilevel problem with a median type of facility location and discriminatory pricing is considered. When solving such a problem, it is necessary to find the threshold stability radius and a semifeasible solution of the original bilevel problem such that the leader's revenue is not less than a predetermined value (threshold) for any deviation of budgets that does not exceed the threshold stability radius and which preserves its semifeasibility. Thus, the threshold stability radius determines the limit of disturbances of consumer budgets with which these conditions are satisfied.

Two approximate algorithms for solving the threshold stability problem based on the heuristic of descent with alternating neighborhoods are developed. These algorithms are based on finding a good approximate location of facilities as well as on calculating the optimal set of prices for the found location of facilities. The algorithms differ in the way they compare various locations of facilities; this ultimately leads to different estimates of threshold stability radius. A numerical experiment has shown the efficiency of the chosen approach both in terms of the running time of the algorithms and the quality of the solutions obtained. Tab. 4, illustr. 2, bibliogr. 24.

Keywords: bilevel optimization, threshold stability problem, threshold stability radius, facility location, discriminatory pricing, variable neighborhood descent.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 558–574 (2024), DOI: 10.1134/S1990478924030165.

References

1. **H. J. Greenberg,** An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization, in *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search* (Springer, New York, 1998), pp. 97–147, DOI: 10.1007/978-1-4757-2807-1_4.
2. **A. Ben-Tal** and **A. Nemirovski,** Robust optimization: Methodology and applications, *Math. Program.* **92**, 453–480 (2002).
3. **L. V. Snyder,** Facility location under uncertainty: A review, *IIE Trans.* **38**, 537–554 (2006), DOI: 10.1080/07408170500216480.
4. **M. Dyer** and **L. Stougie,** Computational complexity of stochastic programming problems, *Math. Program., Ser. A,* **106**, 423–432 (2006), DOI: 10.1007/s10107-005-0597-0.
5. **A. I. Kibzun** and **Yu. S. Kan,** *Stochastic Programming Problems with Probabilistic Criteria* (Fizmatlit, Moscow, 2009) [Russian].
6. **I. Correia** and **F. S. da Gama,** Facility location under uncertainty, in *Location Science* (Springer, Cham, 2015), pp. 177–203.
7. **V. M. Charitopoulos, L. G. Papageorgiou,** and **V. Dua,** Multiparametric mixed integer linear programming under global uncertainty, *Comput. Chem. Eng.* **116**, 279–295 (2018).
8. **E. Carrizosa** and **S. Nickel,** Robust facility location, *Math. Methods Oper. Res.* **58**, 331–349 (2003).
9. **E. Carrizosa, A. Ushakov,** and **I. Vasilyev,** Threshold robustness in discrete facility location problems: A bi-objective approach, *Optim. Lett.* **9**, 1297–1314 (2015).
10. **A. Rossi, E. Gurevsky, O. Battaïa,** and **A. Dolgui,** Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations, *Discrete Appl. Math.* **208**, 123–136 (2016).
11. **A. Pirogov, E. Gurevsky, A. Rossi,** and **A. Dolgui,** Robust balancing of transfer lines with blocks of uncertain parallel tasks under fixed cycle time and space restrictions, *Eur. J. Oper. Res.* **290**, 946–955 (2021).
12. **Yu. N. Sotskov,** Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives, *Algorithms* **16** (2), ID 100 (2023).
13. **V. K. Leontiev,** Stability of the travelling salesman problem, *Vychisl. Mat. Mat. Fiz.* **15** (5), 1298–1309 (1975) [Russian] [*USSR Comput. Math. Math. Phys.* **15** (5), 199–213 (1975)].
14. **V. K. Leontiev** and **Eh. N. Gordeev,** Qualitative investigation of path problems, *Kibern.,* No. 5, 82–89 (1986) [Russian] [*Cybern.* **22**, 636–646 (1986)].
15. **Yu. N. Sotskov, V. K. Leontiev,** and **Eh. N. Gordeev,** Some concepts of stability analysis in combinatorial optimization, *Discrete Appl. Math.* **58** (2), 169–190 (1995).
16. **K. G. Kuz'min,** A general approach to the calculation of stability radii for the max-cut problem with multiple criteria, *Diskretn. Anal. Issled. Oper.* **22** (5), 30–51 (2015) [Russian] [*J. Appl. Ind. Math.* **9** (4), 527–539 (2015)].

17. A. A. Panin and **A. V. Plyasunov,** Stability analysis for pricing, in *Mathematical Optimization Theory and Operations Research* (Rev. Sel. Pap. 19th Int. Conf., Novosibirsk, Russia, July 6–10, 2020) (Springer, Cham, 2020), pp. 57–69 (Commun. Comput. Inf. Sci., Vol. 1275), DOI: 10.1007/978-3-030-58657-7_7.

18. A. A. Panin and **A. V. Plyasunov,** The multilevel facility location and pricing problems: the computational complexity and the stability analysis, *Optim. Lett.* **17**, 1295–1315 (2023).

19. S. Dempe and **A. Zemkoho,** *Bilevel Optimization. Advances and Next Challenges* (Springer, Cham, 2020) (Springer Optim. Its Appl., Vol. 161), DOI: 10.1007/978-3-030-52119-6.

20. Yu. A. Kochetov, A. V. Plyasunov, and **A. A. Panin,** Bilevel discrete optimisation: Computational complexity and applications, in *The Palgrave Handbook of Operations Research* (Palgrave Macmillan, Cham, 2022), pp. 3–42, DOI: 10.1007/978-3-030-96935-6_1.

21. E.-G. Talbi, *Metaheuristics: From Design to Implementation* (Wiley, Berlin, 2009).

22. N. Mladenovic and **P. Hansen,** Variable neighbourhood search, *Comput. Oper. Res.* **24**, 1097–1100 (1997).

23. Yu. A. Kochetov, A. A. Panin, and **A. V. Plyasunov,** Comparison of metaheuristics for the bilevel facility location and mill pricing problem, *Diskretn. Anal. Issled. Oper.* **22** (3), 36–54 (2015) [Russian] [*J. Appl. Ind. Math.* **9** (3), 392–401 (2015)].

24. M. E. Vodyan, A. A. Panin, and **A. V. Plyasunov,** Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem, in *2023 19th Int. Asian School-Seminar Optimization Problems of Complex Systems, Novosibirsk, Russia, Aug. 14–22, 2023* (IEEE, Piscataway, 2023), pp. 130–135, DOI: 10.1109/OPCS59592.2023.10275325.

Maksim E. Vodyan
Artyom A. Panin
Aleksandr V. Plyasunov

Received November 10, 2023
Revised January 23, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 105–122

УДК 519.8+518.25 DOI: 10.33048/daio.2024.31.750

ЗАДАЧА ОДНОГО СТАНКА С РАВНЫМИ ДЛИТЕЛЬНОСТЯМИ РАБОТ И ВОЗМОЖНОСТЬЮ ПРЕРЫВАНИЙ

К. А. Ляшкова[a], *В. В. Сервах*[b]

Омский филиал Института математики им. С. Л. Соболева,
ул. Певцова, 13, 644099 Омск, Россия
E-mail: [a] ksech@bk.ru, [b] svv_usa@rambler.ru

Аннотация. Рассматривается задача минимизации среднего взвешенного времени для выполнения работ одинаковой длительности на одном станке при заданных временах поступления работ и возможности их прерывания. В настоящее время вычислительная сложность этой задачи неизвестна. В работе предложен алгоритм предобработки входных данных, что позволяет свести задачу к более узкому и регулярному классу примеров. Обоснованы свойства оптимальных решений, на основе которых разработан алгоритм построения конечного подмножества решений, содержащего оптимальное расписание. Описан подход к проведению параметрического анализа расписаний из этого подмножества, который позволяет сформировать подкласс расписаний, оптимальных при некоторых значениях весов. Выделен полиномиально разрешимый случай задачи. Табл. 1, ил. 10, библиогр. 16.

Ключевые слова: теория расписаний, один станок, прерывание.

1. Постановка задачи

На единственном станке необходимо выполнить n работ. Заданы моменты поступления работ в систему r_i, длительности их выполнения p_i и веса ω_i, $i = 1, 2, \ldots, n$. Станок в каждый момент времени может выполнять только одну работу. Требуется найти расписание выполнения работ, при котором взвешенная сумма моментов их завершения будет наименьшей. Время завершения работы i будем обозначать через C_i. Тогда целевая функция выглядит следующим образом: $\sum\limits_{i=1}^{n} \omega_i C_i$. В общепринятых обозначениях, например [1], эта задача записывается как $1|r_i|\sum \omega_i C_i$.

Задача NP-трудна в сильном смысле даже в случае единичных весов [2]. В [3] доказана полиномиальная разрешимость задачи $1|r_i, p_i =$

$p|\sum \omega_i C_i$ с одинаковыми длительностями работ. Если допустить прерывания, то задача $1|r_i, pmtn|\sum \omega_i C_i$ остаётся сильно NP-трудной [4]. Для единичных весов задача $1|r_i, pmtn|\sum C_i$ полиномиально разрешима [5]. Вопрос о вычислительной сложности задачи с прерываниями и равными длительностями работ $1|r_i, p_i = p, pmtn|\sum \omega_i C_i$ остаётся открытым.

Подробное исследование задач теории расписаний с возможностью прерывания работ проведено в [6]. В [7, 8] описаны некоторые важные свойства и, в частности, тот факт, что для поиска оптимума достаточно рассматривать расписания, в которых работа прерывается только в момент поступления работы с бóльшим весом. В [9] предложены релаксационные модели и градиентные алгоритмы с оценками точности получаемого решения для задачи $1|r_i|\sum \omega_i C_i$. Отметим также работы [10, 11], посвящённые исследованию модели линейного целочисленного программирования указанной задачи. В [12, 13] описан параметрический подход к исследованию комбинаторной структуры возможных оптимальных решений $1|r_i, p_i = p, pmtn|\sum \omega_i C_i$. Некоторые свойства оптимальных решений, полученные на основе параметрического анализа, изложены в [14]. В [15, 16] рассматривается задача со штрафами за нарушения директивных сроков выполнения работ и возможностью сверхурочных работ на станке.

В разд. 2 предложен алгоритм предобработки входных данных, который позволяет свести задачу к решению примеров более простой и регулярной структуры. В разд. 3 обоснованы некоторые свойства, на основе которых может быть построено конечное подмножество решений, включающее оптимальное расписание. Изложен алгоритм его формирования. В разд. 4 предлагается подход к параметрическому анализу расписаний этого подмножества. В результате такого анализа выделяется подкласс расписаний, которые являются оптимальными при некоторых значениях весов. В разд. 5 описан полиномиально разрешимый случай задачи.

2. Предобработка входных данных

В данном разделе задачу с прерываниями $1|r_i, p_i = p, pmtn|\sum \omega_i C_i$ сведём к совокупности подзадач более простой и регулярной структуры. Через S_i обозначим время начала выполнения работы $i = 1, 2, \ldots, n$. Прежде всего отметим свойство, связанное с порядком выполнения работ.

Утверждение 1. *Если $\omega_i < \omega_j$ и $r_i \geqslant r_j$, то для поиска оптимального решения достаточно рассматривать расписания, в которых работа j целиком выполняется раньше работы i, т. е. $C_j \leqslant S_i$.*

Доказательство. Рассмотрим произвольное расписание и выделим в нём совокупность всех интервалов, в течение которых выполняются

работы i и j. Суммарная длина этих интервалов $2p$. Составим новое расписание, выполняя в этих временны́х интервалах сначала работу j, а потом i. Остальные работы такая перестановка не затронет, и целевая функция при этом не увеличится. Тем самым получим расписание, для которого утверждение справедливо. Утверждение 1 доказано.

Значит, при $\omega_i < \omega_j$ и $r_i \geqslant r_j$ порядок выполнения работ i и j известен. Проблема выбора возникает при условии $r_i < r_j$ и $\omega_i < \omega_j$, когда работа с бóльшим весом поступает позднее. Такие работы назовём *конкурентными*. Заметим, что оба неравенства строгие, так как если выполняется хотя бы одно равенство, то работы не будут конкурентными. Если $r_i = r_j$ и $\omega_i = \omega_j$, то работы идентичны. В такой ситуации более приоритетной будет считаться работа с бóльшим номером.

Утверждение 2. *Для поиска оптимального решения достаточно рассматривать расписания, в которых интервал $[S_j, C_j]$ не содержит фрагментов работы с весом, меньшим ω_j.*

Доказательство. Если есть работа j, которая прерывает работу с меньшим весом, то работа j завершается до того, как работа с меньшим весом возобновит своё выполнение. Это означает, что работа i с меньшим весом либо выполняется целиком до начала работы j с бóльшим весом, либо целиком после, либо окаймляет её. Этот факт подробно исследован в [6]. Утверждение 2 доказано.

Далее опишем некоторые предварительные процедуры обработки данных, позволяющие свести задачу $1|r_i, p_i = p, pmtn|\sum \omega_i C_i$ к решению серии задач более простой структуры.

Упорядочим работы по неубыванию весов $\omega_1 \leqslant \omega_2 \leqslant \ldots \leqslant \omega_n$. Рассмотрим две неконкурентные работы i и j, для которых $\omega_i \leqslant \omega_j$ и $r_i \geqslant r_j$. По утверждению 1 работа j целиком выполняется раньше работы i, значит, $C_j \leqslant S_i$. Если интервалы $[r_j, r_j + p]$ и $[r_i, r_i + p]$ пересекаются, т. е. $r_j \leqslant r_i < r_j + p$, то раньше момента $r_j + p$ работа i начаться не может, поэтому значение r_i можно увеличить до $r_j + p$. Если работы идентичны, то приоритет отдаём работе с бóльшим номером. Сделав это для каждой пары неконкурентных работ с пересекающимися интервалами $[r_j, r_j + p]$ и $[r_i, r_i + p]$ (возможно неоднократно), придём к ситуации, когда все r_i будут различны. Напомним, что если $r_i = r_j$, то работы i и j неконкурентны, и к ним применима эта процедура.

Результаты предобработки отображены на рис. 1 и 2. Под каждую работу отводим временну́ю ось и эти оси располагаем одну под другой, начиная с работы с наибольшим весом и далее по убыванию весов. Моменты поступления работ выделены жирной чертой. На рис. 1 изображён

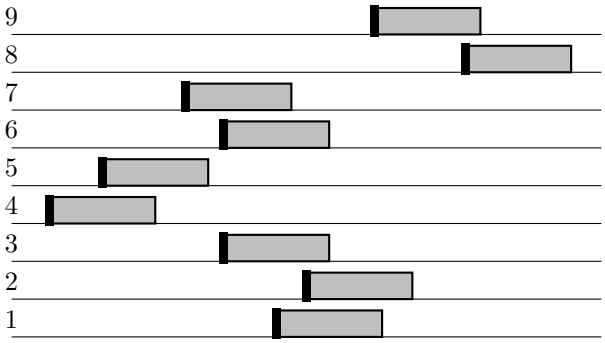


Рис. 1. Входные данные

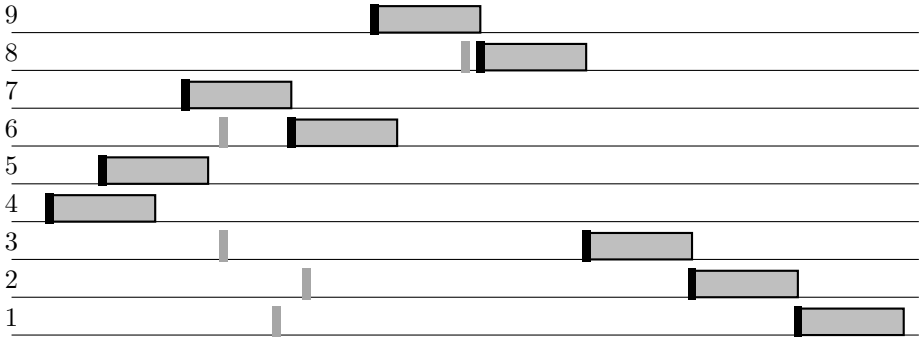


Рис. 2. Входные данные после предобработки

пример входных данных. Здесь, как и ранее, $\omega_1 \leqslant \omega_2 \leqslant \ldots \leqslant \omega_9$. Работы отображены интервалами $[r_i, r_i + p]$. На рис. 2 представлены входные данные после предобработки.

Далее рассмотрим процедуру декомпозиции задачи. Напомним, что работа с номером 1 имеет самый маленький вес. Выделим подмножество работ, которые поступают не позже r_1. Пусть их k штук. По утверждению 1 все эти работы завершаются до начала работы 1, но тогда она не может начинаться раньше момента pk, и можно положить $r_1 = pk$. Далее повторим эту процедуру, пока ранний срок начала выполнения работы 1 можно будет сдвигать.

Если работа 1 сдвинется в самый конец, она завершится в момент pn. Тем самым она не влияет на выполнение других работ, и получаем задачу меньшей размерности. В приведённом выше примере (рис. 1 и 2) задача с 8 работами в итоге сводится к задаче с пятью работами $\{4, 5, 6, 7, 8\}$.

Пусть r_1 увеличилось до pk, $k < n$. Тогда выполнение работы 1 однозначно начинается в этот момент, так как альтернативных работ нет.

Если такие были бы, то работа 1 сдвинулась бы ещё дальше, т. е. для всех остальных работ $r_i > pk$. В этом случае задачу можно разбить на две подзадачи: в первую войдут k работ с моментами $r_i \leqslant p(k-1)$, во вторую — все остальные, включая работу 1, причём работа 1 поступает раньше всех. Кроме того, отметим, что если во второй подзадаче все $r_i \geqslant pk + p$, то работа 1 полностью выполняется в интервале $[pk, pk + p]$, и можно рассматривать задачу без неё, т. е. задача вновь разбивается. Декомпозиции не будет, если отрезки $[r_1, r_1 + p]$ и $[r_i, r_i + p]$ пересекаются.

Отметим, что в первой подзадаче, которая решается на интервале $[0, kp]$, вновь можно применить этот подход со сдвигом работы с меньшим весом, если она не поступает первой. Таким образом, исходная задача разбивается на задачи меньшей размерности, в каждой из которых первой поступает работа с наименьшим весом, а до её завершения поступает следующая работа.

Развивая описанный подход далее, можно утверждать, что j-я по порядку поступления в систему работа должна поступить не позднее момента $p(j-1) - 1$. Иначе все предшествующие $j - 1$ работ успевают завершить своё выполнение и к задаче снова можно применить декомпозицию. Задачу, которую нельзя упростить указанными процедурами, будем называть *приведённой*. В приведённой задаче

- работа с наименьшим весом поступает первой в нулевой момент времени;
- все времена поступлений r_i, $i = 1, 2, \ldots, n$, различны;
- j-я по порядку поступления работа поступает не позднее момента $p(j-1) - 1$, $j = 2, 3, \ldots, n$;
- достаточно рассматривать расписания длины pn, в которых простои станка отсутствуют.

Далее будем работать только с приведёнными расписаниями.

3. Построение конечного множества расписаний

В этом разделе опишем процедуру выделения конечного подмножества расписаний, которое содержит оптимальное.

Утверждение 3. *Для построения оптимального решения достаточно рассмотреть расписания, в которых переключение станка на выполнение другой работы происходит только в момент r_i или в момент C_j окончания некоторой работы.*

Доказательство. В момент окончания некоторой работы станок освобождается и, естественно, переходит в состояние простоя или выполнения другой доступной работы. Предположим, что переключение произошло в момент, не совпадающий с некоторым r_i, и станок переключился с выполнения работы j на выполнение работы k. В этом случае

в соответствии с [6] работа j окаймляет работу k. Тогда найдётся $\varepsilon > 0$ такое, что некоторый фрагмент работы j длины ε, предшествующий моменту прерывания, можно поменять с фрагментом работы k в интервале $[C_k - \varepsilon, C_k]$. В этом случае время завершения работы k, а значит, и значение целевой функции уменьшатся. Параметр ε может быть выбран как минимальное значение трёх величин: длины фрагмента работы j, предшествующего прерыванию, длины заключительного фрагмента работы k и резерва сдвига работы k, равного $S_k - r_k$. Утверждение 3 доказано.

Отсюда вытекает, что в приведённой задаче существует оптимальное расписание, в котором все моменты прерываний работ принадлежат множеству $\{r_2, r_3, \ldots, r_n\}$, а суммарное число переключений станка не превосходит $2n-2$. Далее будем иметь дело только с такими расписаниями. Рассмотрим возможные альтернативы выбора работ в моменты переключения станка.

Утверждение 4. *Для построения оптимального решения достаточно рассмотреть расписания, в которых прерывание текущей работы в момент r_j происходит только в том случае, когда поступившая работа имеет наибольший вес среди всех незавершённых и доступных для выполнения работ. При этом в момент r_j начинает выполнение работа j.*

Доказательство. Первая часть утверждения очевидна, поскольку если вес работы j меньше веса некоторой ранее поступившей работы, то по утверждению 1 эта работа должна быть завершена раньше j. Доказательство второй части утверждения 4 аналогично обоснованию утверждения 3. Если текущая работа прервалась работой k, для которой $r_k < r_j$, то фрагменты работы k и текущей работы могут быть переставлены, как ранее. При этом целевая функция уменьшится.

Таким образом, в момент r_j имеет место альтернатива выбора между текущей работой i и поступившей работой j. При этом если выбор происходит в пользу работы j, то она целиком завершится до возобновления работы i. Если выбирается работа i, то до её завершения работа j не начнёт выполняться. Утверждение 4 доказано.

Опишем, какие есть варианты продолжения расписания в момент C_j окончания некоторой работы.

1. Все работы завершили своё выполнение. Расписание построено.

2. Не все работы выполнены, но в момент времени C_j незавершённых и готовых к выполнению работ нет. Тогда станок будет простаивать до поступления следующей работы. Такая ситуация не возникает, если мы решаем приведённую задачу.

3. Есть доступные работы, незавершённых работ нет. Выполнение начнёт работа с наибольшим весом.

4. Есть незавершённые работы, при этом все доступные уже начаты. Тогда продолжит выполнение последняя прерванная работа.

5. Есть и прерванные, и доступные работы. Если i — последняя из прерванных работ, j — доступная работа с наибольшим весом и при этом $i > j$, то продолжаем выполнять работу i.

6. Наконец, есть и прерванные, и доступные работы, i — последняя из прерванных работ, j — доступная работа с наибольшим весом и при этом $i < j$. Тогда возникает альтернатива выбора между i и j, т. е. либо продолжит выполнение последняя из прерванных работ, либо начнёт выполнение доступная работа с максимальным весом.

Опишем алгоритм формирования всех расписаний, удовлетворяющих описанным выше свойствам. Работы упорядочены в лексикографическом возрастающем порядке векторов (ω_1, r_i). Алгоритм заключается в последовательном просмотре моментов r_i и C_i в порядке их возрастания, и ветвлении вариантов в случае альтернативы выбора работ. Напомним, что в момент r_i возможна альтернатива между текущей работой и поступившей работой i, а в C_i — между последней прерванной работой и самой тяжёлой из доступных, но не начавших выполнения работ.

Первоначально формируется стек T из работ, находящихся в процессе выполнения и множество работ D, доступных для выполнения. Стек T организуем по стандартному правилу «последний зашёл — первый вышел». Первоначально полагаем $T = (1)$ и $D = \varnothing$. Получаем некоторое частичное расписание.

Очередной шаг. Рассматриваем любое из построенных частичных расписаний. Пусть в этом расписании часть работ уже завершена, часть прервана, и выполняется работа k, которая стоит последней в стеке T. Выполняем эту работу до момента C_k или ближайшего r_m. Если $C_k \leqslant r_m$ или все работы уже поступили в систему, то работу k из T исключаем. При $r_m \leqslant C_k$, в D добавляем работу m.

Далее в T берём последний элемент (пусть это будет номер i), а в D — элемент с наибольшим весом (пусть это j). Если $i > j$, то в соответствии со случаем 5 продолжаем выполнять работу i, иначе ветвим. В одном варианте выполняем работу i. Тогда T и D не меняем. Во втором выполняем работу j. Её надо удалить из D и поместить в стек T. Получили два новых варианта частичных расписаний. Переходим на очередной шаг.

Построение любого частичного расписания завершается, когда D и T становятся пустыми.

Число построенных расписаний обозначим через Q. Так как выбор осуществляется только в моменты r_j, кроме минимального, и в моменты C_j, кроме двух последних, может быть сгенерировано не более 2^{2n-3} различных расписаний, но это грубая оценка. На самом деле значение Q меньше, поскольку ветвление происходит только тогда, когда и стек,

и очередь одновременно не пустые. Максимум Q достигается на примерах, когда все пары работ конкурентные и работы поступают в систему равномерно через промежуток времени $\frac{p}{2}$. Анализ результатов работы программы показывает, что в этом случае $Q = O(e^n)$.

Далее проведём анализ расписаний построенного множества с учётом весов работ.

4. Параметрический анализ

Зафиксируем значения p и r_i, $i = 1, 2, \ldots, n$, и описанным выше алгоритмом построим конечное множество расписаний. При формировании этого множества не учитываются значения весовых коэффициентов ω_i, $i = 1, 2, \ldots, n$. Их будем рассматривать как параметры. Без ограничения общности можно считать, что $0 < \omega_1 \leqslant \omega_2 \leqslant \ldots \leqslant \omega_n$. Более того, веса можно ограничить, положив $\omega_n = 1$.

Пусть C_i^q — момент завершения работы i в расписании $q = 1, 2, \ldots, Q$. Чтобы расписание q_0 было оптимальным, необходимо и достаточно, чтобы была совместна следующая система неравенств:

$$\sum_{i=1}^{n} C_i^{q_0}\omega_i \leqslant \sum_{i=1}^{n} C_i^{q}\omega_i, \quad q = 1, 2, \ldots, Q, \ q \neq q_0,$$
$$0 < \omega_1 \leqslant \omega_2 \leqslant \ldots \leqslant \omega_n = 1.$$

Проверяя совместность системы для каждого из Q сформированных расписаний, выделим подмножество расписаний, оптимальных при некоторых значениях весов. Обозначим его через OPT. Отметим некоторую особенность формирования этого подмножества. В единичном $(n-1)$-мерном кубе рассмотрим область $0 < \omega_1 \leqslant \omega_2 \leqslant \ldots \leqslant \omega_{n-1} \leqslant 1$. Каждая гиперплоскость $\sum\limits_{i=1}^{n} C_i^{q_1}\omega_i = \sum\limits_{i=1}^{n} C_i^{q_2}\omega_i$ будет разделять области, в которых будут лучше расписания q_1 или q_2 соответственно. Все C_Q^2 гиперплоскостей различны. Каждому расписанию $q_0 \in$ OPT будет соответствовать некоторый открытый многогранник, в котором это расписание лучше всех остальных. Если такой области для расписания нет, то оно может быть оптимальным при некоторых значениях весов. Однако при этих весах такое же значение целевой функции имеют расписания из OPT, и в результате по совокупности доминируют его. Такие расписания рассматривать не имеет смысла. В программной реализации для формирования OPT использовали неравенства $\sum\limits_{i=1}^{n} C_i^{q_0}\omega_i + \varepsilon \leqslant \sum\limits_{i=1}^{n} C_i^{q}$ для некоторого малого ε.

В табл. 1 приведены значения Q и мощность выделенного подмножества OPT для случая $p = 2$ и $r_j = j - 1$, $j = 1, 2, \ldots, n$.

Таблица 1

n	3	4	5	6	7	8	9	10	11	12	n
Q	4	9	21	51	127	323	835	2188	5798	15511	$\sim e^n$
$\lvert\mathrm{OPT}\rvert$	3	6	12	24	48	96	192	384	768	1532	$3 \cdot 2^{n-3}$

Как видно из табл. 1, мощность этого подмножества также растёт экспоненциально с ростом числа работ. Однако его исследование позволяет получить дополнительные свойства и сформировать подходы к построению алгоритма решения задачи.

Параметрический анализ одного примера при фиксированных r_i, $i = 1, 2, \ldots, n$, и p для $n = 12$ занимает несколько минут на процессоре Intel Core i7-6700HQ 2,60 ГГц. Анализ предусматривает формирование подмножества OPT и поиск весов, при которых расписание $q \in \mathrm{OPT}$ оптимально. С помощью этой программы можно получать граничные значения, в пределах которых могут изменяться веса при сохранении оптимальности расписания q. Для остальных расписаний выдаётся информация, что система строгих неравенств несовместна. При размерности $n > 12$ полный анализ проводить нет необходимости, однако для $n \leqslant 24$ программа позволяет за приемлемое время проводить выборочный анализ конкретных расписаний и проверять различные гипотезы.

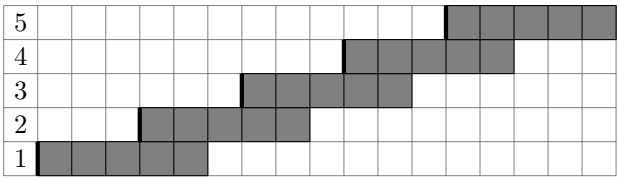


Рис. 3. Входные данные примера

Проиллюстрируем, что может дать такой подход. Как и выше, под каждую работу отводим временну́ю ось, и эти оси располагаем одну под другой, начиная с работы с наибольшим весом, и далее по убыванию. Время поступления работы будем выделять жирной чертой. Рассмотрим пример с пятью работами длительности $p = 5$ и временами поступления $r_1 = 0$, $r_2 = 3$, $r_3 = 6$, $r_4 = 9$, $r_5 = 12$. В нём достаточно отражены свойства, по которым можно проследить основные особенности и сделать необходимые выводы. Входные данные примера изображены на рис. 3. Предобработки входных данных не требуется, так как задача приведённая.

С помощью описанного выше алгоритма построены возможные расписания. Всего их 49, причём оптимальными могут быть только 25. Ниже отражены несколько типичных расписаний, для которых выписанная

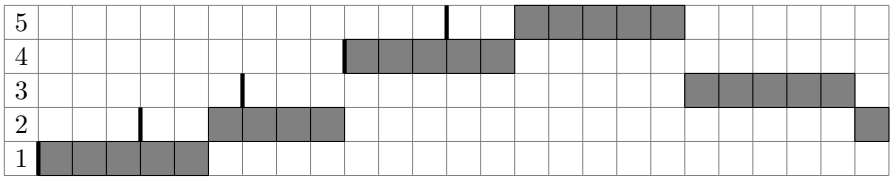


Рис. 4. Прямая перестановка фрагментов расписания

система линейных неравенств несовместна, т. е. решение не будет оптимальным ни при каких весах. Проанализируем причины их неоптимальности.

Самый простой случай — это непосредственная перестановка частей работ. Фрагменты работ 2 и 3 расписания на рис. 4 могут быть переставлены следующим образом (рис. 5.) При этом значение C_2 не изменилось, а C_3 уменьшилось на 3 единицы.

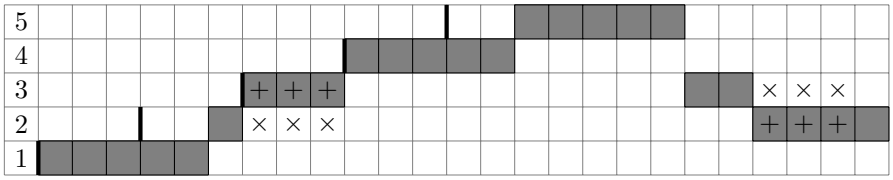


Рис. 5. Исправленное решение

Расписание на рис. 6 также неоптимально по следующей причине. Фрагмент работы 1 уступает фрагменту работы 2, а следовательно, и 3, а далее работе 4 и при этом выигрывает у работы 5, вес которой больше, чем у работы 4: $\frac{5}{\omega_4} < \frac{2}{\omega_3} < \frac{2}{\omega_2} < \frac{2}{\omega_2} < \frac{5}{\omega_5}$. Тем самым $\omega_4 > \omega_5$; противоречие.

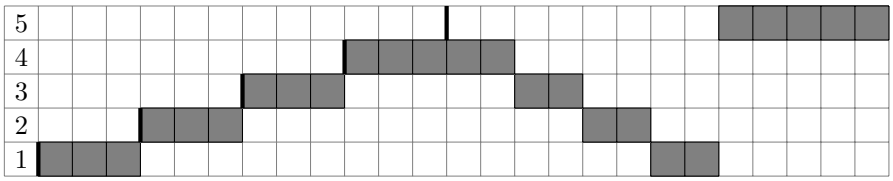


Рис. 6. Транзитивность приоритетов фрагментов работ

С помощью параметрического анализа конкретных примеров удалось выявить и в последствии обосновать несколько свойств оптимальных расписаний.

Утверждение 5. *Для приведённой задачи существует оптимальное расписание, в котором время завершения работы с минимальным весом C_1 кратно p.*

Доказательство очевидно, так как по утверждению 2 все работы, которые прерывали её, должны быть завершены к моменту C_1. Утверждение 5 доказано.

Утверждение 6. *Если в оптимальном решении приведённой задачи выполнено $C_1 \geqslant \max\limits_{j=1,2,\ldots,n} r_j$, то в интервале $[C_1, pn]$ все работы выполняются без прерываний и все фрагменты имеют длительность p.*

Доказательство непосредственно следует из утверждения 2, так как работа 1 имеет наименьший вес. Любая другая работа либо завершится до момента C_1, либо начнёт выполнение после работы 1. Во втором случае условие $C_1 \geqslant \max\limits_{j=1,2,\ldots,n} r_j$ и утверждение 3 гарантируют непрерывность выполнения работ после момента C_1. Утверждение 6 доказано.

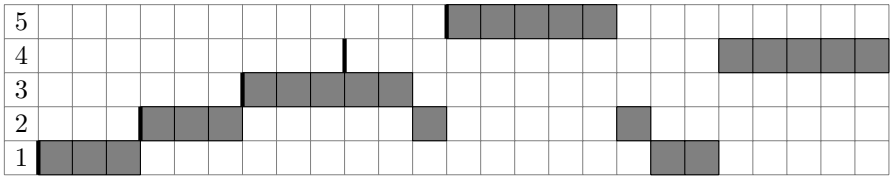


Рис. 7. Неоптимальное расписание

Важным является расписание на рис. 7. Оно не принадлежит множеству OPT, но одного доминирующего его расписания не существует. В данном случае доминирующей является группа из трёх следующих расписаний $(C_1, C_2, C_3, C_4, C_5)$: $(25, 23, 11, 21, 17)$, $(25, 8, 23, 19, 17)$ и $(5, 25, 11, 21, 17)$.

5. Полиномиально разрешимый случай задачи

В данном разделе выделен полиномиально разрешимый случай задачи, когда к моменту поступления очередной работы остаётся невыполненной только одна работа из ранее поступивших. Такое поступление можно назвать разреженным.

Рассмотрим приведённую подзадачу и упорядочим работы по возрастанию времён поступления $r_1 < r_2 < \cdots < r_n$. В приведённой задаче первая работа имеет наименьший вес, $r_1 = 0$ и $r_i < (i-1)p$, $i = 2, 3, \ldots, n$. Пусть для всех $i = 3, 4, \ldots, n$ имеет место условие $r_i \geqslant r_{i-1} + p$. Тогда к моменту r_i успевают завершиться ровно $i-2$ ранее поступивших работ,

и только одна работа к этому моменту не завершает своего выполнения. Пример входных данных такого типа приведён на рис. 8.

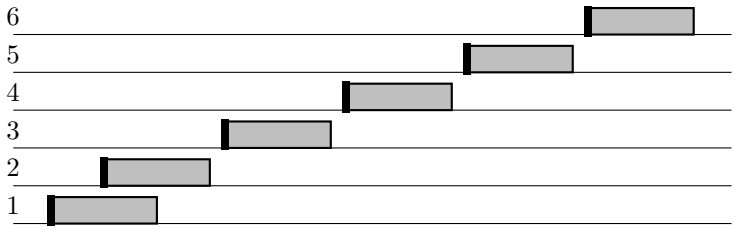


Рис. 8. Пример слабого перекрытия

Ключевым аспектом при составлении расписания является принятие решений в моменты времени r_i, $i = 2, 3, \ldots, n-1$, относительно того, продолжить выполнение текущей работы или прервать её и начать выполнение поступившей работы i. Это справедливо в общем случае. Конкретика для данной подзадачи заключается в том, что к моменту r_i уже $i-2$ работ завершили своё выполнение. Это позволяет реализовать схему динамического программирования с полиномиальной трудоёмкостью.

Обозначим через $\varphi(i, j)$ оптимальное значение взвешенной суммы работ, завершившихся после момента r_i, если в этот момент из ранее начатых не завершена работа $j \in \{1, 2, \ldots, i-1\}$. Такая работа только одна. Необходимо найти $\varphi(2, 1)$.

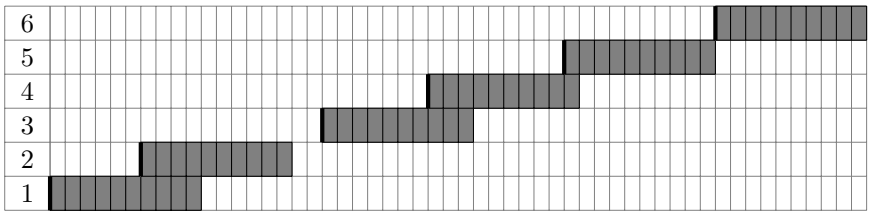


Рис. 9. Входные данные примера

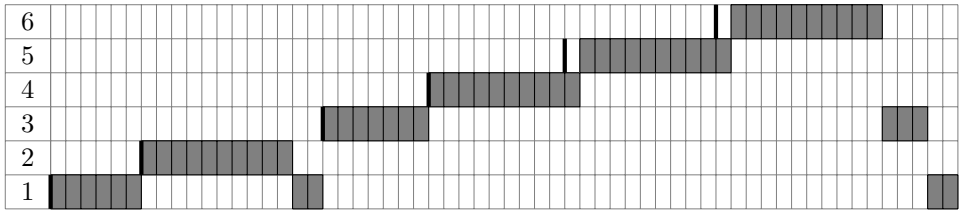


Рис. 10. Оптимальное решение примера

Начинаем с конца, с момента r_n. Для всех $j \in \{1, 2, \ldots, n-1\}$

$$\varphi(n, j) = \min\{(r_n + p)\omega_n + np\omega_j,\ (n-1)p\omega_j + np\omega_n\}.$$

В первом случае приоритет у работы n, а работа j прерывается и завершается в конце. Во втором случае приоритет у работы j, а работа n сдвигается в конец.

Далее для всех $i = n-1, n-2, \ldots, 2$ и всех $j = 1, 2, \ldots, i-1$

$$\varphi(i, j) = \min\{(r_i + p)\omega_i + \varphi(i+1, j),\ (i-1)p\omega_j + \varphi(i+1, i)\}.$$

Построение оптимального решения происходит стандартным способом.

Иллюстративный пример с $n = 6$, $p = 10$, временами поступлений $(0, 6, 10, 25, 34, 44)$ и весами $(8, 12, 18, 30, 33, 50)$ приведён на рис. 9 и 10. Трудоёмкость предложенного алгоритма динамического программирования составляет $O(n^2)$ операций.

Заключение

В статье исследуется задача минимизации среднего взвешенного времени завершения работ одинаковой длительности на одном станке с заданными сроками поступления работ и разрешением прерываний их выполнения. Описан алгоритм предобработки входных данных, позволяющий свести задачу к последовательному решению задач более простой и регулярной структуры. Предложен алгоритм построения конечного подмножества решений, которое содержит оптимальное расписание.

Разработана и реализована технология анализа конкретных расписаний. Это позволяет выдвигать и проверять различные гипотезы по структуре оптимальных расписаний. В частности, для каждого расписания построенного конечного подмножества можно найти веса и границы их изменений, при которых расписание будет оптимальным, или установить, что таких весов нет. Во втором случае можно выделить подмножество расписаний, которое доминирует указанное, и сделать анализ причин этого доминирования. Анализ всех расписаний построенного конечного подмножества позволяет сформировать ещё более узкий класс решений OPT без потери оптимума.

Описан полиномиально разрешимый случай задачи.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0020). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Pinedo M. L.** Scheduling: Theory, algorithms, and systems. New York: Springer, 2008. 678 p. DOI: 10.1007/978-0-387-78935-4.
2. **Lenstra J. K., Rinnooy Kan A. H. G., Brucker P.** Complexity of machine scheduling problems // Ann. Discrete Math. 1977. V. 1. P. 343–362. DOI: 10.1016/S0167-5060(08)70743-X.
3. **Baptiste P.** Scheduling equal-length jobs on identical parallel machines // Discrete Appl. Math. 2000. V. 103, No. 1–3. P. 21–32. DOI: 10.1016/S0166-218X(99)00238-3.
4. **Labetoulle J., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Preemptive scheduling of uniform machines subject to release dates // Progress in combinatorial optimization. Toronto: Acad. Press, 1984. P. 245–261. DOI: 10.1016/B978-0-12-566780-7.50020-9.
5. **Baker K. R.** Introduction to sequencing and scheduling. New York: John Wiley & Sons, 1974. 318 p.
6. **Баптист Ф., Карлье Ж., Кононов А. В., Керан М., Севастьянов С. В., Свириденко М. И.** Структурные свойства оптимальных расписаний с прерываниями операций // Дискрет. анализ и исслед. операций. 2009. Т. 16, № 1. Р. 3–36.
7. **Batsyn M., Goldengorin B., Pardalos P. M., Sukhov P.** Online heuristic for the preemptive single machine scheduling problem of minimizing the total weighted completion time // Optim. Methods Softw. 2014. V. 29. P. 955–963. DOI: 10.1080/10556788.2013.854360.
8. **Batsyn M., Goldengorin B., Sukhov P., Pardalos P. M.** Lower and upper bounds for the preemptive single machine scheduling problem with equal processing times // Models, algorithms, and technologies for network analysis. Proc. 2nd Int. Conf. Network Analysis (Nizhny Novgorod, Russia, May 7–9, 2012). New York: Springer, 2013. P. 11–27. (Springer Proc. Math. Stat.; V. 59). DOI: 10.1007/978-1-4614-8588-9_2.
9. **Goemans M.-X., Queyranne M., Schulz A.-S., Skutella M., Wang Y.** Single machine scheduling with release dates // SIAM J. Discrete Math. 2002. V. 15, No. 2. P. 165–192. DOI: 10.1137/S089548019936223X.
10. **Kravchenko S. A., Werner F.** Scheduling jobs with equal processing times // IFAC Proc. Volumes. 2009. V. 42, No. 4. P. 1262–1267. DOI: 10.3182/20090603-3-RU-2001.0042.
11. **Fomin A., Goldengorin B.** An efficient model for the preemptive single machine scheduling of equal-length jobs. Ithaca, NY: Cornell Univ., 2020. (Cornell Univ. Libr. e-Print Archive; arXiv:2012.08152). DOI: 10.48550/arXiv.2012.08152.
12. **Chernykh K. A., Servakh V. V.** The structure of the optimal solution of the problem of one machine with the possibility of interruptions of jobs // Proc. 14th Int. Asian School-Seminar «Optimization Problems of Complex Systems» (Kara-Oi, Kyrgyzstan, July 20–31, 2018). Piscataway: IEEE, 2018. P. 312–321.

13. Chernykh K. A., Servakh V. V. Combinatorial structure of optimal solutions to the problem of a single machine with preemption // Proc. 15th Int. Asian School-Seminar «Optimization Problems of Complex Systems» (Novosibirsk, Russia, Aug. 26–30, 2019). Piscataway: IEEE, 2019. P. 21–26. DOI: 10.1109/OPCS.2019.8880148.

14. Chernykh K. A., Servakh V. V. Analysis of optimal solutions to the problem of a single machine with preemption // Mathematical optimization theory and operations research: Recent trends. Rev. Sel. Pap. 20th Int. Conf. (Irkutsk, Russia, July 5–10, 2021). Cham: Springer, 2021. P. 163–174. (Commun. Comput. Inf. Sci.; V. 1476). DOI: 10.1007/978-3-030-86433-0_11.

15. Jaramillo F., Erkoc M. Minimizing total weighted tardiness and overtime costs for single machine preemptive scheduling // Comput. Ind. Eng. 2017. V. 107. P. 109–119. DOI: 10.1016/j.cie.2017.03.012.

16. Jaramillo F., Keles B., Erkoc M. Modeling single machine preemptive scheduling problems for computational efficiency // Ann. Oper. Res. 2020. V. 285. P. 197–222. DOI: 10.1007/s10479-019-03298-9.

Ляшкова Ксения Андреевна
Сервах Владимир Вицентьевич

Статья поступила
26 июля 2022 г.
После доработки —
17 января 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 105–122

UDC 519.8+518.25 DOI: 10.33048/daio.2024.31.750

THE PROBLEM OF ONE MACHINE WITH EQUAL PROCESSING TIME AND PREEMPTION

K. A. Lyashkova [a] and *V. V. Servakh* [b]

Omsk Branch of the Sobolev Institute of Mathematics,
13 Pevtsov Street, 644099 Omsk, Russia
E-mail: [a] ksech@bk.ru, [b] svv_usa@rambler.ru

Abstract. We consider the problem of minimizing the weighted average execution time of equal-length jobs performance on one machine at the specified time of job arrival and the possibility of their interruption. The computational complexity of this problem is currently unknown. The article proposes an algorithm for preprocessing input data that allows reducing the problem to a narrower and more regular class of examples. The properties of optimal solutions are substantiated. Based on them, an algorithm for constructing a finite subset of solutions containing an optimal schedule has been developed. A parametric analysis of the schedules in this subset has been carried out that makes it possible to form a subclass of schedules that are optimal at some values of weights. A polynomially solvable case of the problem is isolated. Tab. 1, illustr. 10, bibliogr. 16.

Keywords: schedule theory, one machine, preemption.

References

1. **M. L. Pinedo,** *Scheduling: Theory, Algorithms, and Systems* (Springer, New York, 2008), DOI: 10.1007/978-0-387-78935-4.
2. **J. K. Lenstra, A. H. G. Rinnooy Kan,** and **P. Brucker,** Complexity of machine scheduling problems, *Ann. Discrete Math.* **1**, 343–362 (1977), DOI: 10.1016/S0167-5060(08)70743-X.
3. **P. Baptiste,** Scheduling equal-length jobs on identical parallel machines, *Discrete Appl. Math.* **103** (1–3), 21–32 (2000), DOI: 10.1016/S0166-218X(99)00238-3.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 479–488 (2024), DOI: 10.1134/S1990478924030098.

4. **J. Labetoulle, E. L. Lawler, J. K. Lenstra,** and **A. H. G. Rinnooy Kan,** Preemptive scheduling of uniform machines subject to release dates, in *Progress in Combinatorial Optimization* (Acad. Press, Toronto, 1984), pp. 245–261, DOI: 10.1016/B978-0-12-566780-7.50020-9.
5. **K. R. Baker,** Introduction to Sequencing and Scheduling (John Wiley & Sons, New York, 1974).
6. **P. Baptiste, J. Carlier, A. V. Kononov, M. Queyranne, S. V. Sevast'yanov,** and **M. I. Sviridenko,** Structural properties of optimal schedules with preemption, *Diskretn. Anal. Issled. Oper.* **16** (1), 3–36 (2009) [Russian] [*J. Appl. Ind. Math.* **4** (4), 455–474 (2010)].
7. **M. Batsyn, B. Goldengorin, P. M. Pardalos,** and **P. Sukhov,** Online heuristic for the preemptive single machine scheduling problem of minimizing the total weighted completion time, *Optim. Methods Softw.* **29**, 955–963 (2014), DOI: 10.1080/10556788.2013.854360.
8. **M. Batsyn, B. Goldengorin, P. Sukhov,** and **P. M. Pardalos,** Lower and upper bounds for the preemptive single machine scheduling problem with equal processing times, in *Models, Algorithms, and Technologies for Network Analysis* (Proc. 2nd Int. Conf. Network Analysis, Nizhny Novgorod, Russia, May 7–9, 2012) (Springer, New York, 2013), pp. 11–27 (Springer Proc. Math. Stat., Vol. 59), DOI: 10.1007/978-1-4614-8588-9_2.
9. **M.-X. Goemans, M. Queyranne, A.-S. Schulz, M. Skutella,** and **Y. Wang,** Single machine scheduling with release dates, *SIAM J. Discrete Math.* **15** (2), 165–192 (2002), DOI: 10.1137/S089548019936223X.
10. **S. A. Kravchenko** and **F. Werner,** Scheduling jobs with equal processing times, *IFAC Proc. Volumes* **42** (4), 1262–1267 (2009), DOI: 10.3182/20090603-3-RU-2001.0042.
11. **A. Fomin** and **B. Goldengorin,** An efficient model for the preemptive single machine scheduling of equal-length jobs (Cornell Univ., Ithaca, NY, 2020) (Cornell Univ. Libr. e-Print Archive, arXiv:2012.08152), DOI: 10.48550/arXiv.2012.08152.
12. **K. A. Chernykh** and **V. V. Servakh,** The structure of the optimal solution of the problem of one machine with the possibility of interruptions of jobs, in *Proc. 14th Int. Asian School-Seminar "Optimization Problems of Complex Systems", Kara-Oi, Kyrgyzstan, July 20–31, 2018* (IEEE, Piscataway, 2018), pp. 312–321.
13. **K. A. Chernykh** and **V. V. Servakh,** Combinatorial structure of optimal solutions to the problem of a single machine with preemption, in *Proc. 15th Int. Asian School-Seminar "Optimization Problems of Complex Systems", Novosibirsk, Russia, Aug. 26–30, 2019* (IEEE, Piscataway, 2019), pp. 21–26, DOI: 10.1109/OPCS.2019.8880148.
14. **K. A. Chernykh** and **V. V. Servakh,** Analysis of optimal solutions to the problem of a single machine with preemption, in *Mathematical Optimization Theory and Operations Research: Recent Trends* (Rev. Sel. Pap. 20th Int. Conf., Irkutsk, Russia, July 5–10, 2021) (Springer, Cham, 2021), pp. 163–174 (Commun. Comput. Inf. Sci., Vol. 1476), DOI: 10.1007/978-3-030-86433-0_11.

15. F. Jaramillo and **M. Erkoc,** Minimizing total weighted tardiness and overtime costs for single machine preemptive scheduling, *Comput. Ind. Eng.* **107**, 109–119 (2017), DOI: 10.1016/j.cie.2017.03.012.

16. F. Jaramillo, B. Keles, and **M. Erkoc,** Modeling single machine preemptive scheduling problems for computational efficiency, *Ann. Oper. Res.* **285**, 197–222 (2020), DOI: 10.1007/s10479-019-03298-9.

Ksenia A. Lyashkova
Vladimir V. Servakh

Received July 26, 2022
Revised January 17, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 123–143

УДК 519.8+519.6 DOI: 10.33048/daio.2024.31.782

ПОИСК ЛОКАЛЬНО ОПТИМАЛЬНЫХ СТРАТЕГИЙ В ЛИНЕЙНОЙ ИГРОВОЙ ЗАДАЧЕ С БЛАГОПРИЯТНЫМИ СИТУАЦИЯМИ

А. Р. Маматов

Самаркандский гос. университет им. Ш. Рашидова,
Университетский б-р, 15, 140104 Самарканд, Узбекистан

E-mail: akmm1964@rambler.ru

Аннотация. Рассматривается линейная игровая задача двух игроков. Два игрока поочерёдно выбирают свои стратегии из соответствующих множеств. Сначала первый игрок выбирает свою стратегию, затем, зная стратегию первого игрока, второй игрок выбирает свою стратегию. Множество стратегий второго игрока зависит от стратегии первого игрока. Целью первого игрока является выбор стратегии для того, чтобы максимизировать выпуклую кусочно линейную функцию (функцию минимума по стратегии второго игрока). Цель второго игрока — минимизировать линейную функцию. Предложен алгоритм, позволяющий строить стратегии для этой, а также для двойственной задачи, удовлетворяющие необходимым условиям оптимальности «высокого порядка». Этот алгоритм использует формулу приращения целевой функции двойственной задачи. Доказаны теоремы о конечности предложенного алгоритма и его модификации. Приведён пример, иллюстрирующий работу алгоритма. Также приведены результаты численного эксперимента по построению стратегий, удовлетворяющих необходимым условиям оптимальности «высокого порядка» в задачах, элементы которых генерировались датчиком случайных чисел. По результатам численного эксперимента можно сделать вывод, что при исполнении предложенного алгоритма зачастую имеется возможность перехода от одной локально оптимальной стратегии первого игрока к другой стратегии, обеспечивающей возрастание целевой функции. Табл. 1, ил. 1, библиогр. 21.

Ключевые слова: линейная игра, максиминная задача, условие оптимальности, опора, алгоритм.

Введение

Обзор результатов. Конечной и основной проблемой теории оптимизации считается проблема вычислительных методов. В общей теории условий оптимальности в негладких задачах оптимизации [1–6] особое место занимает вывод необходимых условий оптимальности, учитывающих специфику задач, предназначенных для численного решения [7–10]. Как правило, решения таких задач представляются в виде алгоритмов. Существенное место при разработке алгоритмов решения многоэкстремальных задач занимает составление блока алгоритма, осуществляющего переход от одного локально оптимального плана к другому — позволяющему улучшить значение целевой функции.

Так же, как в линейном программировании, наряду с прямыми методами исследования экстремальных задач важную роль играют двойственные методы [11, 12]. Понятие двойственной полуигры (двойственной задачи для линейной максиминной задачи со связанными переменными) введено в работе [13], где доказано, что цены этих полуигр (оптимальные значения целевых функций этих задач) совпадают. Аналогичное утверждение для линейной игры с запрещёнными ситуациями [14] доказано в [15]. В [7, 16] исследованы отношения между максимином и минимаксом.

Отметим, что слабая задача линейного двухуровневого программирования [17] при $d_1 = d_2$ или $d_2 = 0 \in \mathbb{R}^m$ является линейной минимаксной задачей со связанными переменными. В [7, 17] для слабой задачи линейного двухуровневого программирования разработан подход, сочетающий метод штрафной функции и метод ветвей и границ. В этих же работах рассматриваемая задача сведена к линейной минимаксной (максиминной) задаче со связанными переменными.

В [18] для линейной максиминной задачи со связанными переменными доказаны условия, при выполнении которых из совпадения значений целевых функций взаимно двойственных задач следует локальная оптимальность соответствующих стратегий (планов) первых игроков этих задач. В [19] эти теоремы обобщены на случай необходимого условия глобальной оптимальности.

В настоящей работе предложен двойственный алгоритм, который позволяет строить стратегии для линейной игровой задачи с благоприятными ситуациями (для линейной максиминной задачи со связанными переменными), а также стратегии для двойственной задачи, удовлетворяющие необходимым условиям оптимальности «высокого порядка» [19]. Приведены результаты численного эксперимента по построению удовлетворяющих таким условиям стратегий. Задачи и их параметры для численного эксперимента генерировались датчиком случайных чисел.

Постановка задачи и предварительные сведения. Пусть имеются два игрока, которые поочерёдно выбирают векторы из множеств

$$X = \{x \in \mathbb{R}^n \mid f_* \leqslant x \leqslant f^*\},$$
$$Y(x) = \{y \in \mathbb{R}^l \mid g_* \leqslant y \leqslant g^*,\ Ax + By = b\}$$

соответственно: сначала первый игрок выбирает $x \in X$, затем второй игрок выбирает $y \in Y$, зная x. Здесь $f_*, f^* \in \mathbb{R}^n$, $g_*, g^* \in \mathbb{R}^l$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times l}$. Будем предполагать, что $\operatorname{rang} B = m < l$ и $Y(x) \neq \varnothing$ для любого $x \in X$.

Пусть $c \in \mathbb{R}^n$, $d \in \mathbb{R}^l$, а штрих $'$ обозначает транспонирование. Цель первого игрока — найти $\widehat{x} \in X$, доставляющий максимальное значение функции $\varphi\colon X \to \mathbb{R}$:

$$\varphi(x) = \min_{y\in Y(x)} (c'x + d'y),$$

т. е. $\varphi(\widehat{x}) = \max\limits_{x\in X} \varphi(x)$. Цель второго игрока — найти $\widehat{y} \in Y(\widehat{x})$, минимизирующий функцию $d'y\colon Y(\widehat{x}) \to \mathbb{R}$, т. е.

$$d'\widehat{y} = \min_{y\in Y(\widehat{x})} d'y.$$

Другими словами, имеем линейную игровую задачу с благоприятными ситуациями (линейную максиминную задачу со связанными переменными) [18, 19]:

$$\varphi(x) = \min_{y\in Y(x)} (c'x + d'y) \to \max_{x\in X}. \tag{1}$$

Очевидно, что $\widehat{x}$ является точкой глобального максимума функции $\varphi(x)$, $x \in X$. Поиск $\widehat{x}$ составляет важную задачу теории оптимизации из класса задач математического программирования максиминного типа. В отличие от задач линейного программирования, до настоящего времени неизвестен эффективный алгоритм решения этой задачи.

Функция $\varphi(x)$, $x \in X$, выпукла и кусочно линейна [7, 16], поэтому, в общем случае задача (1) многоэкстремальна, т. е. локальный максимум может не дать глобального максимума. В связи с этим представляет особый интерес исследование задачи поиска локального максимума.

Определение 1. Вектор $x \in X$ называется *стратегией* (*планом*) первого игрока, а вектор $y \in Y(x)$ — *x-стратегией* второго игрока.

Определение 2. Стратегия $\widehat{x} \in X$ называется *оптимальной,* если $\varphi(x) \leqslant \varphi(\widehat{x})$ для всех $x \in X$.

Определение 3. При заданном $x \in X$ x-стратегия $\widehat{y} \in Y(x)$ второго игрока называется *оптимальной*, если $d'\widehat{y} \leqslant d'y$ для любого $y \in Y(x)$.

Таким образом, оптимальная стратегия $\widehat{x} \in X$ является решением задачи (1), а оптимальная x-стратегия $\widehat{y} \in Y(x)$, $x \in X$ — решением задачи

$$d'y \to \min_{y \in Y(x)}. \tag{2}$$

Определение 4. Стратегия $\widetilde{x} \in X$ называется *локально оптимальной,* если $\varphi(x) \leqslant \varphi(\widetilde{x})$ для всех $x \in X$ из некоторой окрестности точки $\widetilde{x}$ относительно X.

При решении таких задач естественно обратиться к теории двойственности [11]. В связи с этим в настоящей работе для задачи (1) предложен алгоритм, основанный на теории двойственности, который позволяет строить стратегии игроков, удовлетворяющие необходимым условиям оптимальности «высокого порядка» [19].

1. Двойственная задача

1.1. Формулировка двойственной задачи. В отличие от задач классического линейного программирования, формулировка двойственной к задаче (1) требует некоторой изобретательности. Насколько известно, такая формулировка для задач типа (1) впервые была предложена Ю. П. Иваниловым [13].

Положим

$$M = \{(\mu, s, t) \in \mathbb{R}^{m+2l} \mid B'\mu - t + s = d;\ s \geqslant 0,\ t \geqslant 0\},$$
$$\Lambda(\mu, s, t) = \{(\lambda, \nu) \in \mathbb{R}^{2n} \mid A'\mu - \nu + \lambda = c;\ \nu \geqslant 0,\ \lambda \geqslant 0\},$$
$$\psi(\mu, s, t) = \min_{(\lambda,\nu)\in\Lambda(\mu,s,t)} (b'\mu + g'_* s - g^{*\prime} t + f^{*\prime}\lambda - f'_*\nu).$$

Легко заметить, что $\Lambda(\mu, s, t) \neq \varnothing$ при любом $(\mu, s, t) \in M$.

Определение 5. Задача максимизации функции $\psi(\mu, s, t)$ на M

$$\psi(\mu, s, t) \to \max_{(\mu,s,t)\in M} \tag{3}$$

называется *двойственной* к задаче (1).

В связи с этим задачу (1) назовём *прямой* задачей. Следующее предложение устанавливает связь между прямой и двойственной задачами.

Теорема 1 [20]**.** *Оптимальные значения целевых функций задач* (1) *и* (3) *совпадают.*

Определение 6. Вектор $(\mu, s, t) \in M$ называется *стратегией* первого игрока, вектор $(\lambda, \nu) \in \Lambda(\mu, s, t)$ называется (μ, s, t)-*стратегией* второго игрока в задаче (3).

Подчеркнём, что следует различать соответствующих игроков, участвующих в задачах (1) и (3), а именно: первый игрок в задаче (1) не является также первым игроком в задаче (3), а второй игрок в задаче (1) не является вторым игроком в задаче (3).

Понятия оптимальной и локально оптимальной стратегий первого игрока в задаче (3) вводятся аналогично определениям 1 и 4.

Определение 7. Векторы

$$\delta = B'\mu - d \in \mathbb{R}^l, \quad \nabla = A'\mu - c \in \mathbb{R}^n, \tag{4}$$

построенные по стратегии (μ, s, t) первого игрока (по её компоненте μ) в задаче (3), называются *кострате́гиями* первого и второго игроков для задачи (1) соответственно.

Пусть $K = \{1, 2, \ldots, l\}$, $J = \{1, 2, \ldots, n\}$. Предполагается, что между переменными μ, s, t, λ, ν и переменными μ, δ, ∇ имеются следующие связи:

$$(s_k, t_k) = \begin{cases} (0, \delta_k), & \text{если } \delta_k \geqslant 0, \\ (-\delta_k, 0), & \text{если } \delta_k < 0, \end{cases} \quad k \in K, \tag{5}$$

$$(\nu_j, \lambda_j) = \begin{cases} (\nabla_j, 0), & \text{если } \nabla_j \geqslant 0, \\ (0, -\nabla_j), & \text{если } \nabla_j < 0, \end{cases} \quad j \in J, \tag{6}$$

которые представляют собой условия согласования стратегий (μ, s, t), (λ, ν) игроков задачи (3) с кострате́гиями δ, ∇ задачи (1). Для краткости пару $\beta = (\delta, \nabla)$ назовём *коситуа́цией* (*копла́ном*) задачи (1).

Определение 8. Коситуация $\beta \in \mathbb{R}^{l+n}$ называется *оптима́льной* (*локально оптима́льной*), если согласованная с ней стратегия (μ, s, t) является оптимальной (локально оптимальной) стратегией первого игрока в задаче (3).

1.2. Опорные конструкции. В дальнейшем координаты векторов размерности l разделяются на два типа: опорные и неопорные. Вектор, составленный из опорных координат (в том же порядке), обозначим нижним индексом «оп», а вектор, составленный из остальных координат, обозначим нижним индексом «н». Эти обозначения используются и для подматриц, составленных из опорных и неопорных столбцов.

Пусть $x \in X$ — стратегия первого игрока в задаче (1).

Определение 9 [12]**.** Совокупность $K_{\text{оп}} = \{k_1, k_2, \ldots, k_m\} \subset K$ индексов, удовлетворяющих условию $\det B_{\text{оп}} \neq 0$, называется *опо́рой* задачи (2).

Определение 10. Пара $\{\beta, K_{\text{оп}}\}$ из коситуации β и опоры $K_{\text{оп}}$ называется *опо́рной коситуа́цией* задачи (1).

Определение 11. При заданном $\chi \in \mathbb{R}^n$ вектор $\omega \in \mathbb{R}^l$, удовлетворяющий равенству $A\chi + B\omega = b$, называется *χ-псевдостратегией* второго игрока в задаче (1).

По опорной коситуации $\{\beta, K_{\text{оп}}\}$ построим соответствующие ей стратегию первого игрока χ и χ-псевдостратегию второго игрока ω в задаче (1):

$$\chi_j = \begin{cases} f_{*j}, & \text{если } \nabla_j > 0, \\ f_j^*, & \text{если } \nabla_j < 0, \end{cases} \quad \chi_j \in [f_{*j}, f_j^*], \text{ если } \nabla_j = 0, \quad j \in J,$$

$$\omega_k = \begin{cases} g_{*k}, & \text{если } \delta_k < 0, \\ g_k^*, & \text{если } \delta_k > 0, \end{cases} \quad k \in K_{\text{н}} = K \setminus K_{\text{оп}},$$

$$\omega_k \in [g_{*k}, g_k^*], \quad \text{если } \delta_k = 0,\ k \in K_{\text{н}},$$

$$\omega_{\text{оп}} = B_{\text{оп}}^{-1}(b - A\chi - B_{\text{н}}\omega_{\text{н}}).$$

Для формулировки теоремы, являющейся необходимым условием глобальной оптимальности «высокого порядка», воспользуемся понятием условно соседних опор и лучшей опоры относительно другой опоры. По опоре $K_{\text{оп}}$ построим вектор $\mu \in \mathbb{R}^m$ следующим образом:

$$\mu' = d'_{\text{оп}} B_{\text{оп}}^{-1}. \tag{7}$$

Значение целевой функции задачи (3), вычисленное с помощью опоры $K_{\text{оп}}$ по формулам (4)–(7), обозначим через $\psi((\mu, s, t) \mid K_{\text{оп}})$ [19].

Определение 12 [19]**.** Опоры $K_{\text{оп}}^1, K_{\text{оп}}^2$ называются *условно соседними,* если они отличаются на один элемент.

Пусть $K_{\text{оп}}^1, K_{\text{оп}}^2$ — условно соседние опоры, $(\mu^1, s^1, t^1), (\mu^2, s^2, t^2)$ — соответствующие им стратегии первого игрока задачи (3).

Определение 13 [19]**.** Опора $K_{\text{оп}}^1$ называется *лучшей* по отношению к опоре $K_{\text{оп}}^2$, если $\psi\big((\mu^1, s^1, t^1) \mid K_{\text{оп}}^1\big) > \psi\big((\mu^2, s^2, t^2) \mid K_{\text{оп}}^2\big)$.

Теорема 2 (необходимые условия оптимальности «высокого порядка» [19])**.** *Пусть $\{\beta, K_{\text{оп}}\}$ — опорная коситуация задачи* (1)*, $\delta_{\text{оп}} = 0$, а χ и ω — соответствующие ей стратегия первого игрока и χ-псевдостратегия второго игрока задачи* (1)*. Для оптимальности коситуации β необходимы:*

1) *выполнение соотношений*

$$\omega_k = \begin{cases} g_{*k}, & \text{если } \delta_k < 0, \\ g_k^*, & \text{если } \delta_k > 0, \end{cases} \quad \omega_k \in [g_{*k}, g_k^*], \quad \text{если } \delta_k = 0,\ k \in K_{\text{оп}};$$

2) *отсутствие для опоры $K_{\text{оп}}$ условно соседних лучших опор.*

Приводимый ниже алгоритм основан на формуле приращения целевой функции задачи (3) [19]:

$$\begin{aligned}\Delta\psi(\mu,s,t) = {}& \sum_{k\in K_{\text{оп}}} (\Delta\delta_k\omega_k + g_{*k}\Delta s_k - g_k^*\Delta t_k) + \\ &+ \sum_{\delta_k=0,\,\overline{\delta}_k<0,\,k\in K_{\text{н}}} \overline{\delta}_k(\omega_k - g_{*k}) + \sum_{\delta_k>0,\,\overline{\delta}_k<0,\,k\in K_{\text{н}}} \overline{\delta}_k(g_k^* - g_{*k}) + \\ &+ \sum_{\delta_k=0,\,\overline{\delta}_k\geqslant 0,\,k\in K_{\text{н}}} \overline{\delta}_k(\omega_k - g_k^*) + \sum_{\delta_k<0,\,\overline{\delta}_k\geqslant 0,\,k\in K_{\text{н}}} \overline{\delta}_k(g_{*k} - g_k^*) + \\ &+ \sum_{\nabla_j=0,\,\overline{\nabla}_j<0,\,j\in J} \overline{\nabla}_j(\chi_j - f_j^*) + \sum_{\nabla_j>0,\,\overline{\nabla}_j<0,\,j\in J} \overline{\nabla}_j(f_{*j} - f_j^*) + \\ &+ \sum_{\nabla_j=0,\,\overline{\nabla}_j\geqslant 0,\,j\in J} \overline{\nabla}_j(\chi_j - f_{*j}) + \sum_{\nabla_j<0,\,\overline{\nabla}_j\geqslant 0,\,j\in J} \overline{\nabla}_j(f_j^* - f_{*j}). \end{aligned} \tag{8}$$

2. Основной результат

2.1. Описание алгоритма. Пусть $\{\beta, K_{\text{оп}}\}$ — начальная опорная ситуация задачи (1) с $\delta_{\text{оп}} = 0$, χ и ω — соответствующие ей стратегия первого игрока и χ-псевдостратегия второго игрока в задаче (1).

Шаг 1. Найти набор чисел

$$\gamma_k = \begin{cases} \omega_k - g_{*k}, & \text{если } \omega_k < g_{*k}, \\ \omega_k - g_k^*, & \text{если } \omega_k > g_k^*, \\ 0, & \text{если } \omega_k \in [g_{*k}, g_k^*], \end{cases} \qquad k \in K_{\text{оп}}.$$

Шаг 2. Найти k_0 такое, что $|\gamma_{k_0}| = \max\limits_{k\in K_{\text{оп}}} |\gamma_k|$, и положить $\overline{\alpha}_0 = |\gamma_{k_0}|$.

Шаг 3. Если $\overline{\alpha}_0 = 0$, то положить $\alpha\alpha := 0$, $z := 1$ и перейти к шагу 10.

Шаг 4. Положить $\Delta\delta_{k_0} = \operatorname{sgn}\gamma_{k_0}$, $\Delta\delta_k = 0$, $k \in K_{\text{оп}} \setminus \{k_0\}$, $\overline{\alpha}_0 = |\gamma_{k_0}|$.

Шаг 5. Вычислить

$$\begin{gathered} \Delta\delta'_{\text{н}} = \Delta\delta'_{\text{оп}} B_{\text{оп}}^{-1} B_{\text{н}}, \quad \Delta\nabla' = \Delta\delta'_{\text{оп}} B_{\text{оп}}^{-1} A, \\ \alpha_0 = \overline{\alpha}_0 + \sum_{k\in K_{\text{н}0}^-} \Delta\delta_k(\omega_k - g_{*k}) + \sum_{k\in K_{\text{н}0}^+} \Delta\delta_k(\omega_k - g_k^*) + \\ + \sum_{j\in J_0^-} \Delta\nabla_j(\chi_j - f_j^*) + \sum_{j\in J_0^+} \Delta\nabla_j(\chi_j - f_{*j}), \end{gathered} \tag{9}$$

здесь

$$K_{\text{н}0}^+ = \{k \in K_{\text{н}} \mid \delta_k = 0,\ \Delta\delta_k > 0\}, \quad K_{\text{н}0}^- = \{k \in K_{\text{н}} \mid \delta_k = 0,\ \Delta\delta_k < 0\},$$
$$J_0^+ = \{j \in J \mid \nabla_j = 0,\ \Delta\nabla_j > 0\}, \quad J_0^- = \{j \in J \mid \nabla_j = 0,\ \Delta\nabla_j < 0\}.$$

Шаг 6. Если $\alpha_0 \leqslant 0$, то положить $\sigma^0 = 0$ и, взяв любой индекс (для определённости первый элемент) из $K_{\text{н}0} = K^+_{\text{н}0} \cup K^-_{\text{н}0}$ в качестве k_*, перейти к шагу 8. Иначе определить числа

$$\sigma_k = \begin{cases} -\delta_k/\Delta\delta_k, & \text{если } \delta_k/\Delta\delta_k < 0, \\ \infty & \text{иначе}, \end{cases} \tag{10}$$
$$k \in K^0 = K_{\text{н}} \setminus K_{\text{н}0},$$
$$\xi_j = \begin{cases} -\nabla_j/\Delta\nabla_j, & \text{если } \nabla_j/\Delta\nabla_j < 0, \\ \infty & \text{иначе}, \end{cases} \tag{11}$$
$$j \in J^0 = J \setminus J_0, \quad J_0 = J_0^+ \cup J_0^-.$$

Найти $\sigma^0 = \min\limits_{k \in K^0} \sigma_k$.

Шаг 7. Упорядочить числа $\sigma_k, \xi_j < \infty$, $k \in K^0$, $j \in J^0$ по возрастанию:

$$\begin{gathered} \sigma_{k_1} \leqslant \sigma_{k_2} \leqslant \ldots \leqslant \sigma_{k_r}, \quad r \leqslant |K^0|, \\ \xi_{j_1} \leqslant \xi_{j_2} \leqslant \ldots \leqslant \xi_{j_h}, \quad h \leqslant |J^0|. \end{gathered} \tag{12}$$

Построить множества

$$J_1 = \{j_i \in J^0 \mid \xi_{j_i} < \sigma_{k_1}\},$$
$$J_v = \left\{ j_i \in J^0 \setminus \bigcup_{q=1}^{v-1} J_q \mid \xi_{j_i} < \sigma_{k_v} \right\}, \quad v \in \overline{2, r},$$
$$J_{r+1} = \left\{ j_i \in J^0 \setminus \bigcup_{q=1}^{r} J_q \mid \xi_{j_i} < \infty \right\}.$$

Положить

$$\alpha_0 := \alpha_0 + \sum_{j_i \in J_1 \cap J^-} \Delta\nabla_{j_i}(\chi_{j_i} - f^*_{j_i}) + \sum_{j_i \in J_1 \cap J^+} \Delta\nabla_{j_i}(\chi_{j_i} - f_{*j_i}), \tag{13}$$

здесь $J^+ = \{j \in J \mid \Delta\nabla_j > 0\}$, $J^- = \{j \in J \mid \Delta\nabla_j < 0\}$. Найти

$$\alpha_\tau = \begin{cases} \alpha_{\tau-1} + \Delta\delta_{k_\tau}(\omega_{k_\tau} - g_{*k_\tau}) + \sum\limits_{j_i \in J_{\tau+1} \cap J^-} \Delta\nabla_{j_i}(\chi_{j_i} - f^*_{j_i}) + \\ \quad + \sum\limits_{j_i \in J_{\tau+1} \cap J^+} \Delta\nabla_{j_i}(\chi_{j_i} - f_{*j_i}), \quad \text{если } \Delta\delta_{k_\tau} < 0, \\ \alpha_{\tau-1} + \Delta\delta_{k_\tau}(\omega_{k_\tau} - g^*_{k_\tau}) + \sum\limits_{j_i \in J_{\tau+1} \cap J^-} \Delta\nabla_{j_i}(\chi_{j_i} - f^*_{j_i}) + \\ \quad + \sum\limits_{j_i \in J_{\tau+1} \cap J^+} \Delta\nabla_{j_i}(\chi_{j_i} - f_{*j_i}), \quad \text{если } \Delta\delta_{k_\tau} > 0, \end{cases} \tag{14}$$

для всех $\tau \in \overline{1, r}$, а также индекс $\zeta \in \overline{1, r}$ такой, что $\alpha_{\zeta-1} > 0$, $\alpha_\zeta \leqslant 0$.

Положить

$$k_* = k_\zeta, \quad \sigma^0 = \sigma_{k_*}, \tag{15}$$

$$\chi_{j_i} = \begin{cases} f^*_{j_i}, & \text{если } \Delta\nabla_{j_i} < 0, \\ f_{*j_i}, & \text{если } \Delta\nabla_{j_i} > 0, \end{cases}, \quad j_i \in J_{\tau+1}, \tag{16}$$

$$\omega_{k_\tau} = \begin{cases} g^*_{k_\tau}, & \text{если } \Delta\delta_{k_\tau} > 0, \\ g_{*k_\tau}, & \text{если } \Delta\delta_{k_\tau} < 0, \end{cases}, \quad \tau \in \overline{0, \zeta - 1}. \tag{17}$$

Шаг 8. Вычислить $\overline{\beta} = \beta + \sigma^0\Delta\delta$, $\overline{\nabla} = \nabla + \sigma^0\Delta\nabla$, положить $\beta := \overline{\beta}$, $\nabla := \overline{\nabla}$, $K_{\text{оп}} := (K_{\text{оп}} \setminus \{k_0\}) \cup \{k_*\}$, $K_{\text{н}} := K \setminus K_{\text{оп}}$.

Шаг 9. Вычислить $\omega_{\text{оп}} = B^{-1}_{\text{оп}}(b - A\chi - B_{\text{н}}\omega_{\text{н}})$ и перейти к шагу 1.

Шаг 10. Положить $\alpha\alpha := \alpha\alpha + 1$. Если $\alpha\alpha > m$, то положить $\alpha\alpha := 0$, $z := -1$ и перейти к шагу 17. Иначе положить

$$k_0 := k_{\alpha\alpha} \in K_{\text{оп}} = \{k_1, k_2, \ldots, k_{\alpha\alpha}, \ldots, k_m\},$$

$\Delta\delta_{k_0} := 1$, $\Delta\delta_k := 0$, $k \in K_{\text{оп}} \setminus \{k_0\}$, $\overline{\alpha_0} := \omega_{k_0} - g^*_{k_0}$.

Шаг 11. Определить $\Delta\delta_{\text{н}}$, $\Delta\nabla$, α_0, σ_k, $k \in K^0$, ξ_j, $j \in J^0$ согласно (9)–(11). Найти $\sigma^0 = \min\limits_{k \in K^0} \sigma_k$.

Шаг 12. Если $\sigma^0 = \infty$ и $z = 1$, то перейти к шагу 10.

Шаг 13. Если $\sigma^0 = \infty$ и $z = -1$, то перейти к шагу 17.

Шаг 14. Определить α_τ, $\tau \in \overline{1, r}$, независимо от значения α_0 согласно (12)–(14).

Шаг 15. Если существует индекс $\zeta \in \overline{1, r}$ такой, что

$$\Delta\psi(\mu, s, t) = \sum_{\tau=0}^{\zeta-1} \Big(\alpha^1_\tau(\sigma_{k_{\tau+1}} - \sigma_{k_\tau}) + \\ + \sum_{j_i \in J_{\tau+1} \cap J^-} \Delta\nabla_{j_i}(\chi_{j_i} - f^*_{j_i})(\sigma_{k_{\tau+1}} - \xi_{j_i}) + \\ + \sum_{j_i \in J_{\tau+1} \cap J^+} \Delta\nabla_{j_i}(\chi_{j_i} - f_{*j_i})(\sigma_{k_{\tau+1}} - \xi_{j_i}) \Big) > 0, \tag{18}$$

то определить k_*, σ^0, χ_{j_i}, $j_i \in J_{\tau+1}$, ω_{k_τ}, $\tau \in \overline{0, \zeta - 1}$ согласно (15)–(17) и перейти к шагу 8 (α^1_τ — слагаемое α_τ, соответствующее K).

Шаг 16. Если $z = 1$, то перейти к шагу 10.

Шаг 17. Положить $\alpha\alpha := \alpha\alpha + 1$. Если $\alpha\alpha > m$, то перейти к шагу 18. Иначе положить $k_0 := k_{\alpha\alpha} \in K_{\text{оп}} = \{k_1, k_2, \ldots, k_{\alpha\alpha}, \ldots, k_m\}$, $\Delta\delta_{k_0} := -1$, $\Delta\delta_k := 0$, $k \in K_{\text{оп}} \setminus \{k_0\}$, $\overline{\alpha}_0 := -\omega_{k_0} + g_{*k_0}$ и перейти к шагу 11.

Шаг 18. Конец. Построенная опорная коситуация $\{\beta, K_{\text{оп}}\}$ удовлетворяет необходимым условиям оптимальности «высокого порядка».

Замечание. Параметры $\alpha_0, \alpha_1, \ldots, \alpha_r$ означают скорости изменения целевой функции задачи (3) (начальная и т. д.) вдоль направления $\Delta\mu' = \Delta\delta'_{\text{оп}} B_{\text{оп}}^{-1}$ изменения двойственных стратегий и коситуации.

2.2. Обоснование алгоритма.

Определение 14. Переход от одной опорной коситуации $\{\beta, K_{\text{оп}}\}$ к другой опорной коситуации $\{\overline{\beta}, \overline{K}_{\text{оп}}\}(\{\beta, K_{\text{оп}}\} \to \{\overline{\beta}, \overline{K}_{\text{оп}}\})$ назовём *итерацией.*

Определение 15. Итерацию назовём *регулярной,* если $\sigma^0 > 0$.

Теорема 3. *Если в процессе работы алгоритма встречается конечное число нерегулярных итераций, то за конечное число итераций алгоритм строит опорную коситуацию* $\{\beta, K_{\text{оп}}\}$, *удовлетворяющую необходимым условиям оптимальности «высокого порядка».*

Доказательство. Во-первых, количество опор задачи (2) не превосходит $N = C_l^m = \frac{l!}{m!(l-m)!}$. Во-вторых, при конечности нерегулярных итераций никакая опора в ходе работы алгоритма не может встречаться дважды. Таким образом, за конечное число итераций алгоритм построит опорную коситуацию $\{\beta, K_{\text{оп}}\}$, удовлетворяющую необходимым условиям оптимальности «высокого порядка». Теорема 3 доказана.

Теорема 4. *Для произвольной задачи* (1) *при любой начальной опорной коситуации* $\{\beta, K_{\text{оп}}\}$ *приведённый алгоритм с модификацией выбора индекса* k_0 *на шаге* 2 ($k_0 = \min\{k \in K_{\text{оп}} \mid \omega_k \overline{\in} [g_{*k}, g_k^*]\}$) *и индекса* k_* *при* $\sigma^0 = 0$ *на шаге* 6 ($k_* = \min K_{\text{н0}}$) *за конечное число итераций строит опорную коситуацию* $\{\beta, K_{\text{оп}}\}$, *удовлетворяющую необходимым условиям оптимальности «высокого порядка».*

Доказательство. Предположим противное утверждению теоремы. Пусть число итераций алгоритма при решении некоторой задачи (1) бесконечно. Замена опорной коситуации $\{\beta^\tau, K_{\text{оп}}^\tau\}$ опорной коситуацией $\{\beta^{\tau+1}, K_{\text{оп}}^{\tau+1}\}$ на τ-й итерации алгоритма происходит однозначно. Следовательно, в случае бесконечного числа итераций существует цикл-последовательность $\{\beta^\tau, K_{\text{оп}}^\tau\}$, $\{\beta^{\tau+1}, K_{\text{оп}}^{\tau+1}\}$, …, $\{\beta^{\tau+\gamma}, K_{\text{оп}}^{\tau+\gamma}\} = \{\beta^\tau, K_{\text{оп}}^\tau\}$, реализующаяся на последовательных итерациях алгоритма (верхний индекс обозначает номер итерации). Из описания алгоритма вытекает, что целевая функция задачи (3) не уменьшается на итерациях, т. е.

$$\psi^\tau \leqslant \psi^{\tau+1} \leqslant \ldots \leqslant \psi^{\tau+\gamma} = \psi^\tau,$$

где для краткости $\psi^v = \psi(\mu^v, s^v, t^v)$, $v \in \overline{\tau, \tau+\gamma}$. Действительно, для итерации алгоритма возможны следующие варианты последовательности

шагов: а) 1–9; б) 1–6, 8, 9; в) 1–3, (10–12)+, 14, 15, 8, 9; г) 1–3, 10, (11–13, 17)+, 11, 14, 15, 8, 9. В случае а) целевая функция задачи (3) не уменьшается, поскольку скорость её изменения вдоль соответствующих направлений неотрицательна. В случае б) значение целевой функции задачи (3) не меняется. В случаях в), г) приращения целевой функции задачи (3) положительны. Тем самым на протяжении цикла $\{\beta^\tau, K_{\text{оп}}^\tau\}$, $\{\beta^{\tau+1}, K_{\text{оп}}^{\tau+1}\}$, …, $\{\beta^{\tau+\gamma}, K_{\text{оп}}^{\tau+\gamma}\} = \{\beta^\tau, K_{\text{оп}}^\tau\}$ выполняются равенства

$$\psi^\tau = \psi^{\tau+1} = \cdots = \psi^{\tau+\gamma} = \psi^\tau, \quad \psi^v = \psi(\mu^v, s^v, t^v),\ v \in \overline{\tau, \tau+\gamma}.$$

Это возможно только в том случае, когда $\sigma^0 = 0$ начиная с τ-й итерации, следовательно,

$$(\mu^\tau, s^\tau, t^\tau) = (\mu^{\tau+1}, s^{\tau+1}, t^{\tau+1}) = \cdots = (\mu^{\tau+\gamma}, s^{\tau+\gamma}, t^{\tau+\gamma}) = (\mu^\tau, s^\tau, t^\tau),$$
$$\delta^\tau = \delta^{\tau+1} = \cdots = \delta^{\tau+\gamma} = \delta^\tau.$$

Рассмотрим последовательность итераций $\tau, \tau+1, \ldots, \tau+\gamma-1$. Следуя [12, § 6], разобьём множество K на непересекающиеся подмножества:

$$K_{\text{поп}} = \bigcap_{ii=\tau}^{\tau+\gamma-1} K_{\text{оп}}^{ii}, \quad K_{\text{пн}} = \bigcap_{ii=\tau}^{\tau+\gamma-1} K_{\text{н}}^{ii}, \quad K_{0\text{н}} = K \setminus (K_{\text{поп}} \cup K_{\text{пн}}).$$

Таким образом, $K_{\text{поп}}$ — множество индексов, которые начиная с некоторой итерации L_1 постоянно находятся в опоре; $K_{\text{пн}}$ — множество индексов, которые начиная с некоторой итерации L_2 постоянно находятся вне опоры и не выбираются в качестве k_*; $K_{0\text{н}}$ — множество индексов, которые выбираются в качестве k_0 и k_* бесконечное число раз.

Рассмотрим итерации начиная с номера $\max\{L_1, L_2\}$. Пусть $s1 = \max K_{\text{оп}}$ и p, q номера итераций цикла такие, что $k_0 = s1$ на итерации p, а $k_* = s1$ на итерации q впервые после итерации p, при этом $k_0 = r$ на итерации q. Определим знак величины

$$\begin{aligned}
\Delta\delta'^{(q)}(\omega^{(p)} - \omega^{(q)}) + \Delta\nabla'^{(q)}(\chi^{(p)} - \chi^{(q)}) &= \\
= \sum_{k\in K} \Delta\delta_k^{(q)}\big(\omega_k^{(p)} - \omega_k^{(q)}\big) + \sum_{j\in J} \Delta\nabla_j^{(q)}\big(\chi_j^{(p)} - \chi_j^{(q)}\big) &= \\
= \sum_{k\in K_{\text{поп}}\setminus\{r\}} \Delta\delta_k^{(q)}\big(\omega_k^{(p)} - \omega_k^{(q)}\big) + \sum_{k\in K_{\text{пн}}} \Delta\delta_k^{(q)}\big(\omega_k^{(p)} - \omega_k^{(q)}\big) &+ \\
+ \Delta\delta_{s1}^{(q)}\big(\omega_{s1}^{(p)} - \omega_{s1}^{(q)}\big) + \Delta\delta_r^{(q)}\big(\omega_r^{(p)} - \omega_r^{(q)}\big) &+ \\
+ \sum_{k\in(K_{\text{поп}}\setminus\{r\})\cap K_{\text{н}}^{(q)}} \Delta\delta_k^{(q)}\big(\omega_k^{(p)} - \omega_k^{(q)}\big) + \sum_{j\in J} \Delta\nabla_j^{(q)}\big(\chi_j^{(p)} - \chi_j^{(q)}\big), &
\end{aligned}$$

для чего рассмотрим знаки каждого слагаемого. Из описания алгоритма следует, что выполняются соотношения

$$\Delta\delta_k^{(ii)} = 0, \quad k \in K_{\text{поп}},\ ii \in \overline{\tau, \tau+\gamma-1},$$
$$\omega_k^{(i)} = \omega_k^{(z)}, \quad k \in K_{\text{пн}},\ i, z \in \overline{\tau, \tau+\gamma-1}.$$

Тогда с учётом (8) имеем

$$\sum_{k \in K_{\text{поп}} \setminus \{r\}} \Delta\delta_k^{(q)} \big(\omega_k^{(p)} - \omega_k^{(q)}\big) + \sum_{k \in K_{\text{пн}}} \Delta\delta_k^{(q)} \big(\omega_k^{(p)} - \omega_k^{(q)}\big) = 0,$$
$$\sum_{j \in J} \Delta\nabla_j^{(q)} \big(\chi_j^{(p)} - \chi_j^{(q)}\big) \geqslant 0.$$

Так как $s1 \in K_{\text{н}}^{(q)}$, то $g_{*s1} \leqslant \omega_{s1}^{(q)} \leqslant g_{s1}^*$. Более того, $\omega_{s1}^{(q)} \in \{g_{*s1}, g_{s1}^*\}$, поскольку на итерации $p < q$ компонента $\omega_{s1}^{(p)}$ была в опоре, а при выходе из неё приняла граничное значение и больше не изменялась в силу определения $s1$, итерации q и правила выбора k_*. Так как $s1 = k_0$ на итерации p, то $\omega_{s1}^{(p)} \notin [g_{*s1}, g_{s1}^*]$ и возможны два случая:

1) $\omega_{s1}^{(p)} < g_{*s1}$, тогда $\omega_{s1}^{(q)} = g_{*s1}$ и $\Delta\delta_{s1}^{(q)} < 0$,
2) $\omega_{s1}^{(p)} > g_{s1}^*$, тогда $\omega_{s1}^{(q)} = g_{s1}^*$ и $\Delta\delta_{s1}^{(q)} > 0$.

В обоих случаях $\Delta\delta_{s1}^{(q)} \big(\omega_{s1}^{(p)} - \omega_{s1}^{(q)}\big) > 0$.

В силу выбора k_0 на итерациях p и q имеем неравенства $g_{*r} \leqslant \omega_r^{(q)} \leqslant g_r^*$ ($r \in K_{\text{н}}^{(p)} \cap K_{\text{н}}^{(q)}$), поэтому $\Delta\delta_r^{(q)} \big(\omega_r^{(p)} - \omega_r^{(q)}\big) > 0$.

С другой стороны,

$$\begin{aligned}
\Delta\delta'^{(q)}(\omega^{(p)} - \omega^{(q)}) + \Delta\nabla'^{(q)}(\chi^{(p)} - \chi^{(q)}) &= \\
= \Delta\delta'^{(q)}\omega^{(p)} + \Delta\nabla'^{(q)}\chi^{(p)} - \Delta\delta'^{(q)}\omega^{(q)} - \Delta\nabla'^{(q)}\chi^{(q)} &= \\
= \Delta\mu'^{(q)}(A\chi^{(p)} + B\omega^{(p)}) - \Delta\mu'^{(q)}(A\chi^{(q)} + B\omega^{(q)}) &= \\
= \Delta\mu'^{(q)}b - \Delta\mu'^{(q)}b = 0.&
\end{aligned}$$

Полученное противоречие доказывает теорему. Теорема 4 доказана.

Пример. Рассмотрим задачу (1) при следующих значениях параметров [19]:

$$n = 2, \quad m = 3, \quad l = 5, \quad c = (-1; 0)', \quad f_* = (-6; -8)', \quad f^* = (2; 2)',$$
$$d = (-2; 1; 0; 0; 0)', \quad g_* = (0; 0; 0; 0; 0)', \quad g^* = (6; 6; 100; 100; 100)',$$
$$A = \begin{pmatrix} 1 & -1 \\ -1 & 2 \\ 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 1 \end{pmatrix}, \quad b = (4; 10; 2)'.$$

В качестве начальной опорной коситуации рассмотрим

$$\{\beta, K_{\text{оп}}\}, \quad \beta = (\delta, \nabla), \quad \nabla = (-1; -1)', \quad \delta = (1; 0; 0; 0; -1)',$$
$$K_{\text{оп}} = \{2, 3, 4\}, \quad \chi = (2; 2)', \quad \omega = (6; 10; 0; -8; 0)'.$$

Применим описанный алгоритм.

Итерация 1. $\{\beta, K_{\text{оп}}\} \to \{\overline{\beta}, \overline{K}_{\text{оп}}\}$:

$$\{\overline{\beta}, \overline{K}_{\text{оп}}\}, \quad \overline{\beta} = (\overline{\delta}, \overline{\nabla}), \overline{\delta} = (0; 0; 0; -1/2; -3/2)', \quad \overline{\nabla} = (-3/2; -5/2)',$$
$$\overline{K}_{\text{оп}} = \{1, 2, 3\}, \quad \overline{\chi} = (2; 2)', \quad \overline{\omega} = (2; 6; 0; 0; 0)',$$

$\overline{\beta}$ — локально оптимальная коситуация задачи (1), $(\overline{\mu}, \overline{s}, \overline{t})$ — локально оптимальная стратегия первого игрока задачи (3),

$$\overline{\mu} = (0; -1/2; -3/2)', \quad \overline{t} = (0; 0; 0; 0; 0)', \quad \overline{s} = (0; 0; 0; 1/2; 3/2)',$$

$\overline{\chi}$ — локально оптимальная стратегия первого игрока задачи (1) согласно теореме 6 [19], при этом $\varphi(\overline{\chi}) = 0$.

Итерация 2. $\{\beta, K_{\text{оп}}\} \to \{\overline{\beta}, \overline{K}_{\text{оп}}\}$:

$$\{\beta, K_{\text{оп}}\}, \quad \beta = (\delta, \nabla), \quad \delta = (0; 0; 0; -1/2; -3/2)', \quad \nabla = (-3/2; -5/2)',$$
$$K_{\text{оп}} = \{1, 2, 3\}, \quad \chi = (2; 2)', \quad \omega = (2; 6; 0; 0; 0)',$$
$$\{\overline{\beta}, \overline{K}_{\text{оп}}\}, \quad \overline{\beta} = (\overline{\delta}, \overline{\nabla}), \quad \overline{\nabla} = (3; -4)', \quad \overline{\delta} = (0; -3; 0; -2; 0)',$$
$$\overline{K}_{\text{оп}} = \{1, 3, 5\}, \quad \overline{\chi} = (-6; 2)', \quad \overline{\omega} = (0; 0; 12; 0; 12)'.$$

$\overline{\beta}$ — локально оптимальная коситуация задачи (1), $(\overline{\mu}, \overline{s}, \overline{t})$ — локально оптимальная стратегия первого игрока задачи (3),

$$\overline{\mu} = (0; -2; 0)', \quad \overline{t} = (0; 0; 0; 0; 0)', \quad \overline{s} = (0; 3; 0; 2; 0)',$$

$\overline{\chi}$ — локально оптимальная стратегия первого игрока задачи (1) согласно теореме 6 [19], при этом $\varphi(\overline{\chi}) = 6$.

Для опорной коситуации $\{\overline{\beta}, \overline{K}_{\text{оп}}\}$ условия теоремы 2 выполняются. При этом коситуация β оптимальна, в чём можно убедиться, решив задачу алгоритмом, предложенным в работе [20].

Замечания. 1. Задача (1) для параметров, приведённых в примере, имеет три локально оптимальные стратегии первого игрока: $x^1 = (2, 2)'$, $x^2 = (-6, -8)'$, $x^3 = (-6, 2)'$.

2. Локальная оптимальность стратегии первого игрока $x^1 = (2, 2)'$ при исследовании задачи прямым методом идентифицируется с помощью пакета x^1-оптимальных опор $K^p_{\text{оп}}(x^0) = \{\{1, 2, 3\}, \{1, 3, 4\}\}$ [19, теорема 4].

3. С помощью предложенного алгоритма удалось перейти от локально оптимальной стратегии первого игрока $x^1 = (2, 2)'$ к локально оптимальной стратегии $x^3 = (-6, 2)'$, при этом значение целевой функции улучшилось.

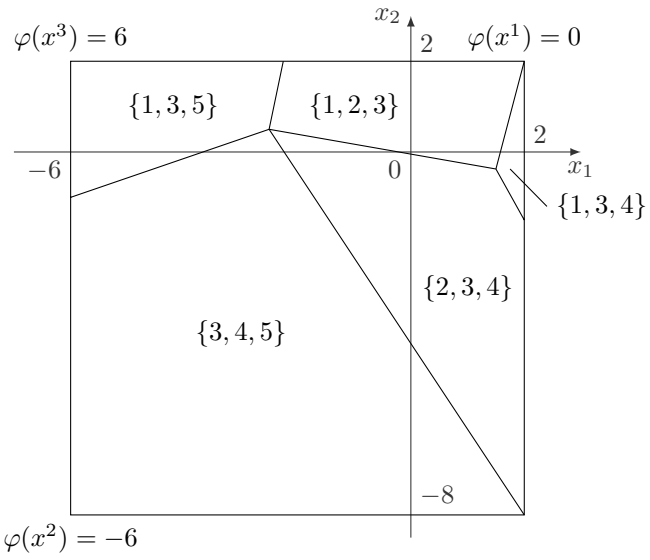


Рис. 1

4. На рис. 1 приведены локально оптимальные стратегии первого игрока и соответствующие значения целевой функции задачи (1), а также x-оптимальные опоры и соответствующие им области [19].

2.3. Численный эксперимент. Численный эксперимент поставлен на ЭВМ. Элементы задачи (1) генерировались датчиком случайных чисел. Координаты векторов c, d и элементы матриц A, B выбирались на отрезке $[-10, 10]$, координаты векторов f_*, g_* — на отрезке $[-10, 0]$, координаты векторов f^*, g^* — на отрезке $[0, 10]$. Вектор b полагался равным $b = Ax^0 + By^0$, где

$$x^0 = \frac{f_* + f^*}{2}, \quad y^0 = \frac{g_* + g^*}{2}.$$

После этого первые m компонент векторов g_*, g^* доопределялись следующим образом:

$$g_{*i} = \min_{x \in [f_*, f^*],\, y_н \in [g_{*н}, g^*_н]} h_i(b - Ax - B_н y_н),$$
$$g^*_i = \max_{x \in [f_*, f^*],\, y_н \in [g_{*н}, g^*_н]} h_i(b - Ax - B_н y_н),$$
$$i \in I = \{1, 2, \ldots, m\}, \quad K_н = \{m+1, m+2, \ldots, l\},$$

h_i — i-я строка матрицы $B_{оп}^{-1}$. Для каждой тройки (m, n, l) рассматривалась серия из трёх задач.

Таблица 1

m	n	l	$K^l_{\text{оп}}$	$K^{gl}_{\text{оп}}$	m	n	l	$K^l_{\text{оп}}$	$K^{gl}_{\text{оп}}$
2	3	5	{1,2}	{1,2}	2	3	5	{1,2}	{1,2}
2	3	5	{1,4}	{1,4}	2	3	8	{2,8}	{2,8}
2	3	8	{4,5}	{4,5}	2	3	8	{2,8}	{2,8}
3	5	8	{1,2,3}	{1,2,3}	3	5	8	{1,2,3}	{1,2,3}
3	5	8	{1,2,3}	{1,2,3}	2	10	8	{1,2}	{1,2}
2	10	8	{1,2}	{1,2}	2	10	8	{1,2}	{1,2}
3	10	8	{1,2,3}	{1,2,3}	3	10	8	{1,2,3}	{1,2,3}
3	10	8	{1,2,3}	{1,2,3}	4	10	8	{1,2,3,4}	{1,2,3,4}
4	10	8	{1,2,3,4}	{1,2,3,4}	4	10	8	{1,2}	{1,2}
2	50	8	{1,2}	{1,2}	2	50	8	{1,2}	{1,2}
2	50	8	{1,2}	{1,2}	3	50	8	{1,2,3}	{1,2,3}
3	50	8	{1,2,3}	{1,2,3}	3	50	8	{1,2,3}	{1,2,3}
3	10	15	{1,2,3}	{1,2,3}	3	10	15	{1,2,15}	{1,2,15}
3	10	15	{1,2,3}	{1,2,3}	3	10	25	{1,8,16}	{1,8,16}
3	10	25	{2,3,16}	{2,3,16}	3	10	25	{2,5,9}	{2,5,9}

Для полученных задач применён предложенный алгоритм, а также — для сравнения — алгоритм из [20]. Результаты эксперимента приведены в табл. 1, в которой приняты следующие обозначения: $K^l_{\text{оп}}$ — опора задачи (2), построенная предложенным алгоритмом, вместе с соответствующей коситуацией, удовлетворяющей условиям теоремы 2; $K^{gl}_{\text{оп}}$ — опора задачи (2), построенная алгоритмом из [20], для которой соответствующая коситуация оптимальна.

Из 30 задач в 22 случаях начальная коситуация удовлетворяла теореме 2, в восьми случаях построена опорная коситуация, удовлетворяющая теореме 2. В трёх случаях из восьми удалось перейти с локально оптимальной кострategии (локально оптимальной стратегии первого игрока) к другой кострategии (стратегии первого игрока), при которой значение целевой функции задачи (3) возрастало.

Заметим, что построенные предложенным алгоритмом кострategии оптимальны согласно алгоритму [20].

Если начальную опору взять равной $K_{\text{оп}} = \{l-m+1, l-m+2, \ldots, l\}$, то только при $m = 2$, $n = 3$, $l = 5$ (в двух случаях из трёх) не удаётся перейти к опоре $K^{gl}_{\text{оп}}$. Во всех остальных случаях эксперимента удаётся перейти к опоре $K^{gl}_{\text{оп}}$, применяя предложенный алгоритм.

Заключение

Для линейной игровой задачи с благоприятными ситуациями предложен двойственный алгоритм, который позволяет строить стратегию первого игрока, удовлетворяющую необходимым условиям оптимальности «высокого порядка». Приведён пример, иллюстрирующий работу алгоритма. Также представлены результаты численного эксперимента по построению стратегий первого игрока, удовлетворяющих необходимым условиям оптимальности «высокого порядка». Элементы этих задач генерировались датчиком случайных чисел. Из результатов численного эксперимента можно сделать вывод, что при предложенном алгоритме зачастую имеется возможность перехода от одной локально оптимальной стратегии первого игрока к другой стратегии, обеспечивающей возрастание целевой функции. Кроме того, предложенный алгоритм аналогично двойственному симплекс-алгоритму решения задачи линейного программирования [21, § 2] характеризуется

1) неубыванием целевой функции задачи (3) на итерациях, его строгим возрастанием при регулярных итерациях;

2) на итерациях $\{\beta, K_{\text{оп}}\} \to \{\overline{\beta}, \overline{K}_{\text{оп}}\}$ отсутствием перехода к опорам, для которых соответствующие значения целевой функции задачи (3) меньше $\varphi((\mu, s, t) \mid K_{\text{оп}})$;

3) максимальным возрастанием целевой функции задачи (3) при движении вдоль направления $\Delta\mu' = \Delta\delta'_{\text{оп}} B_{\text{оп}}^{-1}$ изменения двойственных стратегий и кострategии.

Алгоритм можно использовать или модифицировать как для решения игровой задачи с благоприятными ситуациями, так и для решения игровой задачи с произвольными ситуациями [13, 20], для решения игровой задачи с запрещёнными ситуациями [9, 15], а также для решения слабых задач линейного двухуровневого программирования [7, 17].

Финансирование работы

Исследование выполнено за счёт бюджета Самаркандского государственного университета им. Ш. Рашидова. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Пшеничный Б. Н.** Необходимые условия экстремума. М.: Наука, 1982. 144 с.

2. **Мордухович Б. Ш.** К необходимым условиям экстремума в негладкой оптимизации // Докл. АН СССР. 1985. Т. 283, № 4. С. 816–822.
3. **Кларк Ф.** Оптимизация и негладкий анализ. М.: Гл. ред. физ.-мат. литры, 1988. 288 с.
4. **Демьянов В. Ф., Рубинов А. М.** Основы негладкого анализа и квазидифференциальное исчисление. М.: Наука, 1990. 431 с.
5. **Гороховик В. В.** Выпуклые и негладкие задачи векторной оптимизации. Мн.: Навука і тэхніка, 1990. 239 с.
6. **Иоффе А. Д.** О необходимых условиях минимума // Фундамент. и прикл. математика. 2014. Т. 19, № 4. С. 121–152.
7. **Фёдоров В. В.** Численные методы максимина. М.: Наука, 1979. 280 с.
8. **Габасов Р., Шилкина Е. И.** Прямой точный метод решения одного класса минимаксных задач // Докл. АН БССР. 1981. Т. 25, № 11. С. 971–973.
9. **Азизов И.** Конечный алгоритм решения линейной максиминной задачи со связанными переменными и результаты численного эксперимента. Мн., 1984. 20 с. (Препр. / Ин-т математики АН БССР; № 18 (203)).
10. **Габасов Р., Кириллова Ф. М., Костина Е. А.** Глобальная максимизация специальных классов выпуклых функций на выпуклом многогранном множестве // Журн. вычисл. математики и мат. физики. 1992. Т. 32, № 8. С. 1313–1320.
11. **Гольштейн Е. Г.** Теория двойственности в математическом программировании и её приложения. М.: Наука, 1971. 352 с.
12. **Габасов Р., Кириллова Ф. М., Тятюшкин А. И.** Конструктивные методы оптимизации. Ч. 1. Линейные задачи. Мн.: Университетское, 1984. 214 с.
13. **Иванилов Ю. П.** Двойственные полуигры // Изв. АН СССР. Тех. кибернетика. 1972. № 4. С. 3–9.
14. **Гермейер Ю. Б.** Игры с непротивоположными интересами. М.: Наука, 1976. 327 с.
15. **Лозовану Д. Д.** Максиминные задачи со связанными переменными и их применения к исследованию и решению циклических игр // Изв. АН Респ. Молдова. Математика. 1990. № 2. С. 22–29.
16. **Falk J. E.** Linear max-min problem // Math. Program. 1973. V. 5, No. 2. P. 169–188. DOI: 10.1007/BF01580119.
17. **Liu J., Hong Y., Zheng Y.** A branch and bound-based algorithm for the weak linear bilevel programming problems // Wuhan Univ. J. Nat. Sci. 2018. V. 23, No. 6. P. 480–486. DOI: 10.1007/s11859-018-1352-8.
18. **Маматов А. Р.** Двойственный алгоритм вычисления локального оптимума одной максиминной задачи со связанными переменными // Узб. журн. «Пробл. информатики и энергетики». 2000. № 1. С. 7–12.
19. **Маматов А. Р.** Необходимые условия оптимальности «высокого порядка» в линейной максиминной задаче со связанными переменными // Журн. вычисл. математики и мат. физики. 2010. Т. 50, № 6. С. 1017–1022.
20. **Маматов А. Р.** Алгоритм решения одной игры двух лиц с передачей информации // Журн. вычисл. математики и мат. физики. 2006. Т. 46, № 10. С. 1784–1789.

21. Габасов Р., Кириллова Ф. М., Альсевич В. В., Калинин А. И., Крахотко В. В., Павлёнок Н. С. Методы оптимизации. Мн.: Четыре четверти, 2011. 472 с.

Маматов Акмал Равшанович

Статья поступила
16 августа 2023 г.
После доработки —
17 января 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 123–143

UDC 519.8+519.6 DOI: 10.33048/daio.2024.31.782

SEARCH FOR LOCALLY OPTIMAL STRATEGIES IN A LINEAR GAME PROBLEM WITH FAVORABLE SITUATIONS

A. R. Mamatov

Rashidov Samarkand State University,
15 University Boulevard, 140104 Samarkand, Uzbekistan

E-mail: akmm1964@rambler.ru

Abstract. A linear game problem for two players is considered. The two players alternately choose their strategies from their respective sets. First, player 1 chooses his/her strategy, then player 2, knowing the strategy of player 1, does the same. The set of strategies of player 2 depends on the strategy of player 1. The goal of player 1 is to choose a strategy to maximize a convex and piecewise linear function (the minimum function of the strategy of player 2). The goal of player 2 is to minimize the linear function. An algorithm is proposed that allows constructing strategies in this problem, as well as strategies in the dual problem, that satisfy necessary "higher-order" optimality conditions. This algorithm uses a formula for the increment of the objective function in the dual problem. Theorems that assert the finiteness of the proposed algorithm and its modification are proved. An example illustrating the operation of the algorithm is given. The results of a numerical experiment on the construction of strategies that satisfy the necessary "higher-order" optimality conditions in problems whose elements were generated by a random number generator are also presented. Based on the results of the numerical experiment, we can conclude that with the proposed algorithm, it is often possible to switch from one locally optimal strategy of player 1 to another one increasing the objective function. Tab. 1, illustr. 1, bibliogr. 21.

Keywords: linear game, maximin problem, optimality condition, support, algorithm.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 489–502 (2024), DOI: 10.1134/S1990478924030104.

References

1. **B. N. Pshenichnyi,** *Necessary Conditions for Extremum* (Nauka, Moscow, 1982) [Russian].
2. **B. Sh. Mordukhovich,** On necessary conditions for an extremum in nonsmooth optimization, *Dokl. Akad. Nauk SSSR* **283** (4), 816–822 (1985) [Russian] [*Soviet Math. Dokl.* **32**, 215–220 (1985)].
3. **F. H. Clarke,** *Optimization and Nonsmooth Analysis* (John Wiley & Sons, New York, 1983; Nauka, Moscow, 1988 [Russian]).
4. **V. F. Demyanov** and **A. M. Rubinov,** *Nonsmooth Analysis Foundations and Quasi-Differential Calculus* (Nauka, Moscow, 1990) [Russian].
5. **V. V. Gorokhovik,** *Convex and Nonsmooth Vector Optimization Problems* (Nauka Tekh., Minsk, 1990) [Russian].
6. **A. D. Ioffe,** On necessary conditions for a minimum, *Fundam. Prikl. Mat.* **19** (4), 121–152 (2014) [Russian].
7. **V. V. Fyodorov,** *Numerical Maximin Methods* (Nauka, Moscow, 1979) [Russian].
8. **R. Gabasov** and **E. I. Shilkina,** A direct exact method for solving one class of minimax problems, *Dokl. Akad. Nauk BSSR* **25** (11), 971–973 (1981) [Russian].
9. **I. Azizov,** A finite algorithm for solving a linear maximin problem with bound variables and results of a numerical experiment (Inst. Mat. AN BSSR, Minsk, 1984) (Prepr. № 18 (203)) [Russian].
10. **R. Gabasov, F. M. Kirillova,** and **E. A. Kostina,** Global maximization of special classes of convex functions on a convex polyhedral set, *Zh. Vychisl. Mat Mat. Fiz.* **32** (8), 1313–1320 (1992) [Russian] [*Comput. Math. Math. Phys.* **32** (8), 1171-1177 (1992)].
11. **E. G. Golshtein,** *Duality Theory in Mathematical Programming and Its Aplications* (Nauka, Moscow, 1971) [Russian].
12. **R. Gabasov, F. M. Kirillova,** and **A. I. Tyatyushkin,** *Constructive Optimization Methods. Pt. 1. Linear Problems* (Universitetskoe, Minsk, 1984) [Russian].
13. **Yu. P. Ivanilov,** Dual semi-games, *Izv. Akad. Nauk SSSR, Tekh. Kibern.,* No. 4, 3–9 (1972) [Russian].
14. **Yu. B. Germeier,** *Games with Non-Opposite Interests* (Nauka, Moscow, 1976) [Russian].
15. **D. D. Lozovanu,** Maximin linear problems with connected variables and their applications to the study and solution of cyclic games, *Bul. Acad. Ştiinţe Repub. Mold., Mat.,* No. 2, 22–29 (1990) [Russian].
16. **J. E. Falk,** Linear max-min problem, *Math. Program.* **5** (2), 169–188 (1973), DOI: 10.1007/BF01580119.
17. **J. Liu, Y. Hong,** and **Y. Zheng,** A branch and bound-based algorithm for the weak linear bilevel programming problems, *Wuhan Univ. J. Nat. Sci.* **23** (6), 480–486 (2018), DOI: 10.1007/s11859-018-1352-8.

18. A. R. Mamatov, A dual algorithm for computing a local optimum in a linear maximin problem with coupled variables, *Uzb. Zh. "Probl. Inform. Energ.",* No. 1, 7–12 (2000) [Russian].

19. A. R. Mamatov, High-order necessary optimality conditions in a linear maxmin problem with coupled variables, *Zh. Vychisl. Mat. Mat. Fiz.* **50** (6), 1017–1022 (2010) [Russian] [*Comput. Math. Math. Phys.* **50** (6), 963–968 (2010)].

20. A. R. Mamatov, An algorithm for solving a two-person game with information transfer, *Zh. Vychisl. Mat. Mat. Fiz.* **46** (10), 1784–1789 (2006) [Russian] [*Comput. Math. Math. Phys.* **46** (10), 1699–1704 (2006)].

21. R. Gabasov, F. M. Kirillova, V. V. Alsevich, A. I. Kalinin, V. V. Krakhotko, and **N. S. Pavlyonok,** *Optimization Methods* (Chetyre Chetverti, Minsk, 2011) [Russian].

Akmal R. Mamatov

Received August 16, 2023
Revised January 17, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 144–152

УДК 519.17 DOI: 10.33048/daio.2024.31.793

О МАКСИМАЛЬНОМ ЧИСЛЕ ОТКРЫТЫХ ТРЕУГОЛЬНИКОВ В ГРАФАХ С МАЛЫМ ЧИСЛОМ РЁБЕР

А. В. Пяткин

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: artem@math.nsc.ru

Аннотация. Подмножество из трёх вершин, порождающее подграф ровно с двумя рёбрами, будем называть открытым треугольником (ОТ). Рассматривается задача поиска графов с максимальным числом ОТ. Доказано, что в классе графов, когда число рёбер превышает число вершин на фиксированную константу, такой граф единствен при достаточно большом числе вершин. Библиогр. 11.

Ключевые слова: открытый треугольник, индуцированный подграф, разреженный граф.

Введение

Известно [1–3], что для анализа сбалансированности социальной сети, её однородности, транзитивности и склонности к кластеризации необходимо изучить её триадный перечень — число различных ориентированных трёхвершинных подграфов, встречающихся в сети. Это обуславливает интерес к более общей задаче подсчёта количества заданных порождённых подграфов в графе [4–6], а также определения графов с наибольшим числом тех или иных подграфов как в ориентированном, так и в неориентированном случаях.

В неориентированном графе минимальным по числу вершин интересным случаем является подсчёт числа трёхвершинных подграфов. Так, в работах [7, 8] доказано, что

$$\max_G(\Delta_1(G)+\Delta_2(G)) = \begin{cases} t^3-t^2 & \text{при } n=2t, \\ 8t^3+2t^2 & \text{при } n=4t+1, \\ 8t^3+14t^2+8t+1 & \text{при } n=4t+3, \end{cases} \quad (1)$$

где $\Delta_i(G)$ — число индуцированных трёхвершинных подграфов с i рёбрами в неориентированном графе G.

В [9] доказана единственность графа G с максимальным $\Delta_2(G)$ при фиксированном числе вершин n, а именно $G = K_{\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil}$, причём имеет место формула

$$\max_G \Delta_2(G) = \begin{cases} t^3 - t^2 & \text{при } n = 2t, \\ t^3 + t(t-1)/2 & \text{при } n = 2t+1. \end{cases} \tag{2}$$

Заметим, что при чётных n значения в (1) и (2) совпадают.

В [9] была также поставлена задача максимизации $\Delta_2(G)$ в графах с фиксированным числом вершин n и рёбер m. Нетрудно заметить, что при $m < n$ максимум $\Delta_2(G)$ равен $(m^2-m)/2$ и достигается на единственном графе $G = K_{1,m} \cup (n-m-1)K_1$. В [10] получена полная характеризация графов с максимальным $\Delta_2(G)$ при $n = m$. Оказалось, что за исключением случая $n = m = 6$ максимальный граф всегда единственный. В частности, при $n \geqslant 7$ экстремальный граф получается добавлением одного ребра к звезде $K_{1,n-1}$ (при этом $\Delta_2(G) = (n^2 - 3n)/2$).

В настоящей статье исследуется случай $m = n + c$. Доказано, что при $n \geqslant 4c + 7$ экстремальный граф единствен: он получается добавлением всесмежной вершины к графу $(n - c - 3)K_1 \cup K_{1,c+1}$, при этом имеет место формула $\Delta_2(G) = (n^2 - 3n + c^2 - c)/2$.

В разд. 1 приводится конструкция графов с $\Delta_2 = (n^2 - 3n + c^2 - c)/2$, а также выводится ряд вспомогательных результатов. В разд. 2 доказывается основной результат работы, а именно, что при достаточно большом n данная конструкция оптимальна. В заключении даются некоторые комментарии к работе.

1. Предварительные результаты

Следуя терминологии работы [9], будем называть *открытым треугольником* (ОТ) простой неориентированный граф, состоящий из трёх вершин и двух рёбер, т. е. ОТ — это индуцированный путь длины 2. При этом центральную вершину этого пути — вершину степени 2 — будем называть *центром* ОТ. Тогда $\Delta_2(G)$ — это в точности число ОТ в графе G. Для произвольной вершины v через $N(v)$ обозначим множество её соседей и положим $N[v] = N(v) \cup \{v\}$.

Далее считаем, что G является графом на n вершинах с $m = n + c$ рёбрами, имеющим максимальное число ОТ. Также предполагаем, что n достаточно велико, а именно, имеет место неравенство $n \geqslant 4c + 7$. Сначала приведём пример графов, на которых достигается оценка $\Delta_2(G) = (n^2 - 3n + c^2 - c)/2$.

Лемма 1. *Если граф G имеет вершину v степени $n-1$ (всесмежную вершину), то $G \setminus \{v\} = (n - c - 3)K_1 \cup K_{1,c+1}$ и имеет место формула $\Delta_2(G) = (n^2 - 3n + c^2 - c)/2$.*

Доказательство. Очевидно, что G содержит ровно $\binom{n-1}{2} - c - 1$ ОТ с центром v независимо от того, как расположены оставшиеся $c+1$ рёбер в $N(v)$. Поскольку $n-1 > c+1$, из упомянутого выше результата работы [9] следует, что максимум $\Delta_2(G \setminus \{v\})$ равен $\binom{c+1}{2}$ и достигается на графе $G \setminus \{v\} = (n-c-3)K_1 \cup K_{1,c+1}$. Имеем

$$\Delta_2(G) = \frac{(n-1)^2 - n + 1}{2} - c - 1 + \frac{(c+1)^2 - c - 1}{2} = (n^2 - 3n + c^2 - c)/2.$$

Лемма 1 доказана.

Нам также потребуются структурные свойства графов с максимальным числом ОТ, доказанные в [10].

Лемма 2 [10]**.** *Пусть G — граф с максимальным числом открытых треугольников, содержащий n вершин и m рёбер. Тогда*

1) *максимум одна компонента связности графа G может иметь рёбра;*

2) *если при $m \geqslant n$ граф G имеет k изолированных вершин и одну связную компоненту C, то либо степень всех вершин в C не меньше $k+2$, либо для некоторого единственного $d \leqslant k+1$ компонента C содержит одну или несколько вершин степени d, смежных с d попарно несмежными вершинами максимальной степени Δ, а степени всех остальных вершин в C (если они есть) лежат на отрезке $[k+2, \Delta-1]$.*

Рассмотрев в лемме 2 случай $k=1$, получим простое

Следствие 1. *Если $m \geqslant n$ и граф G имеет хотя бы одну изолированную вершину, то G не может одновременно содержать вершин степеней 1 и 2.*

Наконец, докажем следующий вспомогательный результат.

Лемма 3. *Функция $f(x) = x_1^2 + \cdots + x_n^2$ от целочисленных переменных при ограничениях $\Delta \geqslant x_1 \geqslant \ldots \geqslant x_n \geqslant 0$ и $x_1 + \cdots + x_n = 2m$ достигает максимума в точке $x_1^* = \cdots = x_t^* = \Delta$, $x_{t+1}^* = s$, $x_{t+2}^* = \cdots = x_n^* = 0$, где $0 \leqslant s \leqslant \Delta - 1$ и $\Delta t + s = 2m$.*

Доказательство. Рассмотрим оптимальное решение x^*. По условию $\Delta \geqslant x_1^* \geqslant \ldots \geqslant x_n^* \geqslant 0$. Выберем номер k так, что $x_k^* < \Delta$, но $x_i^* = \Delta$ для всех $i < k$. Аналогично пусть номер l выбран так, что $x_l^* > 0$, но $x_j^* = 0$ для всех $j > l$. Предположим, что $k < l$. Рассмотрим решение

$$x'_i = \begin{cases} x_i^* + 1 & \text{при } i = k, \\ x_i^* - 1 & \text{при } i = l, \\ x_i^* & \text{при } i \notin \{k, l\}. \end{cases}$$

Очевидно, что x' удовлетворяет требуемым ограничениям, при этом

$$f(x') - f(x^*) = (x_k^* + 1)^2 - x_k^{*2} + (x_l^* - 1)^2 - x_l^{*2} = 2(x_k^* - x_l^* + 1) > 0,$$

что противоречит оптимальности решения x^*. Значит, либо $k = l = t+1$ и $s = x^*_{t+1} > 0$, либо $k = t+1 > l = t$ и $s = 0$; в обоих случаях оптимальное решение имеет требуемый вид. Лемма 3 доказана.

2. Основной результат

Основным результатом работы является

Теорема 1. *Если граф $G = (V, E)$ удовлетворяет условиям*

$$|V| = n \geqslant 4c + 7, \quad |E| = m = n + c,$$

то

$$\Delta_2(G) \leqslant (n^2 - 3n + c^2 - c)/2,$$

причём равенство достигается на единственном графе, описанном в лемме 1.

Доказательство. Пусть v — вершина максимальной степени Δ в G. Случай $\Delta = n - 1$ разобран в лемме 1. Достаточно доказать, что при всех $\Delta \leqslant n - 2$ имеет место неравенство $\Delta_2(G) < (n^2 - 3n + c^2 - c)/2$. Рассмотрим несколько случаев.

Случай 1. Пусть $\Delta \geqslant n - 5$. Сначала оценим сверху максимальное число ОТ в графе G. Очевидно, что граф G содержит не более $\binom{\Delta}{2}$ ОТ с центром v, причём равенство достигается только в случае, когда $N(v)$ независимо. Каждое из $m - \Delta$ рёбер, неинцидентных v, может входить ровно в один ОТ с вершиной v; значит, всего имеется не более $m - \Delta$ ОТ, в которых вершина v имеет степень 1, причём равенство достигается только в случае, когда каждое из $m - \Delta$ рёбер, неинцидентных v, имеет ровно один конец в $N(v)$. Наконец, $\Delta_2(G \setminus \{v\}) \leqslant \binom{m-\Delta}{2}$, причём равенство достигается только в случае, когда граф $G \setminus \{v\}$ содержит в качестве подграфа звезду $K_{1,m-\Delta}$. Заметим, что по выбору n имеем $|N(v)| = \Delta \geqslant n - 5 \geqslant 4c + 2 \geqslant c + 5 \geqslant m - \Delta$ при $c \geqslant 1$. Следовательно, максимально возможное число ОТ достигается на графе, в котором некоторая вершина $u \in V \setminus N[v]$ смежна ровно с $m - \Delta$ вершинами из $N(v)$ (тогда во всех трёх приведённых выше верхних оценках достигается равенство). Из следствия 1 получаем, что либо в G нет изолированных вершин (тогда $\Delta = n-2$), либо в G нет вершин степени 1 (тогда $\Delta = (n + c)/2$).

В случае $\Delta = n - 2$ при $n \geqslant 4c + 7$ имеем

$$\Delta_2(G) = \binom{n-2}{2} + c + 2 + \binom{c+2}{2} = \\ = \frac{n^2 - 5n + c^2 + 5c + 12}{2} < \frac{n^2 - 3n + c^2 - c}{2}.$$

Если $\Delta = (n + \mathrm{c})/2$, то

$$\Delta_2(G) = 2\binom{(n+\mathrm{c})/2}{2} + \frac{n+c}{2} = \frac{n^2+2cn+c^2}{4} = \\ = \frac{n^2-3n+c^2-c}{2} - \frac{n^2-2cn+c^2-6n-2c}{4},$$

при этом

$$n^2-2cn+c^2-6n-2c = n(n-2c-6)+c^2-2c \geqslant (4c+7)(2c+1)+c^2-2c > 0.$$

В обоих вариантах получаем, что $\Delta_2(G) < (n^2 - 3n + c^2 - c)/2$.

Случай 2. Пусть $n - 6 \geqslant \Delta \geqslant 2n/5$. Оценим сверху $\Delta_2(G)$ так же, как и в предыдущем случае:

$$\Delta_2(G) = \binom{\Delta}{2} + n + c - \Delta + \binom{n+c-\Delta}{2} = \\ = \frac{n^2+c^2+2\Delta^2+2nc-2n\Delta-2c\Delta+n+c-2\Delta}{2} = \\ = \frac{n^2-3n+c^2-c}{2} - \frac{2n\Delta-2nc+2c\Delta-2\Delta^2-4n+2\Delta-2c}{2}.$$

Из условий $n - \Delta \geqslant 6$ и $\Delta \geqslant 2n/5$ вытекает, что

$$n\Delta - nc - 2n - \Delta^2 + c\Delta + \Delta - c = (n-\Delta)(\Delta-c-2) - \Delta - c \geqslant \\ \geqslant 5\Delta - 7c - 12 \geqslant 2n - 7c - 12 \geqslant c + 2 > 0$$

Значит, $\Delta_2(G) < (n^2 - 3n + c^2 - c)/2$.

Случай 3. Пусть $\Delta \leqslant 2n/5$. Перенумеруем вершины графа в порядке невозрастания их степеней $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_n$. Тогда в графе G может существовать не более $\binom{d_i}{2}$ ОТ с центром в i-й вершине. Поскольку $d_1 + \cdots + d_n = 2m$, получаем оценку

$$\Delta_2(G) \leqslant \sum_{i=1}^{n} \binom{d_i}{2} = \frac{f(d_1,\ldots,d_n)}{2} - m.$$

По лемме 3 имеем $f(d_1, \ldots, d_n) \leqslant t\Delta^2 + s^2 \leqslant \Delta(t\Delta + s) = 2m\Delta$, откуда

$$\Delta_2(G) \leqslant m(\Delta-1) \leqslant (n+c)\left(\frac{2n}{5} - 1\right) = \frac{2n^2+2nc-5n-5c}{5} = \\ = \frac{n^2-3n+c^2-c}{2} - \frac{n^2-4nc-5n+5c^2+5c}{10}.$$

Однако $n^2 - 4nc - 5n + 5c^2 + 5c \geqslant 2(4c+7) + 5c^2 + 5c > 0$, следовательно, $\Delta_2(G) < (n^2 - 3n + c^2 - c)/2$. Теорема 1 доказана.

Отметим, что хотя в доказательстве теоремы 1 используется условие $c \geqslant 1$, её результат остаётся верным и при $c = 0$: как следует из результата работы [10], при $m = n \geqslant 7$ максимальное число ОТ, равное $(n^2 - 3n)/2$, достигается на единственном графе, получаемом добавлением ребра к звезде $K_{1,n-1}$, что совпадает с конструкцией из леммы 1 при $c = 0$.

Заключение

В работе исследуется проблема определения максимального числа ОТ в графах с n вершинами и $m = n + c$ рёбрами, а также описания графов, на которых достигается этот максимум. Доказано, что при $n \geqslant 4c + 7$ такой граф единствен и имеет место формула $\Delta_2(G) = (n^2 - 3n + c^2 - c)/2$. В качестве открытых проблем можно предложить следующие.

1) Можно ли улучшить нижнюю оценку на n, при которой данный результат остаётся верным (скажем, до $n \geqslant 3c + 7$)?

2) Найти максимальное число ОТ в графах с линейным относительно числа вершин числом рёбер (т. е. $m = an + b$) и описать оптимальные графы.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Johnsen E. C.** Structure and process: Agreement models for friendship formation // Social Networks. 1986. V. 8. P. 257–306.
2. **Wasserman S., Faust K.** Social network analysis: Methods and applications. Cambridge, UK: Camb. Univ. Press, 1994. 852 p. DOI: 10.1017/cbo9780511815478.
3. **Moody J.** Matrix methods for calculating the triad census // Social Networks. 1998. V. 20. P. 291–299.
4. **Robins G.** A tutorial on methods for the modeling and analysis of social network data // J. Math. Psychol. 2013. V. 57. P. 261–274.
5. **Schank T., Wagner D.** Finding, counting and listing all triangles in large graphs, an experimental study // Experimental and efficient algorithms. Proc. 4th Int. Workshop (Santorini Island, Greece, May 10–13, 2005). Heidelberg: Springer, 2005. P. 606–609. (Lect. Notes Comput. Sci.; V. 3503). DOI: 10.1007/11427186_54.

6. **Milo R., Shen-Orr S., Itzkovitz S., Kashtan N., Chklovskii D., Alon U.** Network motifs: Simple building blocks of complex networks // Science. 2002. V. 298. P. 824–827.
7. **Goodman A. W.** On sets of acquaintances and strangers at any party // Am. Math. Mon. 1959. V. 66, No. 9. P. 778–783. DOI: 10.1080/00029890.1959.11989408.
8. **Sauve L.** On chromatic graphs // Am. Math. Mon. 1961. V. 68, No. 2. P. 107–111. DOI: 10.1080/00029890.1961.11989632.
9. **Pyatkin A., Lykhovyd E., Butenko S.** The maximum number of induced open triangles in graphs of a given order // Optim. Lett. 2018. V. 13, No. 8. P. 1927–1935. DOI: 10.1007/s11590-018-1330-2.
10. **Пяткин А. В., Чёрных О. И.** О максимальном числе открытых треугольников в графах с одинаковым числом вершин и рёбер // Дискрет. анализ и исслед. операций. 2022. Т. 29, № 1. С. 46–55.
11. **Batagelj V., Mrvar A.** A subquadratic triad census algorithm for large sparse networks with small maximum degree // Soc. Networks. 2001. V. 23. P. 237–243.

Пяткин Артём Валерьевич

Статья поступила
25 января 2024 г.
После доработки —
20 февраля 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 144–152

UDC 519.17 DOI: 10.33048/daio.2024.31.793

ON THE MAXIMUM NUMBER OF OPEN TRIANGLES IN GRAPHS WITH FEW EDGES

A. V. Pyatkin

Sobolev Institute of Mathematics,
4 Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: artem@math.nsc.ru

Abstract. A three-vertex subset is called an open triangle (OT) if it induces a subgraph with exactly two edges. The problem of finding graphs with maximum number of OTs is considered. It is proved that, in case of sufficiently many vertices, such a graph is unique in the class of graphs with constant difference between the numbers of edges and vertices. Bibliogr. 11.

Keywords: open triangle, induced subgraph, sparse graph.

References

1. **E. C. Johnsen,** Structure and process: Agreement models for friendship formation, *Social Networks* **8**, 257–306 (1986).
2. **S. Wasserman** and **K. Faust,** *Social Network Analysis: Methods and Applications* (Camb. Univ. Press, Cambridge, UK, 1994), DOI: 10.1017/cbo9780511815478.
3. **J. Moody,** Matrix methods for calculating the triad census, *Social Networks* **20**, 291–299 (1998).
4. **G. Robins,** A tutorial on methods for the modeling and analysis of social network data, *J. Math. Psychol.* **57**, 261–274 (2013).
5. **T. Schank** and **D. Wagner,** Finding, counting and listing all triangles in large graphs, an experimental study, in *Experimental and Efficient Algorithms* (Proc. 4th Int. Workshop, Santorini Island, Greece, May 10–13, 2005) (Springer, Heidelberg, 2005), pp. 606–609 (Lect. Notes Comput. Sci., Vol. 3503), DOI: 10.1007/11427186_54.
6. **R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii,** and **U. Alon,** Network motifs: Simple building blocks of complex networks, *Science* **298**, 824–827 (2002).

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 516–520 (2024), DOI: 10.1134/S1990478924030128.

7. **A. W. Goodman,** On sets of acquaintances and strangers at any party, *Am. Math. Mon.* **66** (9), 778–783 (1959), DOI: 10.1080/00029890.1959.11989408.
8. **L. Sauve,** On chromatic graphs, *Am. Math. Mon.* **68** (2), 107–111 (1961), DOI: 10.1080/00029890.1961.11989632.
9. **A. Pyatkin, E. Lykhovyd,** and **S. Butenko,** The maximum number of induced open triangles in graphs of a given order, *Optim. Lett.* **13** (8), 1927–1935 (2018), DOI: 10.1007/s11590-018-1330-2.
10. **A. V. Pyatkin** and **O. I. Chernykh,** On the maximum number of open triangles in graphs with the same number of vertices and edges, *Diskretn. Anal. Issled. Oper.* **29** (1), 46–55 (2022) [Russian] [*J. Appl. Ind. Math.* **16** (1), 116–121 (2022)].
11. **V. Batagelj** and **A. Mrvar,** A subquadratic triad census algorithm for large sparse networks with small maximum degree, *Soc. Networks* **23**, 237–243 (2001).

Artyom V. Pyatkin

Received January 25, 2024
Revised February 20, 2024
Accepted March 22, 2024

ДИСКРЕТНЫЙ АНАЛИЗ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

2024. Том 31, № 3

Зав. редакцией Ю. В. Шамардин

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 20.11.2024 г.
Формат $70 \times 100\,1/16$. Усл. печ. л. 12,4. Объём 1,13 МБ.

Издательство Института математики,
пр. Академика Коптюга, 4, 630090 Новосибирск, Россия