

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 31 № 4 2024

Новосибирск
Издательство Института математики

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Главный редактор **В. Л. Береснев**
Зам. главного редактора **А. А. Евдокимов**
Ответственный секретарь **Ю. В. Шамардин**

С. В. Августинovich	М. Я. Ковалёв	А. В. Пяткин
Г. П. Агибалов	А. В. Кононов	А. А. Сапоженко
В. Б. Алексеев	А. В. Косточка	М. Свириденко
О. В. Бородин	В. В. Кочергин	Б. Я. Рябко
В. А. Васильев	Ю. А. Кочетов	Н. Н. Токарева
Э. Х. Гимади	В. К. Леонтьев	Ю. А. Флеров
А. Ю. Григорьев	Б. М.-Т. Лин	Ф. В. Фомин
С. Демпе	В. В. Лозин	М. Ю. Хачай
А. И. Ерзин	П. Пардалос	Я. М. Шафранский

Учредители Сибирское отделение РАН
журнала 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, № 4 (162)	Октябрь–декабрь 2024	

СОДЕРЖАНИЕ

Береснев В. Л., Мельников А. А. Алгоритм ветвей, границ и отсечений для неявной динамической задачи конкурентного размещения	5
Гончаров Е. Н. Жадный алгоритм для задачи календарного планирования с ограниченными ресурсами	27
Ильев В. П., Ильева С. Д., Кононов А. В. Приближённые алгоритмы для задач кластеризации на графах с кластерами небольшого размера	40
Коломеец Н. А. Число невозможных разностей по модулю 2^n композиции XOR и циклического сдвига битов	58
Назаренко С. А. Уточнённые оценки для алгоритмов упаковки 2-гистограмм в полосу	90
Панин А. А., Пискеева Д. А., Плясунов А. В. Спуск с чередующимися окрестностями для поиска радиуса пороговой устойчивости в задаче размещения и дискриминационного ценообразования	116
Савенкова Н. П., Мокин А. Ю., Дряженков А. А., Артемьева Л. А. Алгоритм поиска решения задачи календарного планирования на производстве	134

НОВОСИБИРСК
ИЗДАТЕЛЬСТВО ИНСТИТУТА МАТЕМАТИКИ

Сергеев И. С. О сложности параллельных префиксных схем с ограниченным ветвлением	151
Тамасян Г. Ш., Шульга Г. С. Методы негладкого анализа в задаче минимизации суммы модулей от аффинных функций.....	168
Юськов А. Д., Кулаченко И. Н., Мельников А. А., Кочетов Ю. А. Метод декомпозиции для управления запасами в двухэтапной системе складов	186
Содержание тома 31	213

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

© Сибирское отделение РАН, 2024

© Институт математики им. С. Л. Соболева СО РАН, 2024

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

DISKRETNYYI ANALIZ I ISSLEDOVANIE OPERATSII

/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/

Published since 1994	Scientific journal	4 issues per year
Vol. 31, No. 4 (162)	October–December, 2024	

CONTENTS

V. L. Beresnev and A. A. Melnikov . <i>A branch, bound, and cuts algorithm for the dynamic competitive facility location problem</i>	5
E. N. Goncharov . <i>Greedy algorithm for the resource-constrained project scheduling problem</i>	27
V. P. Il'ev , S. D. Il'eva , and A. V. Kononov . <i>Approximation algorithms for graph clustering problems with clusters of bounded size</i>	40
N. A. Kolomeec . <i>The number of impossible additive differentials for the composition of XOR and bit rotation</i>	58
S. A. Nazarenko . <i>Updated estimates for algorithms for packing 2-bar charts in a strip</i>	90
A. A. Panin , D. A. Piskeeva , and A. V. Plyasunov . <i>Variable neighborhood descent for finding the threshold stability radius in the facility location and discriminatory pricing problem</i>	116
N. P. Savenkova , A. Yu. Mokin , A. A. Dryazhenkov , and L. A. Artemyeva . <i>An algorithm to search a solution to the production scheduling problem</i>	134
I. S. Sergeev . <i>On the complexity of fanout-bounded parallel prefix circuits</i>	151

NOVOSIBIRSK
SOBOLEV INSTITUTE PRESS

G. Sh. Tamasyan and G. S. Shulga. <i>Methods of nonsmooth analysis in application to the problem of minimizing the sum of affine functions' modules</i>	168
A. D. Yuskov, I. N. Kulachenko, A. A. Melnikov, and Yu. A. Kochetov. <i>Decomposition approach for a two echelon inventory management system</i>	186
<i>Contents of Volume 31</i>	213

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

© Siberian Branch of RAS, 2024

© Sobolev Institute of Mathematics SB RAS, 2024

АЛГОРИТМ ВЕТВЕЙ, ГРАНИЦ И ОТСЕЧЕНИЙ
ДЛЯ НЕЯВНОЙ ДИНАМИЧЕСКОЙ ЗАДАЧИ
КОНКУРЕНТНОГО РАЗМЕЩЕНИЯ

В. Л. Береснев^{1, 2, a}, А. А. Мельников^{1, 2, b}

¹ Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

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

E-mail: ^aberesnev@math.nsc.ru, ^bmelnikov@math.nsc.ru

Аннотация. Рассматривается динамическая задача конкурентного размещения, моделирующая взаимодействие двух соперничающих сторон (Лидера и Последователя), размещающих свои объекты на рассматриваемом горизонте планирования, состоящего из некоторого числа периодов времени. Предполагается, что Лидер открывает свои объекты в начале горизонта планирования и не изменяет это решение внутри горизонта планирования. При этом Последователь может корректировать своё решение в каждом периоде горизонта планирования. Предлагается алгоритм поиска наилучшего решения, построенный на основе вычислительной схемы ветвей и границ. Для вычисления верхних границ используется специальная релаксация исходной двухуровневой задачи, усиленная дополнительными ограничениями. Предлагаются процедуры построения таких ограничений, в которых используются вспомогательные оптимизационные задачи для получения наиболее сильных отсечений. На примере задачи динамического конкурентного размещения на сети с тремя вершинами демонстрируются возможности модели учитывать информацию об изменениях параметров с течением времени. Реализация алгоритма ветвей и границ демонстрирует значительный эффект от использования отсечений, предложенных специально для динамических моделей: верхняя граница вычисляется более точно и сокращается число вершин дерева ветвления. Ил. 2, библиогр. 8.

Ключевые слова: задача конкурентного размещения предприятий, двухуровневое математическое программирование, точный метод, игра Штакельберга.

Введение

Задачи конкурентного размещения относятся к семейству дискретных задач двухуровневого математического программирования, построенных на базе игры Штакельберга [1]. В таких моделях две соперничающие стороны (Лидер и Последователь) последовательно размещают свои предприятия и стремятся захватить потребителей с целью получения максимальной прибыли от их обслуживания. В случае, когда возможность изменения составов открытых сторонами предприятий в модели не предусмотрена, будем говорить о *статических* моделях конкурентного размещения. Обзор результатов исследований в области статических моделей конкурентного размещения можно найти в недавних обзорах [2, 3].

Динамические задачи конкурентного размещения образуют новый, мало изученный к настоящему моменту класс задач. Они обобщают статические задачи на случай, когда, во-первых, некоторые данные (например, множество потребителей) не фиксированы, а изменяются на рассматриваемом горизонте планирования, а во-вторых, решения сторон о составе размещаемых предприятий могут изменяться внутри горизонта планирования. Обычно предполагается, что горизонт планирования разбит на периоды (отрезки) времени, на каждом из которых внешние данные не изменяются и возможна корректировка решений, принятых сторонами ранее. Первое известное упоминание динамической задачи конкурентного размещения содержится, по-видимому, в [4]. Однако общие постановки и предложения способов для построения наилучших решений таких задач нам не известны.

В работе авторов [5] предложена постановка неявной динамической задачи конкурентного размещения. В неявных динамических задачах [6] учитываются изменения внешних факторов, влияющих на принятие решений, но сами решения о размещении предприятий, принятые в начале горизонта планирования, не корректируются в дальнейшем. Исследуемая в [5] модель является неявной динамической задачей в том смысле, что Лидер открывает свои предприятия в начале горизонта планирования и не изменяет состава своих открытых предприятий. Напротив, Последователь имеет возможность изменять состав своих предприятий в каждом периоде рассматриваемого горизонта планирования. Это задача двухуровневого математического программирования. Для её решения предлагается использовать вычислительную схему ветвей и границ и способ вычисления верхних границ значений целевой функции задачи, базирующийся на использовании специфической для двухуровневых моделей релаксации, называемой в англоязычной литературе *high-point relaxation* [7] (задачи HPR). Для усиления задачи HPR предложено использовать модифицированные *s*-отсечения [8] и новые дополнительные

ограничения, названные d-отсечениями [5], которые построены с учётом специфики переменных динамических задач.

В настоящей работе разработаны процедуры построения указанных дополнительных ограничений, в которых используются вспомогательные оптимизационные задачи для получения наиболее сильных с-отсечений и d-отсечений. Такие ограничения отсекают наибольшее число допустимых решений задачи НРР, не являющихся допустимыми решениями исходной двухуровневой задачи. В работе даётся подробное описание процедуры построения дополнительных ограничений и процедуры вычисления верхних границ, а также алгоритма поиска решения рассматриваемой задачи. Приводится иллюстрация возможностей динамической модели конкурентного размещения и демонстрируется позитивный эффект от использования d-отсечений при вычислении верхних оценок в работе алгоритма ветвей и границ.

1. Формулировка задачи

Рассматриваемая в [5] неявная динамическая задача конкурентного размещения (неявная DCompFLP) записывается как следующая дискретная задача двухуровневого математического программирования:

$$\max_{(x_i), (\chi_{ij})} \left(- \sum_{i \in I} f_i x_i + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J_t} p_{ij} \chi_{ij} \right), \quad (1)$$

$$\sum_{\tau=1}^t z_{i\tau}^0 + \sum_{k \preceq_j i} \chi_{kj} \leq 1, \quad i \in I, t \in T, j \in J_t, \quad (2)$$

$$x_i \geq \chi_{ij}, \quad i \in I, j \in J, \quad (3)$$

$$x_i, \chi_{ij} \in \{0, 1\}, \quad i \in I, j \in J, \quad (4)$$

$$(z_{it}^0), (\zeta_{ij}^0) \text{ — оптимальное решение задачи} \quad (5)$$

$$\max_{(z_{it}), (\zeta_{ij})} \left(- \sum_{t \in T} \sum_{i \in I} g_{it} z_{it} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J_t} q_{ij} \zeta_{ij} \right), \quad (6)$$

$$x_i + \sum_{t \in T} z_{it} \leq 1, \quad i \in I, \quad (7)$$

$$x_i + \sum_{k \preceq_j i} \zeta_{kj} \leq 1, \quad i \in I, j \in J, \quad (8)$$

$$\sum_{\tau=1}^t z_{i\tau} \geq \zeta_{ij}, \quad i \in I, t \in T, j \in J_t, \quad (9)$$

$$z_{it}, \zeta_{ij} \in \{0, 1\}, \quad i \in I, t \in T, j \in J. \quad (10)$$

В этой модели используются следующие обозначения:

- I — множество предприятий (мест возможного размещения предприятий);
- T — множество периодов (единичных отрезков) времени рассматриваемого горизонта планирования;
- $J_t, t \in T$ — множество потребителей в период времени t ; без ограничения общности полагаем, что $J_{t_1} \cap J_{t_2} = \emptyset$ при любых $t_1, t_2 \in T, t_1 \neq t_2$;
- $J = \bigcup_{t \in T} J_t$ — множество потребителей на всём горизонте планирования;
- $f_i, i \in I$ — фиксированные затраты на открытие Лидером предприятия i ;
- $g_{it}, i \in I, t \in T$ — фиксированные затраты на открытие Последователем предприятия i в период t ; для всякого $i \in I$ будем считать, что величины (g_{it}) убывают с ростом номера t ;
- $p_{ij}, i \in I, j \in J$ — величина дохода, получаемого Лидером от обслуживания открытым им предприятием i потребителя j ;
- $q_{ij}, i \in I, j \in J$ — величина дохода, получаемого Последователем от обслуживания открытым им предприятием i потребителя j .

В модели используются следующие двоичные переменные:

- $x_i, i \in I$, равна единице, если Лидер открывает предприятие i , и нулю в противном случае;
- $z_{it}, i \in I, t \in T$, равна единице, если Последователь открывает предприятие i в период t , и нулю в противном случае;
- $\chi_{ij}, i \in I, j \in J$, равна единице, если предприятие i , открытое Лидером, назначено для обслуживания потребителя j , и равна нулю в противном случае;
- $\zeta_{ij}, i \in I, j \in J$, равна единице, если предприятие i , открытое Последователем, назначено для обслуживания потребителя j , и нулю в противном случае.

Линейный порядок $\succeq_j$ на множестве I используется в модели для задания предпочтений потребителя $j \in J$ при выборе предприятия для его обслуживания. Для пары $i_1, i_2 \in I$ отношение $i_1 \succ_j i_2$ означает, что для потребителя $j \in J$ предприятие i_1 предпочтительнее предприятия i_2 . Обозначение $i_1 \succeq_j i_2$ означает, что $i_1 \succ_j i_2$ или $i_1 = i_2$. В модели полагаем, что для обслуживания потребителя может быть назначено любое открытое предприятие одной из сторон, которое более предпочтительно, чем любое предприятие, открытое другой стороной.

Рассматриваемая модель, как и всякая задача двухуровневого программирования, включает задачу верхнего уровня (1)–(5) (задачу Лидера) и задачу нижнего уровня (6)–(10) (задачу Последователя). Задачу (1)–(5) обозначим через $\mathcal{L}$, а задачу нижнего уровня (6)–(10) — через $\mathcal{F}$. Для задачи (1)–(10) в целом используем обозначение $(\mathcal{L}, \mathcal{F})$.

Пару (X, Z) , где $X = ((x_i), (\chi_{ij}))$, назовём *допустимым решением задачи* $(\mathcal{L}, \mathcal{F})$, порождённым двоичным вектором $x = (x_i)$, если X — допустимое решение задачи $\mathcal{L}$, а Z — оптимальное решение задачи $\mathcal{F}$. Допустимое решение (X, Z) , порождённое вектором x , назовём *пессимистическим*, если $L(X, Z) \leq L(X', Z')$ для любого допустимого решения (X', Z') , порождённого вектором x . Наилучшее пессимистическое допустимое решение назовём *пессимистическим оптимальным решением задачи* $(\mathcal{L}, \mathcal{F})$.

Пессимистическое допустимое решение, порождённое вектором x , как и в случае статической задачи [8], может быть построено по вектору x в результате решения двух оптимизационных задач. Первая из них есть задача $\mathcal{F}$, оптимальное значение целевой функции которой обозначим через F^* , а вторая — вспомогательная задача вида

$$\begin{aligned} & \min_{(z_{it}), (\zeta_{ij}), (v_i)} \sum_{j \in J} v_j, \\ & v_j \geq p_{ij} \left(x_i - \sum_{k \succ_j i} \sum_{\tau=1}^t z_{k\tau} \right), \quad i \in I, t \in T, j \in J_t, \\ & - \sum_{i \in I} \sum_{t \in T} g_{it} z_{it} + \sum_{i \in I} \sum_{j \in J} q_{ij} \zeta_{ij} \geq F^*, \\ & x_i + \sum_{t \in T} z_{it} \leq 1, \quad i \in I, \\ & x_i + \sum_{k \preceq_j i} \zeta_{kj} \leq 1, \quad i \in I, j \in J, \\ & \sum_{\tau=1}^t z_{i\tau} \geq \zeta_{ij}, \quad i \in I, t \in T, j \in J_t, \\ & z_{it}, \zeta_{ij} \in \{0, 1\}, \quad v_j \geq 0, \quad i \in I, t \in T, j \in J. \end{aligned}$$

Отметим, что возможность построения пессимистического допустимого решения по вектору x позволяет рассматривать задачу $(\mathcal{L}, \mathcal{F})$ как задачу максимизации некоторой псевдодобулевой функции $L(x)$, поэтому значение целевой функции задачи $(\mathcal{L}, \mathcal{F})$ на пессимистическом допустимом решении (X, Z) , порождённом вектором x , будем обозначать также через $L(x)$.

2. Дополнительные ограничения для задачи $(\mathcal{L}, \mathcal{F})$

Основой алгоритма ветвей и границ для поиска пессимистического оптимального решения исследуемой задачи $(\mathcal{L}, \mathcal{F})$ является процедура

вычисления верхних границ целевой функции на рассматриваемых в ходе работы алгоритма подмножествах решений. Такие подмножества будем задавать частичным двоичным вектором $y = (y_i)_{i \in I}$, где $y_i \in \{0, 1, *\}$. Частичное решение $y = (y_i)$ определяет фиксированные компоненты двоичных векторов $x = (x_i)$, порождающих пессимистические допустимые решения задачи $(\mathcal{L}, \mathcal{F})$ из рассматриваемого подмножества решений. Если $y_i = *$, то значение переменной x_i не фиксировано. Для частичного решения $y = (y_i)$ положим $I^0(y) = \{i \in I \mid y_i = 0\}$ и $I^1(y) = \{i \in I \mid y_i = 1\}$. Тогда подмножество решений задачи $(\mathcal{L}, \mathcal{F})$, задаваемое частичным решением $y = (y_i)$, представляет собой множество пессимистических допустимых решений задачи $(\mathcal{L}, \mathcal{F})$, порождённых двоичными векторами $x = (x_i)$, для которых

$$x_i = y_i, \quad i \in I^0(y) \cup I^1(y). \quad (11)$$

Задачу $(\mathcal{L}, \mathcal{F})$ с дополнительным ограничением (11) в задаче $\mathcal{L}$ будем обозначать через $(\mathcal{L}(y), \mathcal{F})$.

Основой для вычисления верхних границ задачи $(\mathcal{L}, \mathcal{F})$ может служить релаксированная задача, называемая в англоязычной литературе high-point relaxation (далее задача HPR). Эта задача получается из исходной двухуровневой задачи в результате исключения из неё требования оптимальности для переменных задачи нижнего уровня. Применительно к задаче $(\mathcal{L}(y), \mathcal{F})$ задача HPR может быть записана в следующем виде:

$$\max_{(x_i), (\chi_{ij}), (z_{it})} \left(- \sum_{i \in I} f_i x_i + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J_t} p_{ij} \chi_{ij} \right), \quad (12)$$

$$\sum_{\tau=1}^t z_{i\tau} + \sum_{k \preceq_j i} \chi_{kj} \leq 1, \quad i \in I, t \in T, j \in J_t, \quad (13)$$

$$x_i \geq \chi_{ij}, \quad i \in I, j \in J, \quad (14)$$

$$x_i + \sum_{t \in T} z_{it} \leq 1, \quad i \in I, \quad (15)$$

$$x_i = y_i, \quad i \in I^0(y) \cup I^1(y), \quad (16)$$

$$x_i, \chi_{ij}, z_{it} \in \{0, 1\}, \quad i \in I, t \in T, j \in J. \quad (17)$$

Для получения с использованием этой задачи нетривиальных верхних оценок её ограничения необходимо усилить некоторыми дополнительными ограничениями. Эти ограничения должны стимулировать переменные z_{it} принимать ненулевые значения и одновременно выполняться на пессимистических допустимых решениях задачи $(\mathcal{L}(y), \mathcal{F})$. Полученную в результате задачу назовём *усиленной оценочной задачей* (SEP) для задачи $(\mathcal{L}(y), \mathcal{F})$.

В [5] для исследуемой динамической задачи $(\mathcal{L}, \mathcal{F})$ предложены два вида дополнительных ограничений: с- и d-отсечения. Эти ограничения выполняются на пессимистических допустимых решениях, но при этом отсекают оптимальное решение задачи SEP, что позволяет улучшать точность верхней границы путём итеративного добавления новых отсечений. К сожалению, в некоторых случаях указанных дополнительных ограничений недостаточно для гарантированного отсечения оптимального решения задачи SEP, поэтому для улучшения верхней границы будем использовать также f-отсечения, предложенные в [8]. Система неравенств, образующих f-отсечение, гарантированно отсекает оптимальное решение задачи SEP, если оно не является пессимистическим оптимальным решением задачи $(\mathcal{L}(y), \mathcal{F})$.

При построении дополнительных ограничений используются следующие обозначения. Пусть $x = (x_i)_{i \in I} \neq 0$ — двоичный вектор, $J' \subseteq J$ — непустое подмножество, $k \in I$, $j \in J$. Положим

- $I^1(x) = \{i \in I \mid x_i = 1\}$,
- $I^0(x) = \{i \in I \mid x_i = 0\}$,
- $\alpha_j(x)$ — такой элемент $i' \in I^1(x)$, что $i' \succeq_j i$ для всех $i \in I^1(x)$,
- $\alpha_{J'}(x) = \{\alpha_j(x) \mid j \in J'\}$,
- $N_j(x) = \{i \in I \mid i \succ_j \alpha_j(x)\}$,
- $N_{J'}(x) = \bigcup_{j \in J'} N_j(x)$,
- $\overline{N}_j(k) = \{i \in I \mid i \succeq_j k\}$,
- $\overline{N}_{J'}(k) = \bigcup_{j \in J'} \overline{N}_j(k)$.

Пусть (X', z') , $X' = ((x'_i), (\chi'_{ij}))$, $z' = (z'_{it})$ — оптимальное решение задачи SEP, $u'_{it} = \sum_{\tau=1}^t z'_{it}$, $i \in I$, $t \in T$, и $x' = (x'_i)$, $u'_t = (u'_{it})$, $t \in T$. Для всякого $t \in T$ положим

$$J_{0t} = \begin{cases} J_t, & \text{если } u'_t \text{ — нулевой вектор,} \\ \{j \in J_t \mid \alpha_j(x') \succ_j \alpha_j(u'_t)\}, & \text{в противном случае.} \end{cases}$$

Исходные данные для построения с-отсечения решения (X', z') представляет собой номер $k \in I$ такой, что $x'_k = 0$ и $\sum_{t \in T} z'_{kt} = 0$. Рассмотрим множества $J_{0t}(k) = \{j \in J_{0t} \mid k \in N_j(x')\}$, $t \in T$. Пусть $t_1 \in T$ — наименьший номер, для которого при некотором $t_0 \in T$, $1 \leq t_0 \leq t_1$, для множества $J_0 = \bigcup_{t=t_0}^{t_1} J_{0t}(k)$ выполняются неравенства

$$\sum_{j \in J_0} p_{\alpha_j(x')j} > 0, \quad \sum_{j \in J_0} q_{kj} \geq g_{kt_0}. \quad (18)$$

Пусть для подмножества $J' \subseteq J_0$ выполняются аналогичные неравенства. Тогда неравенство

$$\sum_{i \in N_{J'}(x')} u_{it_1} \geq 1 + \sum_{i \in \alpha_{J'}(x')} (x_i - 1) - \sum_{i \in \bar{N}_{J'}(k)} x_i \quad (19)$$

есть с-отсечение решения (X', z') , сгенерированное номером k и подмножеством J' .

Исходные данные для построения d-отсечения решения (X', z') представляет собой номер $k \in I$ такой, что $z'_{kl} = 1$ для $l > 1$. Рассмотрим множества $J_{0t}(k) = \{j \in J_{0t} \mid k \in N_j(x')\}$, $t \in T$. Пусть $t_1 \in T$, $t_1 < l$ — наименьший номер, для которого при некотором $t_0 \in T$, $1 \leq t_0 \leq t_1$, существует подмножество J' множества $J_0 = \bigcup_{t=t_0}^{t_1} J_{0t}(k)$, для которого выполняются неравенства

$$\sum_{j \in J'} p_{\alpha_j(x')j} > 0, \quad \sum_{j \in J'} q_{kj} \geq g_{kt_0} - g_{kl}.$$

Тогда неравенство

$$\sum_{i \in N_{J'}(x')} u_{it_1} \geq 1 + (u_{kl} - 1) + \sum_{i \in \alpha_{J'}(x')} (x_i - 1) - \sum_{i \in \bar{N}_{J'}(k)} x_i \quad (20)$$

есть d-отсечение решения (X', Z') , сгенерированное номером k и подмножеством J' .

Неравенства (19) и (20) стимулируют величины u_{it_1} принимать ненулевые значения при двоичном векторе x' . Однако, правая часть указанных неравенств может быть равной единице и при других двоичных векторах. Число таких векторов зависит от числа элементов множеств $\alpha_{J'}(x')$ и $\bar{N}_{J'}(k)$, которые, в свою очередь, зависят от множества J' . Рассмотрим вспомогательные оптимизационные задачи, оптимальные решения которых дают множества J' , генерирующие наиболее сильные с- и d-отсечения для задачи $(\mathcal{L}(y), \mathcal{F})$.

Введём следующие обозначения:

- a_{ij} , $i \in I$, $j \in J_0$ — величина, равная единице, если $i \in \bar{N}_j(k)$, и нулю в противном случае;
- b_{ij} , $i \in I$, $j \in J_0$ — величина, равная единице, если $i = \alpha_j(x')$, и нулю в противном случае;
- c_j , $j \in J_0$ — величина, равная единице, если $\sum_{i \in I} b_{ij} p_{ij} > 0$, и нулю в противном случае;
- u_j , $j \in J_0$ — переменная, определяющая множество J' и равная единице, если $j \in J'$, и нулю в противном случае;
- w_i , $i \in I$ — переменная, определяющая множество $\alpha_{J'}(x')$, равная единице, если $i \in \alpha_{J'}(x')$, и нулю в противном случае;

• v_i , $i \in I$ — переменная, определяющая множество $\overline{N}_{J'}(k)$ и равная единице, если $i \in \overline{N}_{J'}(k)$, и нулю в противном случае.

С использованием введенных обозначений вспомогательная задача (задача APC) для выбора множества J' при построении с-отсечения решения (X', z') , сгенерированным номером k и множеством J' , записывается следующим образом:

$$\begin{aligned} \min_{(u_j), (v_i), (w_i)} \quad & \sum_{i \in I \setminus I^1(y)} w_i + \sum_{i \in I \setminus I^0(y)} v_i, \\ & \sum_{j \in J_0} q_{kj} u_j \geq g_{kt_0}, \\ & \sum_{j \in J_0} c_j u_j \geq 1, \\ & v_i \geq a_{ij} u_j, \quad i \in I, j \in J_0, \\ & w_i \geq b_{ij} u_j, \quad i \in I, j \in J_0, \\ & u_j, v_i, w_i \in \{0, 1\}, \quad i \in I, j \in J_0. \end{aligned}$$

Оптимальное решение $((u_j), (v_i), (w_i))$ этой задачи определяет подмножество $J' = \{j \in J_0 \mid u_j = 1\}$ и множества $\overline{N}_{J'}(k) = \{i \in I \mid v_i = 1\}$, $\alpha_{J'}(x') = \{i \in I \mid w_i = 1\}$. Неравенство (19), для которого множество J' получено с использованием вспомогательной задачи APC, будем называть с-отсечением решения (X', z') , сгенерированным номером k .

Вспомогательная задача (задача APD) для выбора множества J' при построении d-отсечения отличается от рассмотренной задачи APC только первым ограничением, которое в случае d-отсечения имеет вид

$$\sum_{j \in J_0} q_{kj} u_j \geq g_{kt_0} - g_{kl}.$$

Неравенство (20), для которого множество J' получено с использованием вспомогательной задачи APD, будем называть d-отсечением решения (X', z') , сгенерированным номером k .

Резюмируя, получаем, что при заданном допустимом решении (X', z') задачи $(\mathcal{L}(y), \mathcal{F})$ процедуры построения с- и d-отсечения этого решения аналогичны. При выбранном номере $k \in I$ процедура построения с-отсечения состоит из следующих этапов. Во-первых, выбор номеров $t_1 \in T$ и $t_0 \in T$, $1 \leq t_0 \leq t_1$. Далее построение по номерам t_0 и t_1 множества J_0 и проверка выполнения для этого множества условий (18). Если эти условия выполняются, то вычисляем оптимальное решение вспомогательной задачи APC и получаем искомое подмножество J' и множества $\overline{N}_{J'}(k)$ и $\alpha_{J'}(x')$. При заданном номере $k \in I$ таком, что $z_{kl} = 1$ для некоторого $l \in T$, $l > 1$, процедура построения d-отсечения начинается с выбора

номеров $t_1 \in T$ и $t_0 \in T$, $1 \leq t_0 \leq t_1 < l$. Далее строим множество J_0 и проверяем справедливость условий (14) для множества J_0 . Если условия выполняются, то используем оптимальное решение вспомогательной задачи APD для выбора подмножества J' и множеств $\bar{N}_k(x')$ и $\alpha_{J'}(x')$.

В случае, когда для оптимального решения (X', z') задачи SEP не удаётся построить с- или d-отсечения этого решения, будем использовать f-отсечения. Процедура построения такого отсечения сводится к вычислению пессимистического допустимого решения (X^0, Z^0) задачи $(\mathcal{L}, \mathcal{F})$, порождённого вектором $x' = (x'_i)$. Соответствующие дополнительные ограничения имеют вид

$$\sum_{i: x'_i=1} (1 - x_i) + \sum_{i: x'_i=0} x_i \geq z_{kt}^0 - z_{kt} \quad (21)$$

для $k \in I$, $t \in T$ таких, что $z_{kt}^0 = 1$, и вынуждают переменные z_{kt} принимать значения z_{kt}^0 , если вектор $x' = (x'_i)$ есть часть допустимого решения (X', z') .

На начальном этапе построения задачи SEP для $(\mathcal{L}(y), \mathcal{F})$, когда имеется только задача HPR и $I^1(y) \neq \emptyset$, естественно строить с-отсечения для допустимого решения (X', z') , $X' = ((y'_i), (\chi'_{ij}))$, $z' = (z'_{it})$, где $y'_i = 1$ для $i \in I^1(y)$ и $y'_i = 0$ в противном случае, а $z_{it} = 0$, $i \in I$, $t \in T$.

Исходные данные для такого дополнительного ограничения, которое назовём *начальным с-отсечением*, составляет номер $k \in I \setminus I^1(y')$. При построении с-отсечений решения (X', z') будем использовать следующие обозначения. Пусть $j \in J$, $J' \subseteq J$, $J' \neq \emptyset$. Положим

$$J_{0t}(k) = \{j \in J_t \mid k \in N_j(y')\},$$

$$\bar{N}_j(k) = \{i \in I \mid i \succeq_j k\}, \quad \bar{N}_{J'}(k) = \bigcup_{j \in J'} \bar{N}_j(k).$$

Пусть $t_1 \in T$ — наименьший номер такой, что при некотором $t_0 \in T$, $1 \leq t_0 \leq t_1$, существует подмножество $J' \subseteq J_0 = \bigcup_{t=t_0}^{t_1} J_{0t}(k)$, для которого

$$\sum_{j \in J'} p_{\alpha_j(y')j} > 0, \quad \sum_{j \in J'} q_{kj} \geq g_{kt_0}. \quad (22)$$

Тогда неравенство

$$\sum_{j \in \bar{N}_{J'}(y')} u_{it_1} \geq 1 - \sum_{i \in \bar{N}_{J'}(k)} x_i \quad (23)$$

назовём начальным с-отсечением для частичного решения $y = (y_i)$, сгенерированным номером k и подмножеством J' .

Утверждение 1. Пусть (X, Z) , $X = ((x_i), (\chi_{ij}))$, $Z = ((z_{it}), (\zeta_{ij}))$ есть пессимистическое допустимое решение задачи $(\mathcal{L}(y), \mathcal{F})$. Тогда неравенство (23) выполняется на этом решении.

ДОКАЗАТЕЛЬСТВО. Если вектор $x = (x_i)$ такой, что $x_i = 1$ для некоторого $i \in \overline{N}_{J'}(k)$, то неравенство (23) выполняется. Предположим, что $x_i = 0$ для всех $i \in \overline{N}_{J'}(k)$ и левая часть неравенства (23) равна нулю, т. е. данное неравенство нарушено. Поскольку $\alpha_j(x) \succeq_j \alpha_j(y')$ для $j \in J'$, то $u_{it_1} = 0$ для $i \in N_{J'}(x)$. В частности, имеем $u_{kt_0} = 0$. При заданном векторе x рассмотрим допустимое решение (X', Z') , где $X' = ((x_i), (\chi'_{ij}))$ и $Z' = ((z'_{it}), (\zeta'_{ij}))$ отличаются от текущего решения (X, Z) только в том, что $z'_{kt_0} = 1$, $\zeta'_{kj} = 1$ для всех $j \in J'$ и $\chi_{ij} = 0$ для всех $i \in I$ и $j \in J_t$ таких, что $t \geq t_0$ и $k \succ_j i$. В силу свойств (22) для целевых функций L и F Лидера и Последователя соответственно имеем $L(X') < L(X)$ и $F(Z') \geq F(Z)$, что противоречит тому, что (X, Z) является пессимистическим допустимым решением. Утверждение 1 доказано.

Отметим, что неравенство (23) стимулирует переменные z_{it} принимать ненулевые значения при любом допустимом решении (X', z') задачи SEP для задачи $(\mathcal{L}(y), \mathcal{F})$, если $\overline{N}_{J'}(k) \subseteq I^0(x')$. Число таких решений зависит от выбора множества J' . Аналогично случаю с-отсечения общего вида (19) рассмотрим вспомогательную задачу для поиска множества J' .

Как и в случае с-отсечений, будем использовать ранее введённые обозначения:

- a_{ij} , $i \in I$, $j \in J_0$ — величина, равная единице, если $i \in \overline{N}_j(k)$, и нулю в противном случае;
- b_{ij} , $i \in I$, $j \in J_0$ — величина, равная единице, если $i = \alpha_j(y')$, и нулю в противном случае;
- c_j , $j \in J_0$ — величина, равная единице, если $\sum_{i \in I} b_{ij} p_{ij} > 0$, и нулю в противном случае;
- u_j , $j \in J_0$ — переменная, определяющая множество J' и равная единице, если $j \in J'$, и нулю в противном случае;
- v_i , $i \in I$ — переменная, определяющая множество $\overline{N}_{J'}(k)$ и равная единице, если $i \in \overline{N}_{J'}(k)$, и нулю в противном случае.

Вспомогательная задача для выбора подмножества J' , генерирующего при заданном $k \in I$ наиболее сильное начальное с-отсечение для задачи $(\mathcal{L}(y), \mathcal{F})$ записывается следующим образом:

$$\begin{aligned} & \min_{(v_i), (u_j)} \sum_{i \in I \setminus I^0(y)} v_i, \\ & \sum_{j \in J_0} q_{kj} u_j \geq q_{kt_0}, \end{aligned}$$

$$\begin{aligned} \sum_{j \in J_0} c_j u_j &\geq 1, \\ v_i &\geq a_{ij} u_j, \quad i \in I, j \in J_0, \\ u_j, v_i &\in \{0, 1\}, \quad i \in I, j \in J. \end{aligned}$$

Неравенство (23) с множеством J' , построенное по оптимальному решению рассмотренной вспомогательной задачи, будем называть начальным с-отсечением, сгенерированным номером k .

Заметим, что в силу следующего утверждения множество $K \subseteq I$, включающее номера k , для которых удаётся построить начальные с-отсечения, может быть использовано при выборе номера $k \in I$ для построения с-отсечения решения (X', Z') задачи SEP для задачи $(\mathcal{L}(y), \mathcal{F})$.

Утверждение 2. Если для задачи $(\mathcal{L}(y), \mathcal{F})$ не существует начального с-отсечения, сгенерированного номером $k \in I$, то при любом допустимом решении (X', Z') задачи SEP не существует с-отсечения решения (X', Z') , сгенерированного номером k .

Доказательство. Действительно, поскольку для всякого $t \in T$ множество J_{0t} для начального с-отсечения включает множество J_{0t} для с-отсечения решения (X', Z') , это включение сохраняется и для множеств $J_{0t}(k)$ и J_0 . Утверждение 2 доказано.

3. Процедура вычисления верхней границы

В результате выполнения процедуры для вычисления верхней границы целевой функции задачи $(\mathcal{L}, \mathcal{F})$ на подмножестве решений, заданном частичным решением y , получаем величину $UB(y)$ верхней границы для целевой функции задачи $(\mathcal{L}(y), \mathcal{F})$ и два двоичных вектора x^0 и x' . Ненулевой вектор x^0 порождает пессимистическое допустимое решение (X^0, Z^0) задачи $(\mathcal{L}(y), \mathcal{F})$, найденное при выполнении процедуры, а нулевой вектор x^0 означает, что такого решения найти не удалось. Вектор x' — часть оптимального решения (X', z') финальной задачи SEP для задачи $(\mathcal{L}(y), \mathcal{F})$, построенной в процессе вычисления верхней границы.

При выполнении процедуры кроме заданного частичного решения $y = (y_i)$ считаем известной величину L^* , равную значению $L(x^*)$ целевой функции задачи $(\mathcal{L}, \mathcal{F})$ на наилучшем известном (рекордном) пессимистическом допустимом решении (X^*, Z^*) задачи $(\mathcal{L}, \mathcal{F})$, порождённом вектором x^* .

Если частичное решение $y = (y_i)$ представляет собой двоичный вектор, то процедура вычисления верхней границы сводится к построению пессимистического допустимого решения задачи $(\mathcal{L}, \mathcal{F})$, порождённого вектором $y = (y_i)$. В этом случае полагаем $UB(y) = L(y)$, $x^0 = y$, $x' = y$.

Если частичное решение $y = (y_i)$ не является двоичным вектором и $I^1(y) = \emptyset$, то полагаем $UB(y) = \infty$ и $x^0 = 0$. Далее последовательно выбираем некоторые $i \in I \setminus I^0(y)$ и полагаем $x'_i = 1$. Полагаем $x'_i = 0$ для остальных $i \in I$.

В общем случае, когда $y = (y_i)$ не двоичный вектор и $I^1(y) \neq \emptyset$, вычисление верхней границы есть итерационный процесс, состоящий из начальной итерации, некоторого числа основных итераций и, возможно, одной дополнительной итерации.

На начальной итерации обращаемся к HPR для задачи $(\mathcal{L}(y), \mathcal{F})$ и как результат формируем задачу SEP. Для этого при каждом $k \in I \setminus I^1(y)$ обращаемся к процедуре построения начального с-отсечения, сгенерированного номером k . Если отсечение построено, то добавляем соответствующее неравенство (23) к ограничениям задачи SEP, а номер k включаем в множество номеров K , генерирующих начальные с-отсечения.

На каждой основной итерации рассматриваем задачу SEP, полученную на предыдущей итерации, и вычисляем её оптимальное решение (X', z') . Если для значения $L(X', z')$ целевой функции задачи SEP на полученном решении (X', z') выполняется неравенство $L(X', z') \leq L^*$, то полагаем $UB = L(X', z')$, $x^0 = 0$ и завершаем процедуру вычисления верхней границы. В противном случае начинаем построение дополнительных ограничений, отсекающих решение (X', z') . Для этого последовательно перебираем номера $k \in K$ и выполняем процедуру построения с-отсечения. Если для некоторого k соответствующее с-отсечение построено, то полученное неравенство добавляем к ограничениям задачи SEP и начинаем следующую основную итерацию. В противном случае начинаем процесс построения d-отсечения решения (X', z') . Для этого поочерёдно просматриваем номера $k \in I$ такие, что $z'_{kl} = 1$ для некоторого $l \in T$, $l > 1$, и для выбранного k выполняем процедуру построения d-отсечения, сгенерированного номером k . Если процедура построения d-отсечения для рассматриваемого k завершается успешно, то полученное неравенство (20) добавляем к ограничениям задачи SEP и начинаем следующую основную итерацию.

Если отсечений решения (X', z') построить не удалось, и при этом $I^1(x') \neq I^1(y)$, то полагаем $UB(y) = L(X', z')$, $x^0 = 0$ и завершаем процедуру вычисления верхней границы. Если $I^1(x') = I^1(y)$, то переходим к выполнению дополнительной итерации.

На дополнительной итерации используем f-отсечение решения (X', z') . Для этого полагаем $x^0 = x'$ и вычисляем пессимистическое допустимое решение (X^0, Z^0) задачи $(\mathcal{L}(y), \mathcal{F})$, порождённое вектором x^0 . Добавляем соответствующие неравенства к ограничениям задачи SEP и вычисляем новое оптимальное решение (X', z') задачи SEP. Полагаем $UB(y) = L(X', z')$ и завершаем процедуру вычисления верхней границы.

Заметим, что, если для нового оптимального решения (X', z') получаем $x' = x^0$, то построенное допустимое решение (X^0, Z^0) есть пессимистическое оптимальное решение задачи $(\mathcal{L}(y), \mathcal{F})$. Если же $x' \neq x^0$, то $I^1(x') \neq I^1(y)$.

4. Алгоритм вычисления пессимистического оптимального решения

Для вычисления пессимистического оптимального решения исследуемой задачи $(\mathcal{L}, \mathcal{F})$ предлагается алгоритм, построенный на основе вычислительной схемы ветвей и границ со способом ветвления — поиск в глубину. Алгоритм включает начальный шаг и некоторое число основных шагов.

На начальном шаге готовятся начальные данные, необходимые для основных шагов алгоритма. На основном шаге рассматривается *упорядоченное частичное решение*, включающее частичное решение $y = (y_i)$ и вектор порядка $(i_1, \dots, i_q)$, $q = |I^0(y) \cup I^1(y)|$, показывающий порядок задания значений 0 и 1 элементам частичного решения. На основном шаге с использованием верхней границы проверяется, содержит ли рассматриваемое подмножество решение лучше известного рекордного решения.

На начальном шаге строим исходное рекордное решение задачи $(\mathcal{L}, \mathcal{F})$. Для этого выбираем двоичный вектор x^* и далее строим пессимистическое допустимое решение (X^*, Z^*) , порождённое вектором x^* , а также вычисляем значение $L(x^*)$ целевой функции задачи $(\mathcal{L}, \mathcal{F})$ на решении (X^*, Z^*) . На этом шаге строим также начальное упорядоченное частичное решение. Полагаем $y_i = 1$, если $x_i^* = 1$, и $y_i = *$, если $x_i^* = 0$. Далее полагаем $q = 0$ и для всякого $i \in I$ такого, что $y_i = 1$, последовательно полагаем $q = q + 1$, $i_q = i$.

На каждом основном шаге имеем упорядоченное частичное решение $y = (y_i)$, $(i_1, \dots, i_q)$, $q = |I^0(y) \cup I^1(y)|$. Шаг начинается с вычисления верхней границы $UB(y)$ и сопутствующих векторов x^0 и x' , $I^1(x') \neq I^1(y)$. Если $x^0 \neq 0$ и $L(x^0) > L(x^*)$, то полагаем $x^* = x^0$, $L^* = L(x^0)$. Далее строим упорядоченное частичное решение, которое будем рассматривать на следующем шаге.

Если $UB(y) \leq L^*$, то отбрасываем подмножество решений, задаваемое рассматриваемым частичным решением, и производим «движение вверх» по дереву частичных решений. Для этого находим наибольший номер k , $1 \leq k \leq q$, для которого $y_{i_k} = 1$. Если такого номера найти не удаётся, то алгоритм завершает работу, а вектор x^* порождает пессимистическое оптимальное решение (X^*, Z^*) задачи $(\mathcal{L}, \mathcal{F})$. В противном случае полагаем $y_{i_k} = 0$; $y_{i_l} = *$, $l = k + 1, \dots, q$; $q = k$ и начинаем следующий основной шаг.

Если $UB(y) > L^*$, то рассматриваемое подмножество решений отбросить не удаётся и производим «движение вниз» по дереву частичных решений. Поскольку $I^1(y) \neq I^1(x')$, для каждого $i \in I^1(x') \setminus I^1(y)$ последовательно полагаем $y_i = 1$, $q = q + 1$, $i_q = i$ и начинаем следующий основной шаг.

5. Демонстрационный пример

Материал данного раздела призван продемонстрировать особенности рассматриваемой модели конкурентного размещения, а также детали работы алгоритма поиска точного решения в рамках данной модели. Для наглядности и простоты анализа детально рассмотрим демонстрационный пример, который позволит проявиться важным эффектам, составляющим практическое воплощение теоретических результатов настоящей работы, и опишем указанные эффекты.

На рис. 1 в качестве основного примера представлена ситуация конкурентного размещения предприятий на сети из трёх вершин. Потребители и места возможного размещения предприятий располагаются в вершинах сети. Предпочтения потребителей задаются расстояниями между вершинами: чем меньше расстояние до предприятия, тем оно предпочтительней для потребителя.

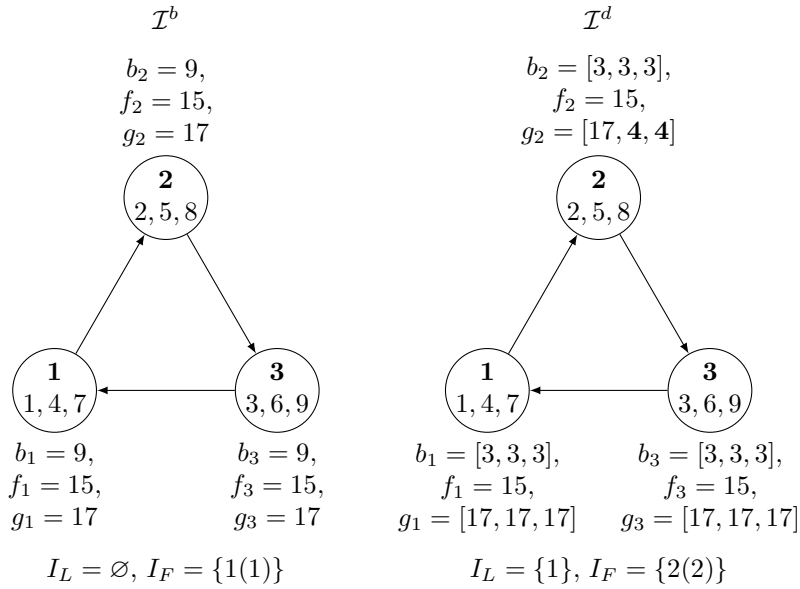


Рис. 1. Входные данные примера: слева — базовый пример $\mathcal{T}^b$, справа — динамический пример $\mathcal{T}^d$

Для демонстрации дополнительных возможностей модели динамического конкурентного размещения по отношению к базовой модели конкурентного размещения рассмотрим следующие входные данные $\mathcal{I}^b$ для базовой модели: $I = J = \{1, 2, 3\}$, $f_i = 15$, $g_i = 17$ для всех $i \in I$, $p_{ij} = q_{ij} = 9$ для всех $i \in I, j \in J$. С учётом того, что предпочтения потребителей задаются кратчайшими расстояниями в сети, имеем $1 \succ_1 2 \succ_1 3$, $2 \succ_2 3 \succ_2 1$ и $3 \succ_3 1 \succ_3 2$. Несложно видеть, что оптимальное решение базового примера $\mathcal{I}^b$ нулевое, поскольку общий бюджет потребителей равен 27, и открытие двух и более предприятий оказывается убыточным. При открытии единственного предприятия с номером $i_L \in I$ оптимальным ответом Последователя будет открытие предприятия с номером $i_F = i_L - 1$ при $i_L > 1$ и $i_F = 3$ при $i_L = 1$. При этом значение прибыли Лидера будет равно -6 , а следовательно, все ненулевые решения Лидера убыточны.

Предположим, что в дополнение к информации, содержащейся в примере $\mathcal{I}^b$, также известно, что через определённое время, равное трети продолжительности обозреваемого горизонта планирования, стоимость открытия предприятия Последователя $i = 2$ снизится и составит 4 единицы. Указанное обстоятельство можно учесть, рассмотрев динамическую модель размещения с горизонтом планирования, разделённым на три равных по длительности периода $T = \{1, 2, 3\}$. Как и ранее, положим $I = \{1, 2, 3\}$, $f_i = 15$, $g_{it} = 17$ для всех $i \in I, t \in T$ за исключением $g_{22} = g_{23} = 4$. Для каждого периода времени заведём отдельный индекс для каждого из трёх потребителей: $J_1 = \{1, 2, 3\}$, $J_2 = \{4, 5, 6\}$, $J_3 = \{7, 8, 9\}$. При этом будем считать, что потребитель, располагающийся в вершине 1, в зависимости от периода времени имеет индекс 1, 4 или 7 соответственно. Бюджет потребителей из базового примера $\mathcal{I}^b$ разделим между периодами. Предполагая, что доход от обслуживания потребителей поступает равномерно с течением времени, и учитывая, что длительности периодов равны, положим $p_{ij} = q_{ij} = 3$ для всех $i \in I, j \in J$.

Полученный таким образом пример динамической задачи конкурентного размещения $\mathcal{I}^d$ может рассматриваться как уточнение статичного примера $\mathcal{I}^b$, причём указанное уточнение оказывается существенным: в отличие от примера $\mathcal{I}^b$ у примера $\mathcal{I}^d$ оптимальное решение ненулевое. Действительно, при $x^* = (1, 0, 0)$ несложно убедиться, что вектор z^* такой, что z_{it}^* равняется единице при $i = 2, t = 2$ и нулю в противном случае, является оптимальным решением задачи Последователя. При этом $L(x^*) = -f_1 + p_{11} + p_{21} + p_{31} + p_{41} + p_{61} + p_{71} + p_{91} = -15 + 3 \cdot 7 = 6$. Таким образом, динамическая модель расширяет возможности базовой, позволяя учитывать дополнительную информацию о параметрах среды, в которой Лидер принимает решение.

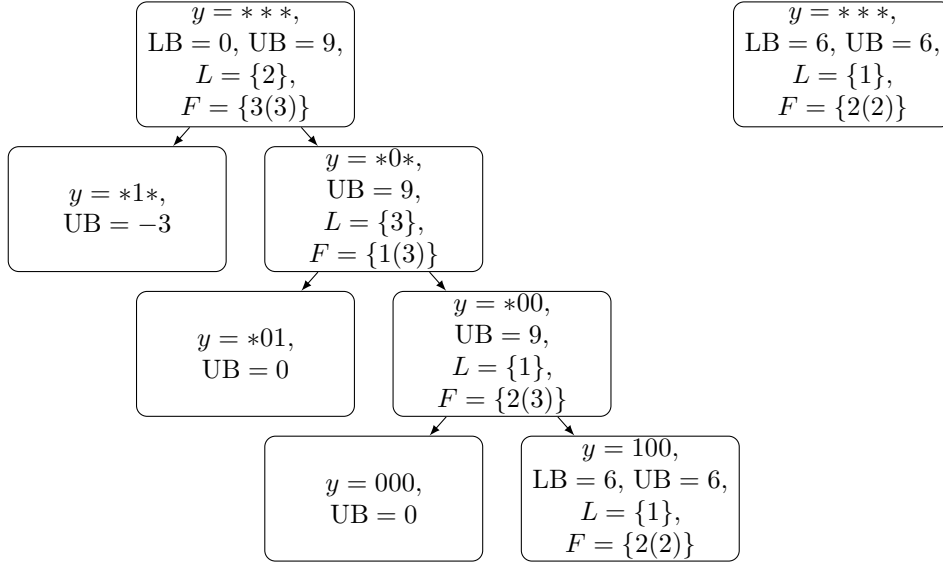


Рис. 2. Дерево ветвления: слева — при использовании только с- и f-отсечений, справа — с дополнительным использованием d-отсечений

Работа алгоритма ветвей, границ и отсечений на примере $\mathcal{I}^d$ представлена в виде схемы на рис. 2. Поле y показывает значение компонент частичного решения в текущей вершине; LB и UB — значение нижней и верхней границ соответственно, если они вычислены; L и F — наборы открытых предприятий Лидера и Последователя соответственно, выбранных в оптимальном решении SEP (в скобках для предприятий Последователя указан момент открытия). Алгоритм запускается в двух конфигурациях: только с с- и f-отсечениями, а также дополнительно вместе с d-отсечениями.

При запуске в конфигурации без d-отсечений в корневой вершине генерируется три с-отсечения:

$$\sum_{i \in \{1,2\}} \sum_{t=1}^3 z_{it} \geq 1 + (x_3 - 1) - x_1 - x_2, \quad (c1)$$

$$\sum_{i \in \{1,3\}} \sum_{t=1}^3 z_{it} \geq 1 + (x_2 - 1) - x_1 - x_3, \quad (c2)$$

$$\sum_{i \in \{2,3\}} \sum_{t=1}^3 z_{it} \geq 1 + (x_1 - 1) - x_2 - x_3. \quad (c3)$$

При этом видим, что сгенерированные отсечения позволяют присваивать ненулевые значения переменным нижнего уровня с большим значением индекса t . Если пользоваться интерпретацией указанных переменных, то можно сказать, что в релаксации двухуровневой задачи, усиленной с-отсечениями, допустимой остаётся возможность открытия предприятий Последователя в насколько возможно поздний период времени. Это создаёт дополнительную возможность для увеличения разрыва между оптимумами исходной двухуровневой задачи и её релаксации: значение верхней границы, вычисленное в корневой вершине, оказывается равным 9 при том, что оптимум равен 6.

Во время запуска с использованием d-отсечений в корневой вершине генерируется девять отсечений: пять с-отсечений и четыре d-отсечения, имеющие вид

$$\sum_{i \in \{1,3\}} \sum_{t=1}^1 z_{it} \geq 1 + \left(\sum_{t=1}^3 z_{3t} - 1 \right) + (x_2 - 1) - x_1 - x_3, \quad (d1)$$

$$\sum_{i \in \{2,3\}} \sum_{t=1}^2 z_{it} \geq 1 + \left(\sum_{t=1}^3 z_{2t} - 1 \right) + (x_1 - 1) - x_2 - x_3, \quad (d2)$$

$$\sum_{i \in \{1\}} \sum_{t=1}^1 z_{it} \geq 1 + \left(\sum_{t=1}^3 z_{1t} - 1 \right) + (x_2 - 1) - x_1, \quad (d3)$$

$$\sum_{i \in \{3\}} \sum_{t=1}^1 z_{it} \geq 1 + \left(\sum_{t=1}^3 z_{3t} - 1 \right) + (x_1 - 1) - x_3. \quad (d4)$$

В совокупности с двумя с-отсечениями, сгенерированными в дополнение к (с1)–(с3), которые здесь приводить не будем, отсечения (d1)–(d4) позволяют найти оптимальное решение исходной двухуровневой задачи, решая её усиленную релаксацию. В частности, видим, что решение с ненулевыми компонентами $x_1 = z_{23} = 1$, возникающее как оптимальное решение релаксации в дереве ветвления с с-отсечениями в вершине с $y = *00$, отсекается неравенством (d2). Это приводит к получению более точной верхней оценки и сокращению числа вершин в дереве ветвления. При этом безусловно следует отметить увеличение времени, затрачиваемого на обработку каждой индивидуальной вершины дерева, вызванное ростом числа итераций в процедуре вычисления верхних границ. Режим использования дополнительных отсечений, позволяющий получать максимальную производительность метода ветвей, границ и отсечений в целом, является предметом для будущей исследовательской работы.

Заключение

В настоящей работе рассмотрен вариант двухуровневой задачи конкурентного размещения предприятий, получивший в литературе название неявной динамической задачи. В таком варианте предполагается, что параметры спроса меняются с течением времени дискретно, и первая сторона — Лидер — принимает решение об открытии предприятий в самом начале горизонта планирования, а вторая сторона — Последователь — может открывать свои предприятия, зная решение Лидера, в любом периоде времени. Для неявной динамической задачи вводится новое семейство дополнительных отсечений, позволяющих дополнительно усилить релаксацию двухуровневой модели. Также представлена процедура генерации отсечений из указанного семейства, вовлекающих наименьшее количество переменных и обладающих в этом смысле максимальной общностью.

Для демонстрации особенностей рассматриваемой модели и предлагаемых процедур рассмотрен пример динамической задачи конкурентного размещения на сети с тремя вершинами. Отмечены возможности модели учитывать информацию о будущих изменениях значений параметров спроса, позволяющие получать с её помощью более эффективные решения. Анализ хода вычислительного процесса поиска оптимального решения рассмотренного примера показывает востребованность использования отсечений предлагаемого в работе семейства для повышения точности верхней границы и сокращения числа вершин в дереве ветвления.

Дальнейшая работа в области изучения динамических моделей конкурентного размещения будет направлена на снятие ограничивающих предположений относительно решения Лидера. На основе алгоритма для такой более общей постановки планируется построение процедуры поддержки решений на неограниченном временном горизонте.

Финансирование работы

Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 23-21-00082). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Stackelberg H.** The theory of the market economy. Oxford: Oxford Univ. Press, 1952. 289 p.

2. Mishra M., Singh S. P., Gupta M. P. Location of competitive facilities: A comprehensive review and future research agenda // *Benchmarking*. 2022. V. 30, No. 4. P. 1171–1230. DOI: 10.1108/BIJ-11-2021-0638.
3. Eiselt H. A., Drezner Z. Competitive location models: A review // *Eur. J. Oper. Res.* 2024. V. 316, No. 1. P. 5–18. DOI: 10.1016/j.ejor.2023.10.030.
4. Santos-Penate D. R., Suarez-Vega R., Dorta-Gonzalez P. The leader — follower location model // *Netw. Spat. Econ.* 2007. V. 7. P. 45–61. DOI: 10.1007/s11067-006-9007-2.
5. Береснев В. Л., Мельников А. А. Дополнительные ограничения для динамической задачи конкурентного размещения // *Дискрет. анализ и исслед. операций*. 2023. Т. 30, № 3. С. 43–56. DOI: 10.33048/daio.2023.30.774.
6. Current J., Ratick S., ReVelle C. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach // *Eur. J. Oper. Res.* 1998. V. 110, No. 3. P. 597–609. DOI: 10.1016/S0377-2217(97)00303-2.
7. Dempe S. Bilevel optimization: Theory, algorithms, applications and a bibliography // *Bilevel optimization: Advances and next challenges*. Cham: Springer, 2020. P. 581–672. DOI: 10.1007/978-3-030-52119-6_20.
8. Beresnev V. L., Melnikov A. A. Approximation of the competitive facility location problem with MIPs // *Comput. Oper. Res.* 2019. V. 104. P. 139–148. DOI: 10.1016/j.cor.2018.12.010.

Береснев Владимир Леонидович
Мельников Андрей Андреевич

Статья поступила
18 октября 2024 г.
После доработки —
25 октября 2024 г.
Принята к публикации
1 ноября 2024 г.

A BRANCH, BOUND, AND CUTS ALGORITHM FOR THE DYNAMIC COMPETITIVE FACILITY LOCATION PROBLEM

V. L. Beresnev^{1, 2, a} and A. A. Melnikov^{1, 2, b}

¹ Sobolev Institute of Mathematics,
4 Acad. Koptug Avenue, 630090 Novosibirsk, Russia

² Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia

E-mail: ^aberesnev@math.nsc.ru, ^bmelnikov@math.nsc.ru

Abstract. We consider a dynamic competitive facility location problem modeling an interaction of two competing parties (Leader and Follower) who place their facilities within a planning horizon split into several time periods. The Leader is assumed to open his/her facilities at the beginning of the planning horizon and does not change his/her decision later, while the Follower can modify his/her choice within each time period. We propose an algorithm that computes the best Leader's decision and is built on the base of the branch-and-bound computational scheme. To compute upper bounds, a special relaxation of the initial bilevel problem strengthened with additional constraints (cuts) is used. The paper describes the construction of these constraints while utilizing auxiliary optimization problems; this provides the strongest cuts. On an instance of a dynamic competitive facility location on a network with three vertices, we demonstrate that the model is capable to take into account information regarding the changes of problem's parameters along the time period. An implementation of the branch-and-bound algorithm shows a significant benefit from using the cuts specially designed for dynamic competitive models: it improves the upper bound's quality and reduces the number of nodes in the branching tree. Illustr. 2, bibliogr. 8.

Keywords: competitive facility location problem, bilevel mathematical programming, exact method, Stackelberg game.

References

1. **H. Stackelberg**, *The Theory of the Market Economy* (Oxford Univ. Press, Oxford, 1952).
2. **M. Mishra, S. P. Singh, and M. P. Gupta**, Location of competitive facilities: A comprehensive review and future research agenda, *Benchmarking* **30** (4), 1171–1230 (2022), DOI: 10.1108/BIJ-11-2021-0638.
3. **H. A. Eiselt and Z. Drezner**, Competitive location models: A review, *Eur. J. Oper. Res.* **316** (1), 5–18 (2024), DOI: 10.1016/j.ejor.2023.10.030.
4. **D. R. Santos-Penate, R. Suarez-Vega, and P. Dorta-Gonzalez**, The leader — follower location model, *Netw. Spat. Econ.* **7**, 45–61 (2007), DOI: 10.1007/s11067-006-9007-2.
5. **V. L. Beresnev and A. A. Melnikov**, Additional constraints for dynamic competitive facility location problem, *Diskretn. Anal. Issled. Oper.* **30** (3), 43–56 (2023), DOI: 10.33048/daio.2023.30.774 [Russian] [*J. Appl. Ind. Math.* **17** (3), 483–490 (2023), DOI: 10.1134/S199047892303002X].
6. **J. Current, S. Ratick, and C. ReVelle**, Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach, *Eur. J. Oper. Res.* **110** (3), 597–609 (1998), DOI: 10.1016/S0377-2217(97)00303-2.
7. **S. Dempe**, Bilevel optimization: Theory, algorithms, applications and a bibliography, in *Bilevel Optimization: Advances and Next Challenges* (Springer, Cham, 2020), pp. 581–672, DOI: 10.1007/978-3-030-52119-6_20.
8. **V. L. Beresnev and A. A. Melnikov**, Approximation of the competitive facility location problem with MIPs, *Comput. Oper. Res.* **104**, 139–148 (2019), DOI: 10.1016/j.cor.2018.12.010.

Vladimir L. Beresnev
Andrey A. Melnikov

Received October 18, 2024
Revised October 25, 2024
Accepted November 1, 2024

ЖАДНЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С ОГРАНИЧЕННЫМИ РЕСУРСАМИ

Е. Н. Гончаров

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

E-mail: gon@math.nsc.ru

Аннотация. Задача календарного планирования с ограниченными ресурсами — достаточно общая задача теории расписаний, которая включает в себя ограничения предшествования работ и ресурсные ограничения. Все ресурсы возобновимы, прерывания работ запрещены. Эта задача NP-трудна в сильном смысле. Мы предлагаем новый детерминированный жадный алгоритм. Он основан на эвристике, использующей информацию, полученную в результате решения релаксированной задачи с кумулятивными ресурсами. Алгоритм протестирован на стандартных наборах данных j60, j90 и j120, предоставляемых библиотекой PSPLIB; показана его эффективность. Табл. 2, библиогр. 17.

Ключевые слова: задача календарного планирования с ограниченными ресурсами, управление проектами, возобновимые ресурсы, жадный алгоритм, PSPLIB.

Введение

Рассматривается задача календарного планирования с ограниченными ресурсами (resource-constrained project scheduling problem, RCPSP). Её можно определить как задачу комбинаторной оптимизации. Дано множество работ, частичный порядок на этом множестве задаётся ациклическим ориентированным графом. Для каждой работы известны длительность её выполнения, множество потребляемых ею ресурсов и их объёмы. Считаем, что в каждый единичный интервал горизонта планирования $\hat{T}$ для каждого ресурса выделяется одинаковое его количество. Предполагается, что ресурсы неограниченны за пределами горизонта проекта $\hat{T}$. Все ресурсы возобновимы. Прерывания работ не допускаются. Задача состоит в минимизации общего времени выполнения

проекта с учётом отношения предшествования работ и выполнения ресурсных ограничений. По классификации, предложенной в [1], эта задача обозначается через $PS \mid prec \mid C_{\max}$, а согласно классификационной схеме из [2] — через $m, 1 \mid cpm \mid C_{\max}$.

RCPSP принадлежит к классу NP-трудных задач [3]. Она имеет обширное применение на практике, поэтому привлекает к себе внимание многих исследователей на протяжении нескольких последних десятилетий. Об этом свидетельствуют многочисленные обзоры по методам её решения, например [4–8].

Эвристики, основанные на правилах приоритета (priority rule based heuristics), являются сравнительно хорошо исследованными методами решения RCPSP. Они относятся к так называемым методам построения. Считается, что более глубокие методы, такие как алгоритмы локального поиска, генетические алгоритмы и др., обеспечивают в большинстве случаев решения лучше, чем методы построения, поскольку они выполняют процедуру поиска, стартуя от некоторого допустимого расписания, сгенерированного одним или несколькими методами построения. Сильная сторона методов построения состоит в том, что они позволяют получать сравнительно хорошие решения с небольшой временной сложностью.

Правило приоритета — это функция, которая присваивает каждой работе j некоторое значение $v(j)$. В случае ничьей используется правило разрешения ничьей. Правила приоритета можно классифицировать по разным критериям. В зависимости от типа информации, используемой для расчёта $v(j)$, можно различать правила, основанные на сети, времени и ресурсах, а также их комбинациях. Правило, основанное на сети, использует множество всех (прямых и косвенных) последователей работы j , а основанное на времени правило использует время начала выполнения работы j в некотором оценочном расписании. В [9] была рассмотрена эвристика, которая использует наиболее поздние моменты старта работ в задаче без ресурсных ограничений (эта эвристика была рассмотрена в совокупности с дополнительными правилами). В [10] при построении стохастического жадного алгоритма в качестве $v(j)$ использовались моменты начала выполнения работы j в оценочном расписании, являющемся решением релаксированной задачи, в которой ограниченные ресурсы складированы (кумулятивны). Для разрешения ничьей было использовано дополнительное ресурсное правило.

Ресурсное правило ориентировано на величину потребления ресурсов: приоритет отдаётся работам, наиболее полно расходующим выделенные ресурсы. Подробная классификация известных в литературе правил приоритета приведена в обзоре [4].

В этой статье предлагается новое правило приоритета, представляющее собой ресурсное правило, в котором $v(j)$ находится с учётом весовых

коэффициентов ресурсов. Эти коэффициенты вычисляются из решения вспомогательной релаксированной задачи, в которой ресурсы складированы. В релаксированной задаче все ресурсы ранжируются по степени дефицитности, и тем самым определяются более дефицитные из них. После этого делается предположение, что ранжирование ресурсов, выполненное для релаксированной задачи, будет также справедливым и для исходной задачи. Приписываем для каждого типа ресурса вес в соответствии с его рангом и находим $v(j)$ уже в соответствии со взвешенными значениями потребляемых этой работой ресурсов.

Качество алгоритма проверено на тестовых примерах из библиотеки PSPLIB [11]. Для тестирования взяты примеры из j60, j90 и j120 размерности 60, 90 и 120 работ соответственно. Далее приводятся результаты численных экспериментов. Показано, что с использованием предложенного алгоритма получаются лучшие значения средних отклонений решений от величины критического пути в сравнении с алгоритмами, использующими правила, основанные на свойствах сети, на времени и неранжированных ресурсах. Выявлены группы примеров из PSPLIB, на которых предложенный алгоритм имеет особо значительное преимущество.

1. Постановка задачи

Задача RCPSP может быть сформулирована следующим образом. Будем рассматривать проект как ориентированный ациклический граф $G = (N, A)$, где N — множество работ проекта, а A — множество пар работ, между которыми установлено отношение предшествования. Обозначим через $N = \{1, \dots, n\} \cup \{0, n + 1\}$ множество работ, в котором работы 0 и $n + 1$ фиктивны: они задают начало и завершение всего комплекса работ. Все другие работы будем называть фактическими. Отношение предшествования на множестве N зададим множеством пар $A = \{(i, j) \in N^2 \mid i \text{ — предшественник } j\}$. Если $(i, j) \in A$, то работа j не может начаться раньше завершения работы i . В множестве A содержатся все пары $(0, j)$ и $(j, n + 1)$, $j = 1, \dots, n$.

Через K обозначим множество типов ресурсов, необходимых для выполнения работ. Горизонт планирования $\hat{T}$ делится на временные интервалы равной длины $[t - 1, t)$, $t = 1, \dots, \hat{T}$. Все ресурсы предполагаются возобновимыми, т. е. в каждый единичный временной интервал горизонта планирования $\hat{T}$ для каждого типа ресурсов выделяется фиксированный объём этого ресурса $R_k \in Z^+$.

Фактическая работа j имеет детерминированную длительность своего выполнения $d_j \in Z^+$ и требует в каждую единицу времени своего выполнения $r_{jk} \geq 0$ единиц ресурса типа $k \in K$. Чтобы избежать неразрешимость задачи будем считать, что $r_{jk} \leq R_k$, $j \in N$, $k \in K$. Фиктивные

работы 0 и $n+1$ имеют нулевую длительность и нулевое потребление ресурсов. Фактические работы имеют ненулевую целочисленную длительность.

Введём переменные задачи. Через $s_j \geq 0$ обозначим момент начала выполнения j -й работы. Так как работы выполняются без прерывания, момент окончания j -й работы определяется равенством $c_j = s_j + d_j$. Определим расписание S как $(n+2)$ -вектор $(s_0, \dots, s_{n+1})$. Время завершения проекта $T(S)$ соответствует моменту завершения последней работы проекта, т. е. $T(S) = c_{n+1}$. Через $J(t) = \{j \in N \mid s_j \leq t < c_j\}$ обозначим множество работ, выполняемых в единичном интервале времени $[t-1, t)$ при расписании S . Задача заключается в нахождении допустимого расписания $S = \{s_j\}$ с минимальным временем завершения проекта $T(S)$. Её можно формализовать следующим образом: минимизировать время завершения проекта

$$T(S) = \max_{j \in N} (s_j + p_j) \quad (1)$$

при ограничениях

$$s_i + p_i \leq s_j, \quad i, j \in A, \quad (2)$$

$$\sum_{j \in J(t)} r_{jk} \leq R_k, \quad k \in K, t = 1, \dots, \hat{T}, \quad (3)$$

$$s_j \in \mathbf{Z}^+, \quad j \in N. \quad (4)$$

Неравенства (2) определяют ограничения предшествования работ. Соотношение (3) обеспечивает соблюдение ресурсных ограничений с возобновимыми ресурсами: суммарное количество ресурса типа k , потребляемое всеми работами, выполняемыми в единичном интервале времени $[t-1, t)$, не должно превышать имеющегося в наличии количества этого ресурса в данный момент времени. Наконец, (4) определяет искомые переменные.

Рассмотрим также релаксированную задачу для (1)–(4), в которой условие возобновимости ресурсов ослаблено до условия их складированности: минимизировать время завершения проекта

$$T(S) = \max_{j \in N} (s_j + p_j) \quad (5)$$

при ограничениях

$$s_i + p_i \leq s_j, \quad i, j \in A, \quad (6)$$

$$\sum_{t'=1}^t \sum_{j \in J(t')} r_{jk} \leq \sum_{t'=1}^t R_k, \quad k \in K, t = 1, \dots, \hat{T}, \quad (7)$$

$$s_j \in \mathbf{Z}^+, \quad j \in N. \quad (8)$$

Соотношения (7) обеспечивают соблюдение ресурсных ограничений со складываемыми ресурсами, а задача (5)–(8) является задачей календарного планирования со складываемыми (кумулятивными) ресурсами.

Отметим, что RCPSP с ограничениями на ресурсы складываемого типа (7) по трудоёмкости решения отличается от задачи с ограничениями с возобновимыми ресурсами (3). Если все ресурсы только складываемого типа, то для довольно широкого класса задач (с целочисленными длительностями работ) задача полиномиально разрешима [12].

Помимо полиномиального алгоритма [12] для решения RCPSP известен приближённый асимптотически точный алгоритм, который применим для более общей постановки задачи. В этой постановке ресурсные ограничения только складываемого типа, моменты начала выполнения работ могут быть неотрицательными вещественными числами, а также могут учитываться директивные сроки выполнения событий. Время работы этого алгоритма есть функция от числа работ N порядка $O(N^2)$, а относительная и абсолютная погрешности стремятся к нулю с ростом размерности задачи [13, 14]. Для приближённого решения задачи (5)–(8) в этой статье будем использовать именно этот алгоритм.

2. Алгоритм решения задачи

Для построения эвристического решения задачи RCPSP будем использовать известную последовательную схему генерации расписаний (serial schedule generation scheme, или serial SGS) [4].

Алгоритм S-SGS состоит в точности из N шагов, на каждом из которых выбирается ровно одна работа из числа нерассмотренных, и для неё определяется (назначается) минимальное время её старта, при этом соблюдаются ограничения предшествования работ и ресурсные ограничения. На каждом шаге $g = 1, \dots, N$ известны множество уже рассмотренных работ S_g и D_g — подмножество множества не рассмотренных работ таких, что все непосредственно предшествующие им работы принадлежат множеству S_g . Руководствуясь некоторым правилом выбора $v(j)$, выбираем работу j из множества D_g и назначаем минимальное время начала её выполнения такое, что выполнены ограничения предшествования работ и ресурсные ограничения.

Порядок выбора работ играет определяющую роль при построении жадного расписания. Правило выбора наиболее приоритетной работы из множества D_g будем определять, исходя из минимального значения функции приоритетов (весовой функции) $v(j)$, $j \in D_g$. В [4] приведён обзор описанных в литературе функций приоритетов. Они могут использовать, например, структуру сети, время свершения событий, интенсивность потребления ресурсов и т. п.

Основное новшество алгоритма заключается в применяемом правиле нахождения значения $v(j)$. Это правило базируется на интенсивности потребления работой $v(j)$ ресурсов, а также на степени относительной дефицитности ресурсов. Представляем степень дефицитности ресурсов в виде весовой функции ресурсов w_k , $k \in K$. Значения w_k назначаем некоторым образом (см. разд. 3), исходя из рангов ресурсов по степени их дефицитности. Ранжирование ресурсов производим по результатам решения релаксированной задачи: чем меньше величина неиспользованного остатка ресурса в релаксированной задаче, тем он дефицитнее.

Алгоритм $\mathcal{A}_{\text{rk-res}}$

ШАГ 1. Решаем задачу (5)–(8) быстрым приближённым асимптотически точным алгоритмом [13, 14] и находим оценочное расписание S^{est} и $\hat{R}_k$, $k \in K$, — неиспользованный остаток ресурсов в момент его завершения. Ранжируем ресурсы по неубыванию $\hat{R}_k$: ресурс с меньшим значением $\hat{R}_k$ будем считать более дефицитным. Для удобства перенумеруем ресурсы от наиболее дефицитного ресурса к наименее дефицитному, который получает последний номер $|K|$.

ШАГ 2. Назначим каждому типу ресурсов вес w_k , $k \in K$, в соответствии с произведённым ранжированием: $w_1 \geq w_2 \geq \dots \geq w_{|K|}$. Подбор значений w_k , $k \in K$, может осуществляться различными способами, а численный эксперимент показал, что для разных групп примеров из PSPLIB выбор наиболее подходящих этих значений неоднозначен. Для каждой работы j определяем её весовую функцию

$$v(j) = - \sum_{k \in K} w_k r_{jk}, \quad j \in N.$$

ШАГ 3. Положим $s_j := 0$, $j \in N$, $g := 1$, $S_1 := \{1\}$, $D_1 := \{j \in N \mid (1, j) \in A\}$, $\tilde{R}_t^k := R_k$, $k \in M$, $t \in [1, \hat{T}]$.

ШАГ 4. Выбираем из множества D_g работу с минимальным значением $v(j)$. Пусть $j \in D_g$ — выбранная работа, $te_j = \max\{s_i + p_i \mid (i, j) \in A\}$. Находим для этой работы момент начала её выполнения, учитывая ограничения предшествования и ресурсные ограничения

$$s_j := \min \{t \geq te_j \mid r_{jk} \leq \tilde{R}_\tau^k, k \in M, \tau \in [t, t + p_j]\}.$$

Полагаем $g := g + 1$. Добавляем работу j к множеству рассмотренных работ $S_g := S_{g-1} \cup \{j\}$. Пересчитываем $\tilde{R}_\tau^k := \tilde{R}_\tau^k - r_{jk}$, $k \in M$, $\tau \in [t, t + p_j]$.

ШАГ 5. Если $g < n$, то идём на ШАГ 4, иначе СТОП.

В результате работы алгоритма получаем приближённое решение исходной задачи — расписание s_j . Это решение можно попробовать улучшить, применяя к нему алгоритм локального улучшения решений, известный как алгоритм Forward-backward improvement (FBI) [15].

3. Численные эксперименты

Численные эксперименты проведены на компьютере с операционной системой Windows 7. Алгоритмы реализованы на языке C++ в системе Visual Studio. Для оценки производительности предложенного алгоритма использованы тестовые примеры из электронной библиотеки PSPLIB, представленной в [11]. В частности, рассмотрены наборы тестовых примеров j60, j90 и j120. Множества j60 и j90 содержат примеры с 60 и 90 работами в каждом соответственно, по 48 серий примеров в каждом множестве, по 10 экземпляров в каждой серии — всего по 480 примеров в каждом множестве. Набор данных j120 содержит 60 серий тестовых примеров, по 10 экземпляров в каждой серии — всего 600 экземпляров. Эти примеры содержатся в электронной библиотеке PSPLIB, доступной по ссылке www.om-db.wi.tum.de/psplib/data.html.

Представленный алгоритм $\mathcal{A}_{\text{rk-res}}$ тестируем в сравнении с тремя другими алгоритмами, построенными на трёх базовых функциях предпочтения $v(j)$. Это алгоритм $\mathcal{A}_{\text{time}}$, функция предпочтения $v(j)$ которого принимает значение времени начала работ в оценочном расписании S^{est} ; алгоритм $\mathcal{A}_{\text{net}}$, в качестве $v(j)$ берём число работ (со знаком минус), непосредственно следующих за работой j в графе G , а в качестве разрешения ничьей используем время начала выполнения работы j в оценочном расписании S^{est} ; и, наконец, алгоритм $\mathcal{A}_{\text{res}}$, в котором в качестве $v(j)$ используется информация об объёмах использования работой j ресурсов без учёта их ранжирования (со знаком минус).

Для запуска алгоритма $\mathcal{A}_{\text{rk-res}}$ необходимо определить применяемые в нём весовые коэффициенты ресурсов w_k , $k \in K$, в соответствии с их ранжированием, произведённым ранее. Все примеры из PSPLIB используют 4 типа ресурсов. В ходе проведения численных экспериментов опробованы несколько вариантов w_k . Оказалось, что для разных серий примеров хорошие решения получаются на разных весах ресурсов и что не существует единого доминирующего набора весов. Выделены следующие три варианта задания весов: $w = (1; 0,8; 0,6; 0,4)$, $w = (1; 0,9; 0,8; 0,7)$ и $w_k = 2 - \frac{\hat{R}_k}{\hat{R}_4}$, $k \in K$. Первые два варианта являются линейными монотонно убывающими функциями, а в третьем уменьшение значений w_k происходит пропорционально величинам остатков неиспользованных ресурсов в релаксированной задаче. В качестве решения будем брать лучшее из этих трёх вариантов.

Показателем качества решений для сравнения эвристических алгоритмов для задачи RCPSP принято считать среднее процентное отклонение (average percent deviation, APD) получаемых решений от их нижних границ, в качестве которых используется величина критического пути в графе G [16]. В табл. 1 приводятся полученные значения APD для алгоритмов, вошедших в тестирование, и для трёх множеств примеров из PSPLIB.

Таблица 1

**Средние отклонения решений (APD, %)
от длины критического пути для тестовых
примеров из PSPLIB**

Алгоритм	Множество тестовых примеров из PSPLIB		
	j60	j90	j120
$\mathcal{A}_{\text{time}}$	15,44	16,16	42,7
$\mathcal{A}_{\text{net}}$	16,29	14,87	40,04
$\mathcal{A}_{\text{res}}$	17,01	15,68	41,7
$\mathcal{A}_{\text{rk-res}}$	14,91	14,03	38,29

Из табл. 1 следует, что алгоритм $\mathcal{A}_{\text{rk-res}}$ показал лучшее APD по сравнению с алгоритмом $\mathcal{A}_{\text{res}}$ с ресурсной функцией предпочтения без учёта ранжирования ресурсов для всех трёх множеств примеров j60, j90 и j120. Более того, он лучший в сравнении со всеми алгоритмами, участвующими в численном эксперименте.

В табл. 1 показаны результаты, полученные на всех сериях примеров множеств j60, j90 и j120. Было бы интересно выделить те серии, на которых новый алгоритм работает особенно хорошо.

Построение каждого отдельного примера в PSPLIB определяют три параметра: сложность сети (NC), коэффициент ресурсов (RF) и ресурсный потенциал (RS). Параметр NC определяет среднее число предшественников для каждой работы, RF устанавливает средний процент количества типов ресурсов, необходимых для выполнения каждой работы, а RS устанавливает средний процент объёма выделяемых ресурсов в каждый момент времени. Нулевое значение RS соответствует минимальной потребности каждого типа ресурсов для выполнения всех работ, в то время как значение RS, равное единице, соответствует случаю неограниченных ресурсов. Значения параметров, использованные для создания примеров из наборов j60 и j90, следующие: $\text{NC} \in \{1,5; 1,8; 2,1\}$, $\text{RF} \in \{0,25; 0,5; 0,75; 1\}$ и $\text{RS} \in \{0,2; 0,3; 0,4; 0,5\}$. Известно [17], что значения параметров $\text{RF} = 1$, $\text{RS} = 0,2$ соответствуют достаточно сложным сериям. В множестве примеров j60 и j90 такими сериями являются j6013, j6029, j6045 и j9013, j9029, j9045 соответственно.

Значения параметров для создания примеров набора j120 следующие: $NC \in \{1,5; 1,8; 2,1\}$, $RF \in \{0,25; 0,5; 0,75; 1\}$ и $RS \in \{0,1; 0,2; 0,3; 0,4; 0,5\}$. Наиболее трудными сериями в j120 являются примеры j12016, j12036, j12056. Перечисленные примеры наиболее трудны, т. е. имеют наибольший средний процент отклонения APD между полученными решениями и длиной критического пути. Более подробно описание процесса создания примеров можно найти в [16].

Произведём сравнение предложенного алгоритма с известными ранее на сериях трудных примеров для каждого множества примеров j60, j90 и j120. В табл. 2 приведены значения APD для наборов трудных серий примеров для всех рассматриваемых алгоритмов.

Таблица 2

**Средние отклонения решений (APD, %)
от длины критического пути для трудных тестовых
серий примеров из PSPLIB**

Алгоритм	Множество тестовых примеров из PSPLIB		
	j60 серии 13, 29, 45	j90 серии 13, 29, 45	j120 серии 16, 36, 56
$\mathcal{A}_{\text{time}}$	92,37	88,70	180,98
$\mathcal{A}_{\text{net}}$	92,32	83,96	184,96
$\mathcal{A}_{\text{res}}$	96,17	87,89	208,29
$\mathcal{A}_{\text{rk-res}}$	89,92	82,73	172,64

Как видно из табл. 2, алгоритм $\mathcal{A}_{\text{rk-res}}$ снова превзошёл по APD как алгоритм с ресурсной функцией предпочтения без учёта ранжирования ресурсов $\mathcal{A}_{\text{res}}$, так и все другие рассматриваемые алгоритмы. Причём его преимущество над другими алгоритмами более значительно. Отметим также рост этого преимущества с увеличением размерности тестовых примеров.

Таким образом, предложенный алгоритм имеет лучшие значения APD среди тестируемых алгоритмов на всех примерах из библиотеки PSPLIB, но его преимущество особенно велико для трудных примеров — тех, в которых задействовано максимально большое количество ресурсов, а объёмы выделяемых ресурсов минимальны. Наблюдение выглядит логичным: чем больше объём выделяемых ресурсов, тем сравнительно меньший эффект имеет ранжирование ресурсов, и наоборот, чем большее количество ресурсов задействовано, тем этот эффект значительнее.

Заключение

В работе предложен детерминированный жадный алгоритм для решения задачи календарного планирования с ограниченными ресурсами

в соответствии с критерием минимизации времени выполнения всего проекта. Он основан на новом правиле приоритета работ. Это правило использует эвристику, учитывающую степень критичности (дефицитности) ресурсов, которую получаем из решения релаксированной задачи с ограничением на складываемые (кумулятивные) ресурсы. Проведены численные эксперименты на примерах из электронной библиотеки PSPLIB. Результаты вычислительных экспериментов показывают, что предлагаемый алгоритм является конкурентоспособной эвристикой. Выявлены серии примеров из PSPLIB, на которых предложенный алгоритм показал себя особенно хорошо.

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. Brucker P., Drexel A., Möhring R., Neumann K., Pesch E. Resource-constrained project scheduling: Notation, classification, models, and methods // Eur. J. Oper. Res. 1999. V. 112, No. 1. P. 3–41.
2. Herroelen W., Demeulemeester E., De Reyck B. A classification scheme for project scheduling // Project scheduling: Recent models, algorithms and applications. Dordrecht: Kluwer Acad. Publ., 1999. P. 1–26.
3. Blażewicz J., Lenstra J. K., Rinnooy Kan A. H. G. Scheduling subject to resource constraints: Classification and complexity // Discrete Appl. Math. 1983. V. 5, No. 1. P. 11–24.
4. Kolisch R., Hartmann S. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis // Project scheduling: Recent models, algorithms and applications. Dordrecht: Kluwer Acad. Publ., 1999. P. 147–178.
5. Pellerin R., Perrier N., Berthaut F. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem // Eur. J. Oper. Res. 2020. V. 280, No. 2. P. 395–416.
6. Abdolshah M. A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions // Int. Trans. J. Eng. Manage. Appl. Sci. Technol. 2014. V. 5, No. 4. P. 253–286.
7. Vanhoucke M. Resource-constrained project scheduling // Project management with dynamic scheduling. Heidelberg: Springer, 2012. P. 107–137.

8. **Hartmann S., Briskorn D.** A survey of variants and extensions of the resource-constrained project scheduling problem // *Eur. J. Oper. Res.* 2010. V. 207, No. 1. P. 1–14.
9. **Кочетов Ю. А., Столяр А. А.** Новые жадные эвристики для задачи календарного планирования с ограниченными ресурсами // *Дискрет. анализ и исслед. операций.* Сер. 2. 2005. Т. 12, № 1. С. 12–36.
10. **Гончаров Е. Н.** Стохастический жадный алгоритм для задачи календарного планирования с ограниченными ресурсами // *Дискрет. анализ и исслед. операций.* 2014. Т. 21, № 3. С. 10–23.
11. **Kolisch R., Sprecher A.** PSPLIB — A project scheduling problem library // *Eur. J. Oper. Res.* 1996. V. 96, No. 1. P. 205–216.
12. **Гимади Э. Х., Залюбовский В. В., Севастьянов С. В.** Полиномиальная разрешимость задач календарного планирования со складываемыми ресурсами и директивными сроками // *Дискрет. анализ и исслед. операций.* 2000. Сер. 2. Т. 7, № 1. С. 9–34.
13. **Гимади Э. Х.** О некоторых математических моделях и методах планирования крупномасштабных проектов // *Модели и методы оптимизации.* Тр. АН СССР. Сиб. отд-ние. Ин-т математики. Вып. 10. Новосибирск: Наука, 1988. С. 89–115.
14. **Gimadi Eh. Kh., Sevastianov S. V.** On solvability of the project scheduling problem with accumulative resources of an arbitrary sign // *Operations research proceedings 2002. Sel. Pap. Int. Conf. Operations Research (Klagenfurt, Austria, Sept. 2–5, 2002).* Heidelberg: Springer, 2003. P. 241–246.
15. **Li K. Y., Willis R. J.** An iterative scheduling technique for resource-constrained project scheduling // *Eur. J. Oper. Res.* 1992. V. 56, No. 3. P. 370–379.
16. **Kolisch R., Sprecher A., Drexel A.** Characterization and generation of a general class of resource-constrained project scheduling problems // *INFORMS Manage. Sci.* 1995. V. 41. P. 1693–1703.
17. **Project scheduling: Recent models, algorithms and applications.** Dordrecht: Kluwer Acad. Publ., 1999. 546 p.

Гончаров Евгений Николаевич

Статья поступила

21 февраля 2024 г.

После доработки —

27 апреля 2024 г.

Принята к публикации

22 июня 2024 г.

A GREEDY ALGORITHM FOR THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

E. N. Goncharov

Sobolev Institute of Mathematics,
4 Akad. Koptuyug Avenue, 630090 Novosibirsk, Russia
E-mail: gon@math.nsc.ru

Abstract. The resource-constrained project scheduling problem (briefly RCPSP) is a general scheduling problem that includes precedence and resource constraints. Activities preemptions are not allowed. Resources are renewable and there is a unique way to perform the activities. The problem with renewable resources is NP-hard in the strong sense. We propose a new deterministic greedy algorithm. It is based on heuristics that use information obtained from a relaxing problem. The algorithm is tested with standard data sets given by Kolisch library PSPLIB for j60, j90, and j120 and found to be performing well. Tab. 2, bibliogr. 17.

Keywords: resource-constrained project scheduling problem, project management, renewable resources, greedy algorithm, PSPLIB.

References

1. **P. Brucker, A. Drexler, R. Möhring, K. Neumann, and E. Pesch**, Resource-constrained project scheduling: Notation, classification, models, and methods, *Eur. J. Oper. Res.* **112** (1), 3–41 (1999).
2. **W. Herroelen, E. Demeulemeester, and B. De Reyck**, A classification scheme for project scheduling, in *Project Scheduling: Recent Models, Algorithms and Applications* (Kluwer Acad. Publ., Dordrecht, 1999), pp. 1–26.
3. **J. Blażewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan**, Scheduling subject to resource constraints: Classification and complexity, *Discrete Appl. Math.* **5** (1), 11–24 (1983).
4. **R. Kolisch and S. Hartmann**, Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis, in *Project Scheduling: Recent Models, Algorithms and Applications* (Kluwer Acad. Publ., Dordrecht, 1999), pp. 147–178.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (4), 679–685 (2024), DOI: 10.1134/S1990478924040057.

5. **R. Pellerin, N. Perrier, and F. Berthaut**, A survey of hybrid metaheuristics for the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **280** (2), 395–416 (2020).
6. **M. Abdolshah**, A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions, *Int. Trans. J. Eng. Manage. Appl. Sci. Technol.* **5** (4), 253–286 (2014).
7. **M. Vanhoucke**, Resource-constrained project scheduling, in *Project Management with Dynamic Scheduling* (Springer, Heidelberg, 2012), pp. 107–137.
8. **S. Hartmann and D. Briskorn**, A survey of variants and extensions of the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **207** (1), 1–14 (2010).
9. **Yu. A. Kochetov and A. A. Stolyar**, New greedy heuristics for the scheduling problem with constrained resources, *Diskretn. Anal. Issled. Oper., Ser. 2*, **12** (1), 12–36 (2005) [Russian].
10. **E. N. Goncharov**, A stochastic greedy algorithm for the resource-constrained project scheduling problem, *Diskretn. Anal. Issled. Oper.* **21** (3), 10–23 (2014) [Russian].
11. **R. Kolisch and A. Sprecher**, PSPLIB — A project scheduling problem library, *Eur. J. Oper. Res.* **96** (1), 205–216 (1996).
12. **Eh. Kh. Gimadi, V. V. Zalyubovskii, and S. V. Sevastianov**, Polynomial solvability of scheduling problems with storable resources and directive deadlines, *Diskretn. Anal. Issled. Oper., Ser. 2*, **7** (1), 9–34 (2000) [Russian].
13. **Eh. Kh. Gimadi**, Some mathematical models and methods for scheduling large-scale projects, in *Models and Methods of Optimization* (Trudy AN SSSR, Sib. Otd., Inst. Mat., Vol. 10) (Nauka, Novosibirsk, 1988), pp. 89–115 [Russian].
14. **Eh. Kh. Gimadi and S. V. Sevastianov**, On solvability of the project scheduling problem with accumulative resources of an arbitrary sign, in *Operations Research Proceedings 2002* (Sel. Pap. Int. Conf. Operations Research, Klagenfurt, Austria, Sept. 2–5, 2002) (Springer, Heidelberg, 2003), pp. 241–246.
15. **K. Y. Li and R. J. Willis**, An iterative scheduling technique for resource-constrained project scheduling, *Eur. J. Oper. Res.* **56** (3), 370–379 (1992).
16. **R. Kolisch, A. Sprecher, and A. Drexel**, Characterization and generation of a general class of resource-constrained project scheduling problems, *INFORMS Manage. Sci.* **41**, 1693–1703 (1995).
17. **Project Scheduling: Recent Models, Algorithms and Applications** (Kluwer Acad. Publ., Dordrecht, 1999).

Evgenii N. Goncharov

Received February 21, 2024

Revised April 27, 2024

Accepted June 22, 2024

ПРИБЛИЖЁННЫЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧ КЛАСТЕРИЗАЦИИ НА ГРАФАХ С КЛАСТЕРАМИ НЕБОЛЬШОГО РАЗМЕРА

В. П. Ильев^{1,2, a}, С. Д. Ильева², А. В. Кононов^{1, b}

¹ Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

² Омский гос. университет им. Ф. М. Достоевского,
пр. Мира, 55-А, 644077 Омск, Россия

E-mail: ^a iljev@mail.ru, ^b alvenko@math.nsc.ru

Аннотация. В задачах кластеризации на графах требуется разбить множество вершин заданного графа на попарно непересекающиеся подмножества (называемые кластерами) так, чтобы минимизировать число рёбер между кластерами и число отсутствующих рёбер внутри кластеров. Мы рассматриваем вариант задачи, в котором размеры кластеров ограничены сверху натуральным числом s . Задача NP-трудна для любого фиксированного $s \geq 3$. Для этого варианта задачи предложены полиномиальные приближённые алгоритмы с гарантированными оценками точности, равными $5/3$ в случае $s = 3$ и 2 в случае $s = 4$. Доказано также, что при $s = 3$ задача APX-трудна. Ил. 5, библиогр. 20.

Ключевые слова: граф, кластеризация, NP-трудная задача, приближённый алгоритм, гарантированная оценка точности.

Введение

В задачах кластеризации требуется разбить заданное множество объектов на попарно непересекающиеся подмножества (кластеры) так, чтобы объекты в каждом кластере были более похожи друг на друга, чем на объекты из других кластеров.

Одной из наиболее наглядных формализаций задач кластеризации являются задачи кластеризации на графах [1]. В 1960–70-х гг. такие задачи изучались под именем задач аппроксимации графов. В этих задачах отношение сходства объектов задаётся посредством неориентированного графа, вершины которого взаимно однозначно соответствуют объектам, а рёбра соединяют похожие объекты. Цель состоит в том, чтобы разбить

множество вершин графа на попарно непересекающиеся подмножества (называемые кластерами) так, чтобы минимизировать число рёбер между кластерами и число отсутствующих рёбер внутри кластеров. Количество кластеров может быть задано, ограничено или заранее не определено. Различные варианты и интерпретации задачи аппроксимации графов содержатся в статьях [2–6].

Позже задачи аппроксимации графов неоднократно и независимо перестраивались и изучались под разными наименованиями (Correlation Clustering, Cluster Editing, Graph Modification Problem и т. д.) [7–9]. В последнее время в зарубежной литературе за невзвешенной версией задачи закрепилось наименование Cluster Editing, в то время как на задачи, в которых рёбрам графа приписаны произвольные веса, обычно ссылаются как на Correlation Clustering [10–12].

Введём необходимые обозначения и определения.

Будем рассматривать только *обыкновенные* графы, т. е. графы без петель и кратных рёбер. Обыкновенный граф называется *кластерным*, если каждая его компонента связности является полным графом [9]. Обозначим через $\mathcal{M}(V)$ семейство всех кластерных графов на множестве вершин V , $\mathcal{M}_k(V)$ — семейство всех кластерных графов на V , имеющих ровно k компонент связности, $\mathcal{M}_{\leq k}(V)$ — семейство всех кластерных графов на множестве V , имеющих не более k компонент связности, $2 \leq k \leq |V|$.

Если $G_1 = (V, E_1)$ и $G_2 = (V, E_2)$ — обыкновенные графы с нумерованными вершинами на одном и том же множестве вершин V , то *расстояние* $d(G_1, G_2)$ между ними определяется как

$$d(G_1, G_2) = |E_1 \Delta E_2| = |E_1 \setminus E_2| + |E_2 \setminus E_1|,$$

т. е. $d(G_1, G_2)$ — число несовпадающих рёбер в графах G_1 и G_2 .

В 1960–80-х гг. изучались следующие варианты задачи аппроксимации графа, которые эквивалентны задачам Cluster Editing (CE).

Задача CE. Для произвольного графа $G = (V, E)$ найти граф $M^* \in \mathcal{M}(V)$ такой, что

$$d(G, M^*) = \min_{M \in \mathcal{M}(V)} d(G, M).$$

Задача CE_k. Для произвольного графа $G = (V, E)$ и натурального числа k , $2 \leq k \leq |V|$, найти граф $M^* \in \mathcal{M}_k(V)$ такой, что

$$d(G, M^*) = \min_{M \in \mathcal{M}_k(V)} d(G, M).$$

Задача CE_{≤k}. Для произвольного графа $G = (V, E)$ и натурального числа k , $2 \leq k \leq |V|$, найти граф $M^* \in \mathcal{M}_{\leq k}(V)$ такой, что

$$d(G, M^*) = \min_{M \in \mathcal{M}_{\leq k}(V)} d(G, M).$$

Задача CE NP-трудна, задачи CE_k и $CE_{\leq k}$ NP-трудны для любого фиксированного $k \geq 2$. Основные результаты о вычислительной сложности и приближённых алгоритмах с гарантированными оценками точности для этих задач содержатся в обзорах [13–15].

В этой работе рассматривается сравнительно новый вариант задачи, в котором размеры кластеров ограничены сверху натуральным числом s . Обозначим через $\mathcal{M}^{\leq s}(V)$ семейство всех кластерных графов на множестве вершин V , в которых размер любой компоненты связности не превосходит s , $2 \leq s \leq |V|$.

Задача $CE^{\leq s}$. Для произвольного графа $G = (V, E)$ и натурального числа s , $2 \leq s \leq |V|$, найти граф $M^* \in \mathcal{M}^{\leq s}(V)$ такой, что

$$d(G, M^*) = \min_{M \in \mathcal{M}^{\leq s}(V)} d(G, M).$$

При доказательстве NP-трудности задачи CE без каких-либо ограничений на число и размеры кластеров Бансал, Блум и Чаула в [7] фактически доказали, что задача $CE^{\leq 3}$ NP-трудна. В 2011 г. Ильев и Навроцкая [16] доказали, что задача $CE^{\leq s}$ NP-трудна для любого фиксированного $s \geq 3$, а задача $CE^{\leq 2}$ полиномиально разрешима. В 2015 г. Пулео и Миленкович [10] предложили 6-приближённый алгоритм для задачи $CE^{\leq s}$, а в 2016 г. Ильев, Ильева и Навроцкая [17] опубликовали для этой задачи приближённый алгоритм, который является 3-приближённым для случая $s = 3$ и 5-приближённым для $s = 4$.

В настоящей работе предлагаются алгоритмы приближённого решения для случаев $s = 3$ и $s = 4$ с улучшенными гарантированными оценками точности. В 2023 г. эти результаты докладывались на международных конференциях MOTOR 2023 и OPTIMA 2023 [18, 19].

В разд. 1 приведены и доказаны необходимые вспомогательные утверждения. В разд. 2 предложен полиномиальный $\frac{5}{3}$ -приближённый алгоритм для задачи $CE^{\leq 3}$, а также доказано, что эта задача APX-трудна. В разд. 3 рассматривается аналогичный полиномиальный 2-приближённый алгоритм для задачи $CE^{\leq 4}$. Гарантированные оценки точности обоих алгоритмов достижимы.

1. Вспомогательные утверждения

Далее будем использовать следующие термины и обозначения. Полный граф на r вершинах будем называть r -кликой и обозначать K_r , а обозначение Q_4 будем использовать для 4-вершинного графа с пятью рёбрами. Для произвольного графа $G = (V, E)$ и его подграфа $H = (U, D)$ обозначим через $\delta(H)$ разрез, определённый множеством вершин графа H , т. е. множество рёбер, ведущих из U в $V \setminus U$.

Докажем некоторые свойства оптимальных решений для задач $CE^{\leq 3}$ и $CE^{\leq 4}$.

Утверждение 1. Пусть $G = (V, E)$ — произвольный граф и H — его подграф. Если $H = K_3$ и $|\delta(H)| \leq 2$, то существует оптимальное решение задачи $CE^{\leq 3}$ на графе G , которое содержит H как компоненту связности.

ДОКАЗАТЕЛЬСТВО. Действительно, пусть вершины 3-клики H принадлежат разным компонентам оптимального решения $M^* \in \mathcal{M}^{\leq 3}(V)$. Тогда не менее двух рёбер клики H вносят вклад в $d(G, M^*)$. Удалим из графа M^* все вершины H вместе со всеми инцидентными им рёбрами и добавим вместо них клику H . Получим новый кластерный граф $M \in \mathcal{M}^{\leq 3}(V)$, причём $d(G, M) \leq d(G, M^*)$. Утверждение 1 доказано.

Аналогично доказывается

Утверждение 2. Пусть $G = (V, E)$ — произвольный граф и H — его подграф. Если $H = K_2$ и $|\delta(H)| \leq 1$, то существует оптимальное решение задачи $CE^{\leq 3}$ на графе G , которое содержит H как компоненту связности.

Из утверждений 1 и 2 следует, что можно произвести предварительную обработку (препроцессинг) данного графа $G = (V, E)$, последовательно находя подграфы, удовлетворяющие условиям утверждений 1 и 2, и добавляя их в решение задачи $CE^{\leq 3}$ с одновременным их удалением из G вместе со всеми рёбрами, инцидентными их вершинам.

Свойство 1. Пусть G — граф, полученный из данного графа после препроцессинга. Если $H \subseteq G$ и $H = K_3$, то $|\delta(H)| \geq 3$.

Свойство 2. Пусть G — граф, полученный из данного графа после препроцессинга. Если $H \subseteq G$ и $H = K_2$, то $|\delta(H)| \geq 2$.

Утверждение 3. В задаче $CE^{\leq 3}$ существует оптимальное решение, являющееся подграфом данного графа.

Утверждение 4. В задаче $CE^{\leq 4}$ существует оптимальное решение, любая клика которого является кликой данного графа, за исключением, быть может, некоторых клик K_4 , которые получены из Q_4 -подграфов данного графа добавлением одного ребра.

ДОКАЗАТЕЛЬСТВО. Докажем утверждение 4, утверждение 3 доказывается аналогично. Очевидно, что если клика $H = K_2$ входит в M^* , то $H \subseteq G$. Предположим, что клика $H = K_3 \subseteq M^*$ получена из некоторого 3-вершинного подграфа $F \subseteq G$ добавлением одного ребра e . Тогда вместо добавления e можем удалить из F другое ребро и получить новый кластерный граф $M' \in \mathcal{M}^{\leq 4}(V)$ с тем же значением целевой функции: $d(G, M') = d(G, M^*)$. Наконец, предположим, что клика $H = K_4 \subseteq M^*$

получена из некоторого 4-вершинного подграфа $F \subseteq G$ добавлением не менее двух рёбер. Тогда вместо их добавления можем удалить из F не более двух рёбер и получить кластерный граф $M' \in \mathcal{M}^{\leq 4}(V)$ такой, что $d(G, M') \leq d(G, M^*)$. Утверждение 4 доказано.

2. Задача $CE^{\leq 3}$

Опишем $\frac{5}{3}$ -приближённый алгоритм для задачи $CE^{\leq 3}$ и докажем его гарантированную оценку точности. Будет также доказано, что задача $CE^{\leq 3}$ APX-трудна.

Пусть $\mathcal{F}$ — некоторое семейство графов. Множество попарно непересекающихся по вершинам подграфов произвольного графа G , каждый из которых изоморфен некоторому графу из $\mathcal{F}$, называется $\mathcal{F}$ -упаковкой в G . Максимальная по включению $\mathcal{F}$ -упаковка называется *максимальной*.

Обозначим через $\mathcal{K}_r$ максимальную $\{K_r\}$ -упаковку в G и подграф графа G , равный объединению всех клик этой упаковки.

Рассмотрим алгоритм приближённого решения задачи $CE^{\leq 3}$. Пусть граф G получен из данного графа после препроцессинга, т. е. для него выполнены свойства 1 и 2.

Алгоритм 1. Приближённый алгоритм для задачи $CE^{\leq 3}$

Вход: граф $G = (V, E)$, обладающий свойствами 1 и 2.

Выход: кластерный граф $M \in \mathcal{M}^{\leq 3}(V)$.

- 1: $G' \leftarrow G$;
 - 2: найти максимальную $\{K_3\}$ -упаковку $\mathcal{K}_3$ в G' ;
 - 3: $G' \leftarrow G' - \mathcal{K}_3$; ▷ Удалить все вершины $\mathcal{K}_3$ вместе со всеми инцидентными рёбрами.
 - 4: найти наибольшее паросочетание $\mathcal{K}_2$ в G' ;
 - 5: $\mathcal{K}_1 \leftarrow G' - \mathcal{K}_2$; ▷ $\mathcal{K}_1$ — множество изолированных вершин графа G' .
 - 6: $M \leftarrow \mathcal{K}_3 \cup \mathcal{K}_2 \cup \mathcal{K}_1$;
-

Теорема 1. Пусть $G = (V, E)$ — произвольный граф. Тогда

$$\frac{d(G, M)}{d(G, M^*)} \leq \frac{5}{3}, \quad (1)$$

где M^* — оптимальное решение задачи $CE^{\leq 3}$ на графе G , M — кластерный граф, построенный алгоритмом 1.

ДОКАЗАТЕЛЬСТВО. Заметим, что если неравенство (1) будет доказано для графа, полученного из G после препроцессинга, то оно будет верно

и для графа G , поэтому далее без ограничения общности будем считать, что G — граф, полученный из данного графа после препроцессинга.

Из утверждения 3 следует, что существует оптимальное решение M^* задачи $CE^{\leq 3}$ на графе G , которое содержит только K_2 - и K_3 -подграфы графа G . Пусть $\mathcal{O}$ — семейство всех клик оптимального решения M^* , а $\mathcal{E}_i$ — семейство всех i -клик в $\mathcal{O}$, $i = 2, 3$. Обозначим через $N(\mathcal{O})$ и $N(\mathcal{A})$ число рёбер в $\mathcal{E}_2 \cup \mathcal{E}_3$ и $\mathcal{K}_2 \cup \mathcal{K}_3$ соответственно. Очевидно, что $N(\mathcal{O}) = 3|\mathcal{E}_3| + |\mathcal{E}_2|$, а из свойств 1 и 2 следует, что $d(G, M^*) \geq \frac{3|\mathcal{E}_3| + 2|\mathcal{E}_2|}{2}$.

В [20] доказано, что

$$N(\mathcal{A}) \geq 2|\mathcal{E}_3| + |\mathcal{E}_2|. \quad (2)$$

Для полноты изложения приведём здесь собственное доказательство этого неравенства.

Инъективно сопоставим (назначим) каждой клике $C \in \mathcal{E}_2$ не менее одного ребра, а каждой клике $C \in \mathcal{E}_3$ — не менее двух рёбер из $\mathcal{K}_2 \cup \mathcal{K}_3$. Рассмотрим произвольную клику $K \in \mathcal{K}_3$ и три её вершины u, v, w .

(а) Пусть $C \in \mathcal{E}_3$ и $C = K$. Сопоставим клике C все три ребра клики K .

(б) Пусть $C \in \mathcal{E}_3 \cup \mathcal{E}_2$ и вершины u, v принадлежат C , а $w \notin C$. Сопоставим рёбра uv и uw клике C .

(в) Пусть $C \in \mathcal{E}_3 \cup \mathcal{E}_2$ и $\{w\} = K \cap C$. Назначим ребро vw клике C .

В результате получим, что если клика $C \in \mathcal{E}_3 \cup \mathcal{E}_2$ имеет i общих вершин с K , то i рёбер клики K сопоставлены C . При этом каждое такое ребро связано только с одной кликой $C \in \mathcal{E}_3 \cup \mathcal{E}_2$.

Далее, рассмотрим граф $G' = G - \mathcal{K}_3$. В G' уже нет треугольников. Обозначим через E^* множество рёбер в $G' \cap (\mathcal{E}_3 \cup \mathcal{E}_2)$. Тогда E^* — некоторое паросочетание в G' .

(г) Пусть $\mathcal{C} = \{C \in \mathcal{E}_3 \cup \mathcal{E}_2 \mid C \cap E^* \neq \emptyset\}$. Так как $\mathcal{K}_2$ — это наибольшее паросочетание в G' , то $|E^*| \leq |\mathcal{K}_2|$. Следовательно, можем сопоставить каждой компоненте $C \in \mathcal{C}$ одно ребро из $\mathcal{K}_2$; если $C \in \mathcal{E}_3$, то это будет уже второе ребро, назначенное клике C .

Таким образом, не менее двух рёбер из $\mathcal{K}_2 \cup \mathcal{K}_3$ сопоставлены каждой клике $C \in \mathcal{E}_3$, и как минимум одно ребро из $\mathcal{K}_2 \cup \mathcal{K}_3$ каждой клике $C \in \mathcal{E}_2$. Тем самым неравенство (2) доказано.

Из (2) следует, что $d(G, M) \leq d(G, M^*) + |\mathcal{E}_3|$ и

$$\frac{d(G, M)}{d(G, M^*)} \leq 1 + \frac{|\mathcal{E}_3|}{d(G, M^*)} \leq 1 + \frac{2|\mathcal{E}_3|}{3|\mathcal{E}_3| + 2|\mathcal{E}_2|} \leq \frac{5}{3}.$$

Теорема 1 доказана.

Замечание 1. Оценка (1) достижима.

Теорема 2. Задача $CE^{\leq 3}$ APX-трудна.

ДОКАЗАТЕЛЬСТВО. Докажем, что задача $CE^{\leq 3}$ APX-трудна на графах максимальной степени 4.

В [20] доказано, что задача о $\{K_2, K_3\}$ -упаковке APX-трудна на графах максимальной степени 4. $\{K_2, K_3\}$ -упаковка в графе G — это множество попарно непересекающихся по вершинам подграфов G , каждый из которых изоморфен K_2 или K_3 . Задача о $\{K_2, K_3\}$ -упаковке состоит в отыскании максимальной по числу рёбер $\{K_2, K_3\}$ -упаковки.

Пусть G — вход задач о $\{K_2, K_3\}$ -упаковке и $CE^{\leq 3}$. Заметим, что любое допустимое решение задачи о $\{K_2, K_3\}$ -упаковке является также допустимым решением задачи $CE^{\leq 3}$ (не считая изолированных вершин). Пусть $f(\text{Sol})$ и $g(\text{Sol})$ — значения целевых функций на допустимом решении Sol задач о $\{K_2, K_3\}$ -упаковке и $CE^{\leq 3}$ соответственно. Очевидно, что $f(\text{Sol}) = m - g(\text{Sol})$, где m — число рёбер графа G . Значит, если Opt — оптимальное решение задачи о $\{K_2, K_3\}$ -упаковке, то Opt есть также оптимальное решение задачи $CE^{\leq 3}$ и $f(\text{Opt}) = m - g(\text{Opt})$. Кроме того,

$$f(\text{Opt}) - f(\text{Sol}) = m - g(\text{Opt}) - m + g(\text{Sol}) = g(\text{Sol}) - g(\text{Opt}).$$

Пусть G — граф максимальной степени 4. Заметим, что $f(\text{Opt}) \geq m/7$. В самом деле, обозначим через z число рёбер некоторого максимального паросочетания M в G . Каждое ребро паросочетания M смежно (т. е. имеет общую вершину) не более чем с 6 рёбрами графа G . Поскольку любое ребро графа G либо принадлежит Opt , либо смежно с некоторым ребром из Opt , то $m \leq 7z \leq 7f(\text{Opt})$.

Предположим, что A — некоторый α -приближённый алгоритм для задачи $CE^{\leq 3}$, и пусть Sol — приближённое решение, найденное этим алгоритмом. По аналогии с утверждением 3 можно доказать, что тогда существует допустимое решение задачи $CE^{\leq 3}$ с тем же значением целевой функции, которое является подграфом графа G . Без ограничения общности будем считать, что Sol — именно такое решение. Тогда Sol является допустимым решением задачи о $\{K_2, K_3\}$ -упаковке, откуда получаем

$$\begin{aligned} f(\text{Sol}) &= f(\text{Opt}) + g(\text{Opt}) - g(\text{Sol}) \geq m - \alpha g(\text{Opt}) = \\ &= m - \alpha(m - f(\text{Opt})) = \alpha f(\text{Opt}) - (\alpha - 1)m \geq \\ &\geq \alpha f(\text{Opt}) - 7(\alpha - 1)f(\text{Opt}) = (7 - 6\alpha)f(\text{Opt}). \end{aligned}$$

Таким образом, если существует α -приближённый алгоритм для задачи $CE^{\leq 3}$, то существует и $(7 - 6\alpha)$ -приближённый алгоритм для задачи о $\{K_2, K_3\}$ -упаковке. В частности, если бы для любого $\varepsilon > 0$ существовал $(1 + \varepsilon)$ -приближённый алгоритм для задачи $CE^{\leq 3}$, то существовал бы $(1 - 6\varepsilon)$ -приближённый алгоритм для задачи о $\{K_2, K_3\}$ -упаковке. Поскольку задача о $\{K_2, K_3\}$ -упаковке APX-трудна на графах максимальной степени 4, задача $CE^{\leq 3}$ также APX-трудна. Теорема 2 доказана.

3. Задача $CE^{\leq 4}$

Опишем 2-приближённый алгоритм для задачи $CE^{\leq 4}$ и докажем его гарантированную оценку точности.

Рассмотрим алгоритм приближённого решения задачи $CE^{\leq 4}$.

Алгоритм 2. Приближённый алгоритм для задачи $CE^{\leq 4}$

Вход: произвольный граф $G = (V, E_G)$.

Выход: кластерный граф $M = (V, E_M) \in \mathcal{M}^{\leq 4}(V)$.

- 1: $G' \leftarrow G$;
 - 2: найти максимальную $\{K_4\}$ -упаковку $\mathcal{K}_4$ в G' ;
 - 3: $G' \leftarrow G' - \mathcal{K}_4$; ▷ Удалить все вершины $\mathcal{K}_4$ вместе со всеми инцидентными рёбрами.
 - 4: найти максимальную $\{K_3\}$ -упаковку $\mathcal{K}_3$ в G' ;
 - 5: $G' \leftarrow G' - \mathcal{K}_3$; ▷ Удалить все вершины $\mathcal{K}_3$ вместе со всеми инцидентными рёбрами.
 - 6: найти максимальное паросочетание $\mathcal{K}_2$ в G' ;
 - 7: $\mathcal{K}_1 \leftarrow G' - \mathcal{K}_2$; ▷ $\mathcal{K}_1$ — множество изолированных вершин графа G' .
 - 8: $M \leftarrow \mathcal{K}_4 \cup \mathcal{K}_3 \cup \mathcal{K}_2 \cup \mathcal{K}_1$;
-

Далее будем рассматривать только те оптимальные решения задачи $CE^{\leq 4}$, что удовлетворяют утверждению 4. Пусть G — произвольный граф, M^* — оптимальное решение задачи $CE^{\leq 4}$ на G . Введём следующие обозначения:

- E_1 — множество рёбер графа G , которые не входят в M^* , но помещены в M алгоритмом 2 (т. е. $E_1 = (E_G \cap E_M) \setminus E_{M^*}$);
- $E_2 = E_{M^*} \setminus E_G$;
- E^* — множество рёбер графа G , которые не входят в M , но входят в M^* (т. е. $E^* = (E_G \cap E_{M^*}) \setminus E_M$).

Лемма 1. *Имеет место неравенство $|E^*| \leq 2(|E_1| + |E_2|)$.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим мысленную процедуру пометки рёбер множества E^* , шаги которой соответствуют шагам алгоритма 2. Процедура присваивает метку каждому ребру из E^* , которое удаляется из текущего графа G' на очередном шаге алгоритма 2. Пометка рёбер осуществляется с помощью рёбер множеств E_1 и E_2 . Каждое ребро $e \in E_1 \cup E_2$ даёт свои метки вида e^* не более чем двум смежным с e непомеченным рёбрам множества E^* . Изначально рёбра E^* не помечены.

Покажем сначала, что в множестве E_1 достаточно рёбер для пометки рёбер всех клик K_4, K_3 и K_2 графа M^* , являющихся подграфами

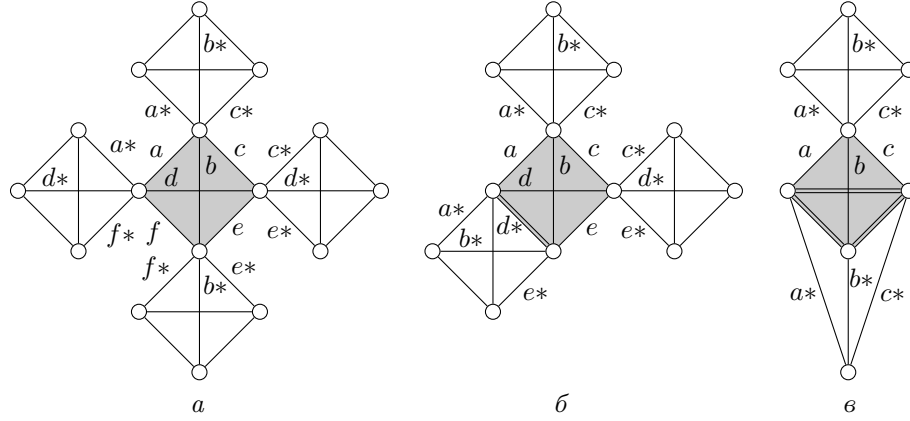


Рис. 1

графа G , и только некоторые рёбра клик $K_4 \subseteq M^*$, полученных из подграфов $Q_4 \subset G$, могут остаться непомеченными с использованием рёбер множества E_1 в момент удаления из текущего графа G' .

1. Заметим, что клика K_4 , найденная алгоритмом 2 на шаге 2, может иметь 0, 1, 3 или 6 общих рёбер с кликами графа M^* . Тем самым только 6, 5 или 3 ребра любой клики $K_4 \subseteq M$ могут принадлежать множеству E_1 . Рассмотрим эти случаи с иллюстрацией на рис. 1: серые тетраэдры — клики K_4 , найденные алгоритмом 2; белые тетраэдры — клики графа M^* (некоторые из них могут быть заменены кликами K_3 , K_2 или K_1). На всех рисунках двойной линией обозначены общие рёбра графов M и M^* .

(а) Пусть K_4 — клика, найденная алгоритмом 2 на шаге 2, все шесть рёбер a, b, c, d, e, f которой принадлежат множеству E_1 . Тогда в графе G' не более 12 непомеченных рёбер множества E^* смежны с этими шестью рёбрами, они получают не более 12 меток вида $a^*, b^*, c^*, d^*, e^*, f^*$ (рис. 1а).

(б) Пусть K_4 — клика, найденная алгоритмом 2 на шаге 2, в которой пять рёбер a, b, c, d, e принадлежат множеству E_1 . Тогда в графе G' не более 10 непомеченных рёбер множества E^* смежны с этими пятью рёбрами, они получают не более 10 меток вида a^*, b^*, c^*, d^*, e^* (рис. 1б).

(в) Пусть K_4 — клика, найденная алгоритмом 2 на шаге 2, три ребра a, b, c которой принадлежат множеству E_1 . Тогда в графе G' не более 6 непомеченных рёбер множества E^* смежны с этими тремя рёбрами, они получают не более 6 меток вида a^*, b^*, c^* (рис. 1в).

2. Заметим, что клика K_3 , найденная алгоритмом 2 на шаге 4, может иметь 0, 1 или 3 общих ребра с кликами графа M^* , поэтому только 3

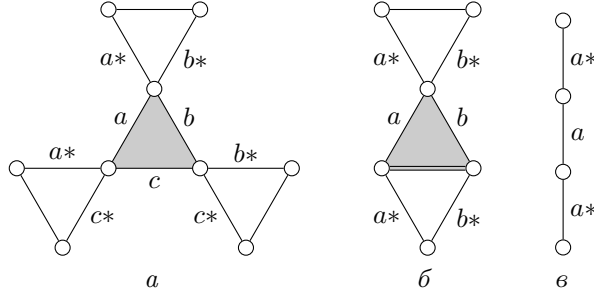


Рис. 2

или 2 ребра каждой клики $K_3 \subseteq M$ принадлежат множеству E_1 . Рассмотрим эти случаи с иллюстрацией на рис. 2а, 2б: серые треугольники — клики, найденные алгоритмом 2, белые треугольники — клики графа M^* , в котором часть клик K_3 могут быть заменены кликами K_2 или K_1 , а часть клик K_3 могут содержаться в кликах $K_4 \subset M^*$ в качестве подграфов.

(а) Пусть K_3 — клика, найденная алгоритмом 2 на шаге 4, все три ребра a, b, c которой принадлежат множеству E_1 . Тогда в графе G' не более 6 непомеченных рёбер множества E^* смежны с этими тремя рёбрами, они получают не более 6 меток вида a^*, b^*, c^* (рис. 2а).

(б) Пусть K_3 — клика, найденная алгоритмом 2 на шаге 4, только два ребра a, b которой принадлежат множеству E_1 . Тогда в графе G' не более 4 непомеченных рёбер множества E^* смежны с этими двумя рёбрами, они получают не более 4 меток вида a^*, b^* (рис. 2б).

3. Заметим, что если ребро e клики K_2 , найденной алгоритмом 2 на шаге 6, принадлежит множеству E_1 , то не более двух рёбер множества E^* смежны с e в текущем графе G' , они получают метки a^* (рис. 2в).

Перечислим теперь случаи, когда рёбер E_1 может оказаться недостаточно для пометки рёбер из E^* , удаляемых из текущего графа G' на очередном шаге алгоритма 2. В каждом из этих случаев рёбра из E^* , оставшиеся непомеченными рёбрами из E_1 , принадлежат некоторому подграфу $Q_4 \subset G$ с недостающим ребром $x \in E_2 = E_{M^*} \setminus E_G$. Эти случаи иллюстрируются рис. 3 и 4, на которых серым обозначены клики, найденные алгоритмом 2, а пунктирной линией — ребро x .

СЛУЧАЙ 1. На шаге 2 алгоритм 2 находит клику $K_4 \subset G'$, три ребра a, b, c которой инцидентны вершине 1 подграфа Q_4 . Тогда $a, b, c \in E_1$, и они дают свои метки a^*, b^*, c^* рёбрам 12, 13 и 14 соответственно (возможно, это их вторые метки, а первые метки a^*, b^*, c^* были даны некоторым рёбрам, инцидентным другим концам рёбер a, b, c) (рис. 3а).

После удаления K_4 и всех инцидентных ей рёбер из G' следующий граф G' будет содержать рёбра $23, 34 \in Q_4$. Заметим, что по меньшей мере одно из этих рёбер должно принадлежать E^* . Возможны два случая.

(а) Только одно из рёбер $23, 34$ принадлежит E^* . Без ограничения общности предположим, что $23 \in E^*$. Это означает, что алгоритм 2 включил в M некоторую клику, содержащую ребро 34 . Если это клика K_4 или K_3 , то она также содержит некоторое ребро d , инцидентное вершине 3. Заметим, что $d \in E_1$, и оно даёт свою метку d^* ребру 23 .

Единственная ситуация, когда ребро 23 в момент его удаления из текущего графа G' может остаться без метки, полученной от рёбер множества E_1 , следующая. Алгоритм 2 на шаге 6 находит клику $K_2 = \langle 34 \rangle$, и таким образом ребро 34 удаляется из графа G' вместе со всеми инцидентными рёбрами. Однако в этом случае $34 \notin E_1$, следовательно, ребро 23 остаётся немеченным с помощью рёбер множества E_1 . Во избежание такой ситуации процедура приписывает ребру 23 метку x^* , где $x = 24 \in E_2$ (рис. 3б).

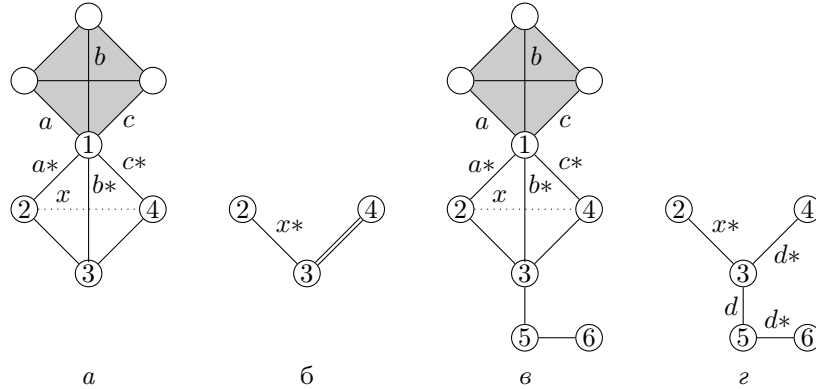


Рис. 3

(б) Оба ребра $23, 34$ принадлежат E^* . Это означает, что алгоритм 2 включил в M некоторые рёбра, инцидентные одной, двум или трём вершинам множества $\{2, 3, 4\}$. Эти рёбра принадлежат E_1 и дают свои метки обоим рёбрам $23, 34$, кроме единственной ситуации, когда алгоритм 2 на шаге 6 находит некоторую клику $K_2 = \langle 35 \rangle$, где $5 \notin \{2, 4\}$, и кроме того, в G' существует ребро $56 \in E^*$ (рис. 3в). Тогда $d = 35 \in E_1$ и d даёт свои метки d^* ребру 56 и одному из рёбер 23 и 34 , скажем 34 . В этой ситуации ребро 23 остаётся немеченным с помощью рёбер множества E_1 . Чтобы избежать такой ситуации, процедура назначает ребру 23 метку x^* , где $x = 24 \in E_2$ (рис. 3г).

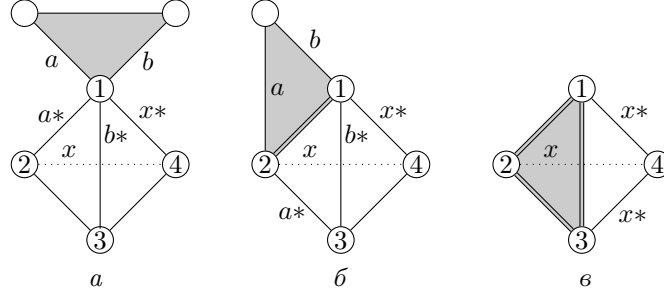


Рис. 4

СЛУЧАЙ 2 отличается от СЛУЧАЯ 1 тем, что алгоритм 2 находит на шаге 4 в графе G' клику K_3 вместо K_4 (рис. 4а). Здесь, как и в СЛУЧАЕ 1, рёбра 12 и 13 получают метки a^* и b^* , но ребро 14 не может быть помечено с помощью рёбер множества E_1 . Таким образом, процедура присваивает ребру 14 метку x^* , где $x = 24 \in E_2$.

И точно так же, чтобы избежать нежелательных ситуаций в следующем графе G' , перечисленных в пп. (а), (б) СЛУЧАЯ 1, ребро 23 получает метку x^* , где $x = 24 \in E_2$.

СЛУЧАЙ 3 отличается от СЛУЧАЯ 2 тем, что клика K_3 , найденная алгоритмом 2 на шаге 4, имеет общее ребро с Q_4 , скажем 12 (рис. 4б). В этом случае, как и ранее, ребро 14 получает метку x^* , где $x = 24 \in E_2$.

СЛУЧАЙ 4. На шаге 4 алгоритм 2 находит клику K_3 с рёбрами 12, 13, 23 и удаляет её из графа G' вместе со всеми инцидентными рёбрами, но ещё до этого удаления процедура присваивает рёбрам 14 и 34 метки x^* , где $x = 24 \in E_2$ (рис. 4в).

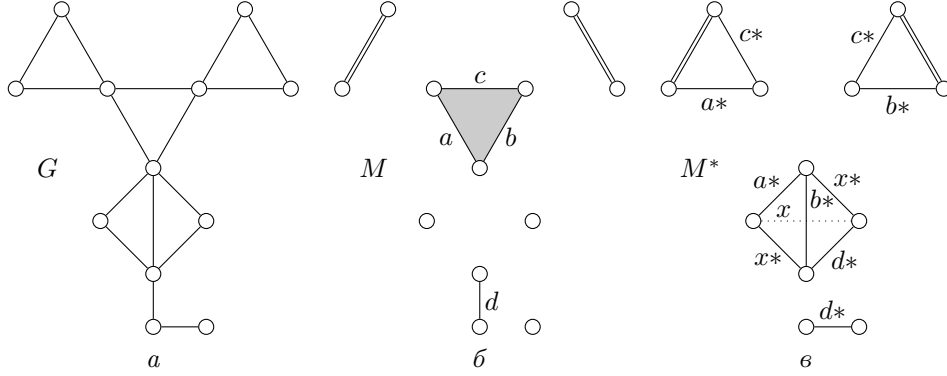
По завершении работы алгоритма 2 граф G' становится пустым. Поскольку все рёбра множества E^* , которые удалялись из текущего графа G' , получили метки, а каждое ребро $e \in E_1$ или E_2 дало свои метки не более чем двум смежным с e непомеченным рёбрам множества E^* , то $|E^*| \leq 2(|E_1| + |E_2|)$. Лемма 1 доказана.

Пример работы алгоритма 2 представлен на рис. 5. Этот пример показывает достижимость следующей оценки.

Теорема 3. Пусть $G = (V, E_G)$ — произвольный граф. Тогда

$$\frac{d(G, M)}{d(G, M^*)} \leq 2, \quad (3)$$

где $M^* = (V, E_{M^*})$ — оптимальное решение задачи $CE^{\leq 4}$ на графе G , $M = (V, E_M)$ — кластерный граф, построенный алгоритмом 2.

Рис. 5. Пример работы алгоритма 2: графы G , M и M^*

ДОКАЗАТЕЛЬСТВО. По определению

$$d(G, M^*) = |E_G \setminus E_{M^*}| + |E_{M^*} \setminus E_G|.$$

Представим разность $E_G \setminus E_{M^*}$ в виде $E_G \setminus E_{M^*} = E_0 \cup E_1$, где E_0 — множество рёбер графа G , которые не входят в M^* и не помещены в M алгоритмом 2, а E_1 — множество рёбер графа G , которые не входят в M^* , но помещены в M алгоритмом 2. Положим $E_2 = E_{M^*} \setminus E_G$. Тогда

$$d(G, M^*) = |E_0| + |E_1| + |E_2|. \quad (4)$$

По построению граф M является подграфом графа G , следовательно, $d(G, M) = |E_G \setminus E_M|$. Запишем разность $E_G \setminus E_M$ в следующем виде: $E_G \setminus E_M = E_0 \cup E^*$, где E^* — множество рёбер графа G , не помещённых в M , но входящих в M^* . Тогда

$$d(G, M) = |E_0| + |E^*|. \quad (5)$$

Следовательно, в силу (4), (5) и леммы 1 получим

$$\frac{d(G, M)}{d(G, M^*)} = \frac{|E_0| + |E^*|}{|E_0| + |E_1| + |E_2|} \leq \frac{|E^*|}{|E_1| + |E_2|} \leq \frac{2(|E_1| + |E_2|)}{|E_1| + |E_2|} = 2.$$

Теорема 3 доказана.

Заключение

В работе рассматривается вариант задачи кластеризации на графе, в котором размеры кластеров ограничены сверху натуральным числом s . Эта задача NP-трудна для любого фиксированного $s \geq 3$. Для указанного варианта задачи предложены полиномиальные приближённые алгоритмы с достижимыми гарантированными оценками точности, равными $\frac{5}{3}$ в случае $s = 3$ и 2 в случае $s = 4$. Доказано также, что для $s = 3$ эта задача APX-трудна.

Финансирование работы

Работа первого автора выполнена в рамках гос. задания Института математики им. С. Л. Соболева (проект № FWNF-2022-0020). Работа второго автора выполнена за счёт бюджета Омского гос. университета им. Ф. М. Достоевского. Работа третьего автора выполнена в рамках гос. задания Института математики им. С. Л. Соболева (проект № FWNF-2022-0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Schaeffer S. E. Graph clustering // Comput. Sci. Rev. 2005. V. 1, No. 1. P. 27–64.
2. Фридман Г. III. Одна задача аппроксимации графов // Управляемые системы. Вып. 8. Новосибирск: Изд-во Ин-та математики, 1971. С. 73–75.
3. Фридман Г. III. Исследование одной задачи классификации на графах // Методы моделирования и обработка информации. Новосибирск: Наука, 1976. С. 147–177.
4. Tomescu I. La reduction minimale d'un graphe à une reunion de cliques // Discrete Math. 1974. V. 10, No. 1–2. P. 173–179 [French].
5. Zahn C. T. Approximating symmetric relations by equivalence relations // J. Soc. Ind. Appl. Math. 1964. V. 12, No. 4. P. 840–847.
6. Агеев А. А., Ильев В. П., Кононов А. В., Талевнин А. С. Вычислительная сложность задачи аппроксимации графов // Дискрет. анализ и исслед. операций. Сер. 1. 2006. Т. 13, № 1. С. 3–11.
7. Bansal N., Blum A., Chawla S. Correlation clustering // Mach. Learn. 2004. V. 56, No. 1–3. P. 89–113.
8. Ben-Dor A., Shamir R., Yakhimi Z. Clustering gene expression patterns // J. Comput. Biol. 1999. V. 6, No. 3–4. P. 281–297.
9. Shamir R., Sharan R., Tsur D. Cluster graph modification problems // Discrete Appl. Math. 2004. V. 144, No. 1–2. P. 173–182.
10. Puleo G. J., Milenkovic O. Correlation clustering with constrained cluster sizes and extended weights bounds // SIAM J. Optim. 2015. V. 25, No. 3. P. 1857–1872.
11. Pandove D., Goel S., Rani R. Correlation clustering methodologies and their fundamental results // Expert Syst. 2018. V. 35, No. 1. Paper ID e12229. 24 p.
12. Chawla S., Makarychev K., Schramm T., Yaroslavl'tsev G. Near optimal LP rounding algorithm for correlation clustering on complete and complete k -partite graphs // Proc. 47th Annu. ACM Symp. Theory of Computing (Portland, OR, USA, June 14–17, 2015). New York: ACM, 2015. P. 219–228. DOI: 10.1145/2746539.2746604.

13. **Il'ev V. P., Il'eva S. D., Kononov A. V.** Short survey on graph correlation clustering with minimization criteria // Discrete optimization and operations research. Proc. 9th Int. Conf. (Vladivostok, Russia, Sept. 19–23, 2016). Cham: Springer, 2016. P. 25–36. (Lect. Notes Comput. Sci.; V. 9869). DOI: 10.1007/978-3-319-44914-2_3.
14. **Wahid D. F., Hassini E.** A literature review on correlation clustering: Cross-disciplinary taxonomy with bibliometric analysis // Oper. Res. Forum. 2022. V. 3. Paper ID 47. 42 p.
15. **Bonchi F., García-Soriano D., Gullo F.** Correlation clustering. Cham: Springer, 2022. 148 p. DOI: 10.1007/978-3-031-79210-6.
16. **Ильев В. П., Навроцкая А. А.** Вычислительная сложность задачи аппроксимации графами с компонентами связности ограниченного размера // Прикл. дискрет. математика. 2011. № 3. С. 80–84.
17. **Ильев В. П., Ильева С. Д., Навроцкая А. А.** О задаче кластеризации графа с ограничением на размеры кластеров // Дискрет. анализ и исслед. операций. 2016. Т. 23, № 3. С. 5–20.
18. **Il'ev V. P., Il'eva S. D.** Approximation algorithms for graph cluster editing problems with cluster size at most 3 or 4 // Mathematical optimization theory and operations research: Recent trends. Rev. Sel. Pap. 22nd Int. Conf. (Yekaterinburg, Russia, July 2–8, 2023). Cham: Springer, 2023. P. 134–145. (Commun. Comput. Inf. Sci.; V. 1881). DOI: 10.1007/978-3-031-43257-6_11.
19. **Kononov A. V., Il'ev V. P.** On cluster editing problem with clusters of small sizes // Optimization and applications. Rev. Sel. Pap. 14th Int. Conf. (Petrovac, Montenegro, Sept. 18–22, 2023). Cham: Springer, 2023. P. 316–328. (Lect. Notes Comput. Sci.; V. 14395). DOI: 10.1007/978-3-031-47859-8_23.
20. **Chataigner F., Manić G., Wakabayashi Y., Yuster R.** Approximation algorithms and hardness results for the clique packing problem // Discrete Appl. Math. 2009. V. 157, No. 7. P. 1396–1406. DOI: 10.1016/j.dam.2008.10.017.

Ильев Виктор Петрович
Ильева Светлана Диадоровна
Кононов Александр Вениаминович

Статья поступила
17 мая 2024 г.
После доработки —
11 июня 2024 г.
Принята к публикации
22 июня 2024 г.

APPROXIMATION ALGORITHMS FOR GRAPH CLUSTERING
PROBLEMS WITH CLUSTERS OF BOUNDED SIZEV. P. Il'ev^{1,2}, S. D. Il'eva², and A. V. Kononov¹¹ Sobolev Institute of Mathematics,
4 Acad. Koptuyug Avenue, 630090 Novosibirsk, Russia² Dostoevsky Omsk State University,
55-A Mir Avenue, 644077 Omsk, RussiaE-mail: ^ailjev@mail.ru, ^balvenko@math.nsc.ru

Abstract. In the cluster editing problem, one has to partition the set of vertices of a graph into pairwise disjoint subsets (called clusters) minimizing the number of edges between clusters and the number of missing edges within clusters. We consider a version of the problem in which cluster sizes are bounded from above by a positive integer s . This problem is NP-hard for any fixed $s \geq 3$. We propose polynomial-time approximation algorithms for this version of the problem. Their performance guarantees equal $5/3$ for the case $s = 3$ and 2 for $s = 4$. We also show that the cluster editing problem is APX-hard for the case of $s = 3$. Illustr. 5, bibliogr. 20.

Keywords: graph, clustering, NP-hard problem, approximation algorithm, performance guarantee.

References

1. S. E. Schaeffer, Graph clustering, *Comput. Sci. Rev.* **1** (1), 27–64 (2005).
2. G. Sh. Fridman, One problem in graph approximation, in *Controlled Systems*, Vol. 8 (Izd. Inst. Mat., Novosibirsk, 1971), pp. 73–75 [Russian].
3. G. Sh. Fridman, Investigation of one problem in graph classification, in *Modelling Methods and Information Processing* (Nauka, Novosibirsk, 1976), pp. 147–177 [Russian].
4. I. Tomescu, La reduction minimale d'un graphe à une reunion de cliques, *Discrete Math.* **10** (1–2), 173–179 (1974) [French].
5. C. T. Zahn, Approximating symmetric relations by equivalence relations, *J. Soc. Ind. Appl. Math.* **12** (4), 840–847 (1964).

6. **A. A. Ageev, V. P. Il'ev, A. V. Kononov, and A. S. Talevnin**, Computational complexity of the graph approximation problem, *Diskretn. Anal. Issled. Oper., Ser. 1*, **13** (1), 3–11 (2006) [Russian] [*J. Appl. Ind. Math.* **1** (1), 1–8 (2023)].
7. **N. Bansal, A. Blum, and S. Chawla**, Correlation clustering, *Mach. Learn.* **56** (1–3), 89–113 (2004).
8. **A. Ben-Dor, R. Shamir, and Z. Yakhimi**, Clustering gene expression patterns, *J. Comput. Biol.* **6** (3–4), 281–297 (1999).
9. **R. Shamir, R. Sharan, and D. Tsur**, Cluster graph modification problems, *Discrete Appl. Math.* **144** (1–2), 173–182 (2004).
10. **G. J. Puleo and O. Milenkovic**, Correlation clustering with constrained cluster sizes and extended weights bounds, *SIAM J. Optim.* **25** (3), 1857–1872 (2015).
11. **D. Pandove, S. Goel, and R. Rani**, Correlation clustering methodologies and their fundamental results, *Expert Syst.* **35** (1), ID e12229 (2018).
12. **S. Chawla, K. Makarychev, T. Schramm, and G. Yaroslavtsev**, Near optimal LP rounding algorithm for correlation clustering on complete and complete k -partite graphs, in *Proc. 47th Annu. ACM Symp. Theory of Computing, Portland, OR, USA, June 14–17, 2015* (ACM, New York, 2015), pp. 219–228, DOI: 10.1145/2746539.2746604.
13. **V. P. Il'ev, S. D. Il'eva, and A. V. Kononov**, Short survey on graph correlation clustering with minimization criteria, in *Discrete Optimization and Operations Research* (Proc. 9th Int. Conf., Vladivostok, Russia, Sept. 19–23, 2016) (Springer, Cham, 2016), pp. 25–36 (Lect. Notes Comput. Sci., Vol. 9869), DOI: 10.1007/978-3-319-44914-2_3.
14. **D. F. Wahid and E. Hassini**, A literature review on correlation clustering: Cross-disciplinary taxonomy with bibliometric analysis, *Oper. Res. Forum* **3**, ID 47 (2022).
15. **F. Bonchi, D. García-Soriano and F. Gullo**, *Correlation Clustering* (Springer, Cham, 2022), DOI: 10.1007/978-3-031-79210-6.
16. **V. P. Il'ev and A. A. Navrotskaya**, Computational complexity of the problem of approximation by graphs with connected components of bounded size, *Prikl. Diskretn. Mat.*, No. 3, 80–84 (2011) [Russian].
17. **V. P. Il'ev, S. D. Il'eva, and A. A. Navrotskaya**, Graph clustering with a constraint on cluster sizes, *Diskretn. Anal. Issled. Oper.* **23** (3), 5–20 (2016) [Russian] [*J. Appl. Ind. Math.* **10** (3), 341–348 (2016)].
18. **V. P. Il'ev and S. D. Il'eva**, Approximation algorithms for graph cluster editing problems with cluster size at most 3 or 4, in *Mathematical Optimization Theory and Operations Research: Recent Trends* (Rev. Sel. Pap. 22nd Int. Conf., Yekaterinburg, Russia, July 2–8, 2023) (Springer, Cham, 2023), pp. 134–145 (Commun. Comput. Inf. Sci., Vol. 1881), DOI: 10.1007/978-3-031-43257-6.
19. **A. V. Kononov and V. P. Il'ev**, On cluster editing problem with clusters of small sizes, in *Optimization and Applications* (Rev. Sel. Pap. 14th Int. Conf., Petrovac, Montenegro, Sept. 18–22, 2023) (Springer, Cham, 2023), pp. 316–328 (Lect. Notes Comput. Sci., Vol. 14395), DOI: 10.1007/978-3-031-47859-8.

- 20. F. Chataigner, G. Manić, Y. Wakabayashi, and R. Yuster,** Approximation algorithms and hardness results for the clique packing problem, *Discrete Appl. Math.* **157** (7), 1396–1406 (2009), DOI: 10.1016/j.dam.2008.10.017.

Victor P. Il'ev

Svetlana D. Il'eva

Aleksandr V. Kononov

Received May 17, 2024

Revised June 11, 2024

Accepted June 22, 2024

ЧИСЛО НЕВОЗМОЖНЫХ РАЗНОСТЕЙ ПО МОДУЛЮ 2^n КОМПОЗИЦИИ XOR И ЦИКЛИЧЕСКОГО СДВИГА БИТОВ

Н. А. Коломеец

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: nkolomeec@gmail.com

Аннотация. Для преобразования $(x \oplus y) \lll r$, $x, y \in \mathbb{Z}_2^n$, $1 \leq r < n$, исследуются невозможные разности по модулю 2^n . Они представляют интерес в контексте разностных атак на шифры, схемы которых состоят из сложений по модулю 2^n , побитовых XOR ($\oplus$) и циклических сдвигов битов на r позиций ($\lll r$). В работе найдено число всех невозможных разностей при всех возможных значениях n и r . В качестве следствия показано, что оно больше $\frac{38}{245}8^n$, причём эта оценка асимптотически точна при r , $n - r \rightarrow \infty$. При фиксированном n число невозможных разностей монотонно убывает с ростом r от 1 до $\lceil n/2 \rceil$ ($n/2 + 1$ при $n \in \{4, 6, 8, 10, 12\}$), а далее — монотонно возрастает. Приведено более компактное описание всех невозможных разностей с точностью до известных симметрий. Табл. 10, библиогр. 25.

Ключевые слова: ARX, разностная характеристика, XOR, сложение по модулю, циклический сдвиг битов, невозможная разность.

Введение

Примитивы симметричной криптографии, ориентированные на программную реализацию, могут быть представлены в виде схем, состоящих только из сложений по модулю 2^n ($\boxplus$), побитовых исключающих «или» (XOR, $\oplus$) и циклических сдвигов битов на r позиций в сторону старших разрядов ($\lll r$). Такие конструкции называются ARX-шифрами (см. Speck [1], CHAM [2], Salsa20 [3], ChaCha [4], Sparkle [5], Chaskey [6] и многие др.). Построение оценок их стойкости к разностным атакам [7] приводит к необходимости анализа разностных характеристик преобразований в этом базисе. Далее будем рассматривать разности относительно операции $\boxplus$, в общем же они могут быть выбраны и иными способами (см. [8–12]).

Значение разностной характеристики adp^f по модулю 2^n (т. е. относительно $\boxplus$) для преобразования $f: (\mathbb{Z}_2^n)^k \rightarrow \mathbb{Z}_2^n$ на наборе разностей $\alpha^1, \dots, \alpha^{k+1} \in \mathbb{Z}_2^n$ определяется следующим образом:

$$\begin{aligned} \text{adp}^f(\alpha^1, \dots, \alpha^k \rightarrow \alpha^{k+1}) &= \\ &= \Pr_{(x^1, \dots, x^k) \in (\mathbb{Z}_2^n)^k} [f(x^1 \boxplus \alpha^1, \dots, x^k \boxplus \alpha^k) = f(x^1, \dots, x^k) \boxplus \alpha^{k+1}], \end{aligned}$$

где $x^1, \dots, x^k$ независимы и равномерно распределены на $\mathbb{Z}_2^n$. Таким образом, это вероятность преобразования функцией f входных разностей $\alpha^1, \dots, \alpha^k$ в выходную разность α^{k+1} . Например, свойства разностной характеристики по модулю 2^n функции $x \mapsto x \lll r$ исследовались в [13, 14], функции $(x, y) \mapsto x \oplus y$ — в [15–17], а функции $(x, y) \mapsto (x \lll r) \oplus y$ — в [18–20]. Обозначим через adp^{XR} разностную характеристику для XR-преобразования $(x, y) \mapsto (x \oplus y) \lll r$, где $x, y \in \mathbb{Z}_2^n$ и $1 \leq r < n$, а через $\mathcal{N}_n^r$ — число невозможных разностей этого преобразования (вероятность возникновения которых равна 0). В [19] найдены все невозможные разности XR-преобразования в виде шаблонов — регулярных выражений (см. разд. 1). В настоящей работе продолжены исследования в этом направлении: приводится точная формула для $\mathcal{N}_n^r$, устанавливаются его свойства и упрощается список невозможных разностей с применением известных симметрий adp^{XR} . Отметим, что все невозможные разности также известны для базовых операций и композиций нескольких XOR [21]. Они наиболее интересны в контексте атак, предложенных в [22, 23].

Разд. 1 содержит базовые определения и обозначения. В разд. 2 приведено описание всех невозможных разностей XR-преобразования (утверждение 1 и табл. 2), полученное в [19]. Точная формула для $\mathcal{N}_n^r$ найдена в разд. 3 (теорема 1). В разд. 4 рассматриваются свойства $\mathcal{N}_n^r$. Нижняя оценка $\mathcal{N}_n^r > \frac{38}{245}8^n$, асимптотически точная при $r, n - r \rightarrow \infty$, доказана в п. 4.1. В п. 4.2 установлено, что

$$\mathcal{N}_n^1 > \mathcal{N}_n^2 > \dots > \mathcal{N}_n^s < \mathcal{N}_n^{s+1} < \dots < \mathcal{N}_n^{n-1},$$

где $s = \lceil n/2 \rceil$ при $n \notin \{4, 6, 8, 10, 12\}$ и $s = n/2 + 1$ иначе (теорема 2). Как следствие, справедливы следующие оценки:

$$\mathcal{N}_n^s \leq \mathcal{N}_n^r \leq \mathcal{N}_n^{n-1} < \mathcal{N}_n^2 < \mathcal{N}_n^1 \quad \text{при } 3 \leq r < n.$$

Оценка $\mathcal{N}_n^r \leq \mathcal{N}_n^{n-1}$ при $r \geq 3$ точнее, чем $\mathcal{N}_n^r < \mathcal{N}_n^1$ из [19], в виду равенств

$$\mathcal{N}_n^1 = \frac{5}{14}8^n - \frac{6}{7}, \quad \mathcal{N}_n^{n-1} = \frac{3}{14}8^n + \frac{2}{3}4^n - \frac{50}{21}.$$

Известно, что преобразование $(x, y) \mapsto x \oplus y$ имеет ровно $\frac{4}{7}8^n - \frac{4}{7}$ невозможных разностей [15]. Таким образом, добавление циклического сдвига существенно снижает их число. В разд. 5 приводится описание всех

невозможных разностей XR-преобразования с точностью до симметрий, сохраняющих значение adp^{XR} (теорема 3 и табл. 10). В отличие от табл. 2, такое описание содержит 10 пар шаблонов, что существенно компактнее.

Заметим, что полученные результаты применимы к преобразованиям RX: $(x, y) \mapsto (x \lll r) \oplus y$, ARX: $(x, y, z) \mapsto ((x \boxplus y) \lll r) \oplus z$ и аналогичным, поскольку их разностные характеристики прямо выражаются через adp^{XR} , а именно: при любых $\alpha, \beta, \gamma, \delta \in \mathbb{Z}_2^n$ имеем

$$\begin{aligned}\text{adp}^{\text{RX}}(\alpha, \beta \xrightarrow{r} \gamma) &= \text{adp}^{\text{XR}}(\gamma, \beta \xrightarrow{n-r} \alpha), \\ \text{adp}^{\text{ARX}}(\alpha, \beta, \gamma \xrightarrow{r} \delta) &= \text{adp}^{\text{RX}}(\alpha \boxplus \beta, \gamma \xrightarrow{r} \delta)\end{aligned}$$

и т. д.

Задачи, связанные с ARX-схемами, встречаются и на криптографических олимпиадах (см. NSUCRYPTO [24, 25]).

1. Определения

Пусть $\mathbb{Z}_2^n$ — линейное пространство двоичных векторов размерности n . Поскольку ARX-схемы комбинируют операции $\oplus$ и $\boxplus$, сопоставим вектору $x = (x_1, \dots, x_n) \in \mathbb{Z}_2^n$ целое число

$$2^{n-1}x_1 + 2^{n-2}x_2 + \dots + 2^0x_n,$$

т. е. x_1 является *наиболее значимым* битом. Под суммой $x + y$ векторов $x, y \in \mathbb{Z}_2^n$ будем подразумевать сложение соответствующих им целых чисел по модулю 2^n , тем самым $x + y \in \{0, \dots, 2^n - 1\}$. Аналогично $-x$ обозначает противоположное относительно сложения по модулю 2^n число к числу, соответствующему x . Обозначим через $(x, z) = (x_1, \dots, x_n, z_1, \dots, z_m)$ *конкатенацию* вектора x с вектором $z \in \mathbb{Z}_2^m$. Положим также $\bar{x} = (x_1 \oplus 1, x_2 \oplus 1, \dots, x_n \oplus 1)$.

Рассматриваемая в работе разностная характеристика относительно сложения по модулю 2^n для XR-преобразования $(x, y) \mapsto (x \oplus y) \lll r$, где $1 \leq r < n$, определяется как

$$\begin{aligned}\text{adp}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) &= \frac{1}{4^n} |\{(x, y) \in (\mathbb{Z}_2^n)^2 \mid \\ &\quad ((x + \alpha) \oplus (y + \beta)) \lll r = \gamma + ((x \oplus y) \lll r)\}|,\end{aligned}$$

а число его невозможных разностей — как

$$\mathcal{N}_n^r = |\{(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3 \mid \text{adp}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) = 0\}|.$$

Для удобства представим тройки $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ в виде восьмеричного слова $\omega \in \{0, \dots, 7\}^n$, определив

$$\omega(\alpha, \beta, \gamma) = (4\alpha_1 + 2\beta_1 + \gamma_1, \dots, 4\alpha_n + 2\beta_n + \gamma_n).$$

Для $\omega(\alpha, \beta, \gamma)_i$, $i \in \{1, \dots, n\}$, используем как численное $4\alpha_i + 2\beta_i + \gamma_i$, так и векторное $(\alpha_i, \beta_i, \gamma_i)$ обозначения.

Таблица 1

Символы в шаблонах

Символ	Значение
[	начало (наиболее значимые биты)
]	конец (наименее значимые биты)
.	0, 1, 2, 3, 4, 5, 6, 7 (любое значение)
e и d	0, 3, 5, 6 и 1, 2, 4, 7 соответственно
$\hat{t}$	$t, t \oplus 3, t \oplus 5$
α_0 и α_1	0, 1, 2, 3 и 4, 5, 6, 7 соответственно
β_0 и β_1	0, 1, 4, 5 и 2, 3, 6, 7 соответственно
γ_0 и γ_1	0, 2, 4, 6 и 1, 3, 5, 7 соответственно
0_6 и 1_7	0, 6 и 1, 7 соответственно
s*	пусто, s, ss, sss, ssss, ...

Для описания слов будем использовать шаблоны — регулярные выражения. В табл. 1 приведены элементы шаблонов и их значения. Пусть $\omega = \omega(\alpha, \beta, \gamma)$ для $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$. Запись $\omega \in \xi$ означает, что ω удовлетворяет шаблону ξ . Приведём примеры:

- $\omega \in [.*d]$ означает $\omega_n \in \{1, 2, 4, 7\}$;
- $\omega \in [.*e11*]$ означает, что $\omega = (\omega_1, \dots, \omega_k, 1, \dots, 1)$, $\omega_k \in \{0, 3, 5, 6\}$ для некоторого $1 \leq k \leq n - 1$;
- $\omega \in [00*2\alpha_0*]$ означает равенство $\omega = (0, \omega_2, \dots, \omega_k, 2, \omega_{k+2}, \dots, \omega_n)$, где $\omega_2, \dots, \omega_k \in \{0, 3, 5\}$, $\omega_{k+2}, \dots, \omega_n \in \{0, 1, 2, 3\}$, $1 \leq k \leq n - 1$. Отметим, что $\omega = (0, 2, \omega_3, \dots, \omega_n)$ при $k = 1$, а при $k = n - 1$ имеем $\omega = (0, \omega_2, \dots, \omega_{n-1}, 2)$, т. е. одна из обозначенных частей слова ω может быть пустой.

Удобно также использовать следующие симметрии.

- 1) Шаблоны ξ и ζ назовём $\alpha\beta$ -эквивалентными, если $\omega(\alpha, \beta, \gamma) \in \xi$ тогда и только тогда, когда $\omega(\beta, \alpha, \gamma) \in \zeta$, для любых $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$;
- 2) Шаблоны ξ и ζ назовём $\overline{\gamma}$ -эквивалентными, если $\omega(\alpha, \beta, \gamma) \in \xi$ тогда и только тогда, когда $\omega(\alpha, \beta, \overline{\gamma}) \in \zeta$, для любых $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$.

Шаблон ξ удовлетворяет шаблону ζ (пишем $\xi \subseteq \zeta$), если для любого $\omega \in \xi$ имеет место $\omega \in \zeta$. Шаблоны ξ и ζ не пересекаются, если не существует слова, удовлетворяющего им обоим.

2. Невозможные разности XR-преобразования

Невозможные разности XR-преобразования описаны в [19, теорема 9]. Шаблоны таких разностей представлены в табл. 2. Через $X.Y$ обозначим шаблон в столбце X и строке Y табл. 2. Будем говорить, что пара (ω, ω')

Таблица 2

Невозможные разности XR-функции

№ пп	1 [.*d00*]	2 [.*e22*]	3 [.*e44*]	4 [.*d66*]	5 [.*d]	6 [.*]	7 [γ ₀ *]
1	[α ₀ *]	[α ₀ *]	[β ₀ *]	[66̂*7.*]	[0*]	[.*d00*]	[.*d]
2	[β ₀ *]	[22̂*]	[44̂*]	[66̂*1γ ₁ *1 ₇]	[1*]	[.*e11*]	
3	[00̂*]	[22̂*3.*]	[44̂*5.*]	[77̂*6.*]			
4	[00̂*1.*]	[22̂*0α ₀ *]	[44̂*0β ₀ *]	[77̂*0γ ₀ *0 ₆]			
5	[00̂*2α ₀ *]	[22̂*5γ ₁ *1 ₇]	[44̂*3γ ₁ *1 ₇]				
6	[00̂*4β ₀ *]	[33̂*]	[55̂*]				
7	[00̂*7]	[33̂*2.*]	[55̂*4.*]				
8	[00̂*7γ ₁ *1 ₇]	[33̂*1α ₀ *]	[55̂*1β ₀ *]				
9	[11̂*]	[33̂*4γ ₀ *0 ₆]	[55̂*2γ ₀ *0 ₆]				
10	[11̂*0.*]						
11	[11̂*3α ₀ *]						
12	[11̂*5β ₀ *]						
13	[11̂*6]						
14	[11̂*6γ ₀ *0 ₆]						

удовлетворяет шаблону $X.Y$, если $\omega \in X.Y$, а ω' удовлетворяет метке столбца X . Например, запись $\omega \in 7.1$ равносильна тому, что $\omega \in [.*d]$, а $(\omega, \omega') \in 7.1$ означает, что $\omega \in [.*d]$ и $\omega' \in [\gamma_0*]$.

Утверждение 1 [19]. Если $\omega = \omega(\alpha, \beta, \gamma)$, $\omega' = \omega(\alpha', \beta', \gamma')$ для некоторых $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, то $\text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta) \xrightarrow{r} (\gamma, \gamma')) = 0$ тогда и только тогда, когда $(\omega, \omega') \in X.Y$ для некоторых столбца X и строки Y табл. 2.

Таблица 3

Символы, удовлетворяющие $\hat{\mathbf{t}}$

$\hat{0}$	$\hat{1}$	$\hat{2}$	$\hat{3}$	$\hat{4}$	$\hat{5}$	$\hat{6}$	$\hat{7}$
0, 3, 5	1, 2, 4	1, 2, 7	0, 3, 6	1, 4, 7	0, 5, 6	3, 5, 6	2, 4, 7
$\mathbf{e} \setminus \{6\}$	$\mathbf{d} \setminus \{7\}$	$\mathbf{d} \setminus \{4\}$	$\mathbf{e} \setminus \{5\}$	$\mathbf{d} \setminus \{2\}$	$\mathbf{e} \setminus \{3\}$	$\mathbf{e} \setminus \{0\}$	$\mathbf{d} \setminus \{1\}$

Далее будем работать только с шаблонами из табл. 2 и их пересечениями. Поскольку они содержат выражения вида $\mathbf{t}\hat{\mathbf{t}}*$, приведём в явном виде очевидные свойства $\hat{\mathbf{t}}$.

Утверждение 2. Значения $\hat{\mathbf{t}}$ для всех $t \in \mathbb{Z}_2^3$ приведены в табл. 3. Кроме того,

- 1) $\hat{\mathbf{t}} \subseteq \mathbf{e}$ ($\hat{\mathbf{t}} \subseteq \mathbf{d}$) тогда и только тогда, когда $t \in \mathbf{e}$ ($t \in \mathbf{d}$), причём $\hat{\mathbf{t}} = \mathbf{e} \setminus \{t \oplus 6\}$ ($\hat{\mathbf{t}} = \mathbf{d} \setminus \{t \oplus 6\}$);
- 2) $\hat{\mathbf{t}} \oplus \hat{\mathbf{s}} = \hat{\mathbf{t}} \oplus s$ для $s \in \mathbb{Z}_2^3$.

Отметим, что в [19, следствия 5, 6] доказано, что

$$\mathcal{N}_n^r < \mathcal{N}_n^1 = \frac{5}{14}8^n - \frac{6}{7}, \quad 2 \leq r < n. \quad (1)$$

3. Число невозможных разностей

Здесь найдём $\mathcal{N}_n^r$, т. е. число невозможных разностей XR-преобразования, разбив раздел на три части. В п. 3.1 столбцы 1–6 табл. 2 представлены в более удобном виде. П. 3.2 классифицирует шаблоны столбцов 1–4 (которые и составляют основную массу шаблонов), а также их пересечения с шаблонами столбца 6. В п. 3.3 получим итоговую формулу для $\mathcal{N}_n^r$.

Для подсчёта числа слов, удовлетворяющих рассматриваемым шаблонам, используется

Лемма 1. Для любых попарно различных целых $a, b, c > 0$ и любого целого $m > 0$ справедливы равенства

$$\begin{aligned} 1) \quad & \sum_{i=0}^m a^i b^{m-i} = \frac{1}{a-b}(a^{m+1} - b^{m+1}); \\ 2) \quad & \sum_{i=0}^m \sum_{j=0}^{m-i} a^i b^j c^{m-i-j} = \frac{a^{m+2}}{(a-b)(a-c)} + \frac{b^{m+2}}{(b-a)(b-c)} + \frac{c^{m+2}}{(c-a)(c-b)}. \end{aligned}$$

Доказательство. П. 1 очевиден в силу формулы

$$(a-b)(a^m b^0 + a^{m-1} b^1 + \dots + a^0 b^m) = a^{m+1} - b^{m+1}.$$

Для доказательства п. 2 воспользуемся п. 1:

$$\begin{aligned} \sum_{i=0}^m \sum_{j=0}^{m-i} a^i b^j c^{m-i-j} &= \sum_{i=0}^m a^i \sum_{j=0}^{m-i} b^j c^{m-i-j} = \\ &= \frac{1}{b-c} \sum_{i=0}^m a^i (b^{m-i+1} - c^{m-i+1}) = \frac{b}{b-c} \sum_{i=0}^m a^i b^{m-i} + \\ &+ \frac{c}{c-b} \sum_{i=0}^m a^i c^{m-i} = \frac{b(a^{m+1} - b^{m+1})}{(b-c)(a-b)} + \frac{c(a^{m+1} - c^{m+1})}{(c-b)(a-c)} = \\ &= \frac{ba^{m+1}}{(a-b)(b-c)} + \frac{ca^{m+1}}{(a-c)(c-b)} + \frac{b^{m+2}}{(b-c)(b-a)} + \frac{c^{m+2}}{(c-a)(c-b)}, \end{aligned}$$

при этом сумма первых двух слагаемых в последней строке равна

$$\begin{aligned} \frac{a^{m+1}}{b-c} \left(\frac{b}{a-b} - \frac{c}{a-c} \right) &= \frac{a^{m+1}(ba - bc - ca + cb)}{(b-c)(a-b)(a-c)} = \\ &= \frac{a^{m+2}(b-c)}{(b-c)(a-b)(a-c)} = \frac{a^{m+2}}{(a-b)(a-c)}. \end{aligned}$$

Лемма 1 доказана.

3.1. Упрощение таблицы 2. Начнём со следующей леммы.

Лемма 2. Пусть $\omega = \omega(\alpha, \beta, \gamma)$ для $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$. Тогда

- 1) если $\omega \in [0\hat{0}*\alpha_0*] ([1\hat{1}*\alpha_0*])$, то ω удовлетворяет 1.3, 1.4 или 1.5 (1.9, 1.10 или 1.11);
- 2) если $\omega \in [0\hat{0}*\beta_0*] ([1\hat{1}*\beta_0*])$, то ω удовлетворяет 1.3, 1.4 или 1.6 (1.9, 1.10 или 1.12);
- 3) если $\omega \in [2\hat{2}*\alpha_0*] ([3\hat{3}*\alpha_0*])$, то ω удовлетворяет 2.2, 2.3 или 2.4 (2.6, 2.7 или 2.8);
- 4) если $\omega \in [4\hat{4}*\beta_0*] ([5\hat{5}*\beta_0*])$, то ω удовлетворяет 3.2, 3.3 или 3.4 (3.6, 3.7 или 3.8).

ДОКАЗАТЕЛЬСТВО. 1) Если $\omega \notin 1.3$, то определим ω^α как часть ω , идущую после $0\hat{0}*$ и удовлетворяющую α_0* , причём $\omega_0^\alpha \notin \hat{0}$. В этом случае $\omega_0^\alpha \in \{1, 2\}$, так как 0 и 3 удовлетворяют $\hat{0}$ (см. табл. 3), но тогда ω удовлетворяет либо 1.4, либо 1.5.

Шаблон $[1\hat{1}*\alpha_0*]$ $\bar{\gamma}$ -эквивалентен уже рассмотренному, поэтому удовлетворяет шаблонам, $\bar{\gamma}$ -эквивалентным 1.3, 1.4 и 1.5, которыми являются 1.9, 1.10 и 1.11 соответственно.

2) Достаточно заметить, что шаблоны $[0\hat{0}*\beta_0*] ([1\hat{1}*\beta_0*])$ $\alpha\beta$ -эквивалентны рассмотренным в п. 1, а шаблоны 1.3, 1.4 и 1.6 (1.9, 1.10 и 1.12) $\alpha\beta$ -эквивалентны 1.3, 1.4 и 1.5 (1.9, 1.10 и 1.11) соответственно.

3) Аналогично п. 1, если $\omega \notin 2.2$, определим ω^α такое, что $\omega_0^\alpha \notin \hat{2}$, т. е. $\omega_0^\alpha \in \{0, 3\}$, поскольку 1 и 2 удовлетворяют $\hat{2}$, но тогда ω удовлетворяет либо 2.4, либо 2.3.

4) Шаблоны этого пункта $\alpha\beta$ -эквивалентны соответствующим шаблонам п. 3 (см. также столбцы 2, 3 табл. 2). Лемма 2 доказана.

Лемма 3. Если $\omega = \omega(\alpha, \beta, \gamma)$, $\omega' = \omega(\alpha', \beta', \gamma')$ для $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, то (ω, ω') удовлетворяет шаблону из столбцов 1–6 табл. 2 тогда и только тогда, когда либо (ω, ω') , либо $(\omega(\alpha, \beta, \bar{\gamma}), \omega')$ удовлетворяет некоторому шаблону из табл. 4. Более того, среди шаблонов одного столбца табл. 4 и $\bar{\gamma}$ -эквивалентных к ним никакие два не пересекаются.

ДОКАЗАТЕЛЬСТВО. Ввиду условия леммы из пары $\bar{\gamma}$ -эквивалентных шаблонов одного столбца табл. 2 достаточно рассмотреть только один. Таким образом, удаляем шаблоны 1.9–14, 2.6–9, 3.6–9, 4.3–4, 5.2 и 6.2. Замыкание оставшихся шаблонов относительно $\bar{\gamma}$ -эквивалентности составит столбцы 1–6 табл. 2.

По лемме 2 шаблон 1.1 можно преобразовать в $[2\alpha_0*]$ и $\bar{\gamma}$ -эквивалентный последнему $[3\alpha_0*]$ (слова, удовлетворяющие $[0_1\alpha_0*]$, удовлетворяют одному из 1.3–1.14); шаблон 1.2 — в $[4\beta_0*]$ и $\bar{\gamma}$ -эквивалентный

Таблица 4

Шаблоны столбцов 1–6 табл. 2 с точностью до $\bar{\gamma}$ -эквивалентности

№ пп	1 [. *d00*]	2 [. *e22*]	3 [. *e44*]	4 [. *d66*]	5 [. *d]	6 [. *]
1	$[2\alpha_0*]$	$[0\alpha_0*]$	$[0\beta_0*]$	$[6\hat{6}*7.*]$	$[0*]$	$[. *d00*]$
2	$[4\beta_0*]$	$[2\hat{2}*]$	$[4\hat{4}*]$	$[6\hat{6}*1\gamma_1*1_7]$		
3	$[0\hat{0}*]$	$[2\hat{2}*3.*]$	$[4\hat{4}*5.*]$			
4	$[0\hat{0}*1.*]$	$[2\hat{2}*0\alpha_0*]$	$[4\hat{4}*0\beta_0*]$			
5	$[0\hat{0}*2\alpha_0*]$	$[2\hat{2}*5\gamma_1*1_7]$	$[4\hat{4}*3\gamma_1*1_7]$			
6	$[0\hat{0}*4\beta_0*]$					
7	$[0\hat{0}*7]$					
8	$[0\hat{0}*7\gamma_1*1_7]$					

ему $[5\beta_0*]$; шаблон 2.1 — в $[0\alpha_0*]$ и $\bar{\gamma}$ -эквивалентный ему $[1\alpha_0*]$; шаблон 3.1 — в $[0\beta_0*]$ и $\bar{\gamma}$ -эквивалентный ему $[1\beta_0*]$.

Таким образом, шаблоны одного столбца табл. 4 попарно не пересекаются, что также имеет место при добавлении $\bar{\gamma}$ -эквивалентных им шаблонов: любые два отличаются как минимум одним фиксированным символом. Лемма 3 доказана.

3.2. Классификация шаблонов из столбцов 1–4 таблицы 2.

Скомбинируем шаблоны столбцов 1–4 табл. 2 и их пересечения с шаблонами 6.1 и 6.2, объединяя те из них, которым удовлетворяет одинаковое число слов. В результате получим табл. 5 и 6, каждая строка которых содержит тип шаблона, число шаблонов этого типа в табл. 2, число слов, удовлетворяющих шаблону этого типа, и ограничение, при котором указанная формула справедлива. Если ограничение не выполняется, то не существует слов длины $n - r$, удовлетворяющих соответствующему шаблону. Отметим, что табл. 5 почти совпадает с [19, табл. 5].

Лемма 4. Табл. 5 классифицирует шаблоны из столбцов 1–4 табл. 2.

ДОКАЗАТЕЛЬСТВО. Будем пользоваться табл. 4, умножая итоговое число шаблонов на 2 в силу $\bar{\gamma}$ -эквивалентности (см. лемму 3). Также если слово удовлетворяет некоторому из шаблонов, то все «звёздочки» расставляются однозначно. Действительно, после $t\hat{t}*$ всегда идёт символ, не удовлетворяющий t (либо его нет).

1. Тип $[2\alpha_0*]$. К нему относятся 1.1, 1.2, 2.1 и 3.1, что даёт $2 \cdot 4$ шаблонов, каждому из которых удовлетворяет 4^{n-r-1} слов.

2. Тип $[0\hat{0}*1.*]$. К нему относятся 1.4, 2.3, 3.3 и 4.1, что даёт $2 \cdot 4$ шаблонов. Каждому из них удовлетворяет

Таблица 5

Классификация шаблонов из столбцов 1–4 табл. 2

№	Тип шаблона	#	Число слов	Ограничение
1	$[2\alpha_0*]$	8	4^{n-r-1}	$n-r \geq 1$
2	$[0\hat{0}*1.*]$	8	$\frac{1}{5}(8^{n-r-1} - 3^{n-r-1})$	$n-r \geq 1$
3	$[0\hat{0}*2\alpha_0*]$	8	$4^{n-r-1} - 3^{n-r-1}$	$n-r \geq 1$
4	$[0\hat{0}*7\gamma_1*1_7]$	8	$2(4^{n-r-2} - 3^{n-r-2})$	$n-r \geq 2$
5	$[0\hat{0}*]$	6	3^{n-r-1}	$n-r \geq 1$
6	$[0\hat{0}*7]$	2	3^{n-r-2}	$n-r \geq 2$

$$8^0 \cdot 3^{n-r-2} + 8^1 \cdot 3^{n-r-3} + \dots + 8^{n-r-2} \cdot 3^0 = \sum_{i=0}^{n-r-2} 8^i \cdot 3^{n-r-2-i}$$

слов, что по лемме 1 равно $\frac{1}{5}(8^{n-r-1} - 3^{n-r-1})$. Если $n-r < 2$, то шаблону не может удовлетворять ни одно слово. Однако $\frac{1}{5}(8^{1-1} - 3^{1-1}) = 0$, поэтому можно считать, что $n-r \geq 1$.

3. Тип $[0\hat{0}*2\alpha_0*]$. К нему относятся 1.5, 1.6, 2.4 и 3.4, что даёт 2·4 шаблонов. Каждому удовлетворяет $\sum_{i=0}^{n-r-2} 4^i \cdot 3^{n-r-2-i}$ слов, что по лемме 1 равно $4^{n-r-1} - 3^{n-r-1}$. Аналогично при $n-r < 2$ шаблону не удовлетворяет ни одно слово. Однако $4^{1-1} - 3^{1-1} = 0$, поэтому формула верна при $n-r \geq 1$.

4. Тип $[0\hat{0}*7\gamma_1*1_7]$. К нему относятся 1.8, 2.5, 3.5 и 4.2, что даёт 2·4 шаблонов. Каждому удовлетворяет $2 \cdot \sum_{i=0}^{n-r-3} 4^i \cdot 3^{n-r-3-i}$ слов (делим шаблон на два, заканчивающихся на 1 и 7). По лемме 1 получаем $2(4^{n-r-2} - 3^{n-r-2})$ слов. При $n-r < 3$ шаблону не удовлетворяет ни одно слово. Однако $2(4^{2-2} - 3^{2-2}) = 0$, поэтому формула верна при $n-r \geq 2$.

Таблица 6

Классификация пересечений шаблонов из столбцов 1–4 с шаблонами столбца 6 табл. 2

№	Тип шаблона	#	Число слов	Ограничение
1	$[2\alpha_0*1_200*]$	16	$\frac{2}{3}(4^{n-r-2} - 1)$	$n-r \geq 2$
2	$[200*]$	8	1	$n-r \geq 2$
3	$[0\hat{0}*1.*d00*]$	16	$\frac{4}{35}8^{n-r-2} - \frac{2}{5}3^{n-r-2} + \frac{2}{7}$	$n-r \geq 2$
4	$[0\hat{0}*100*]$	22	$\frac{1}{2}(3^{n-r-2} - 1)$	$n-r \geq 2$
5	$[0\hat{0}*11*]$	6	$\frac{1}{2}(3^{n-r-1} - 1)$	$n-r \geq 1$
6	$[0\hat{0}*2\alpha_0*1_200*]$	24	$\frac{2}{3}4^{n-r-2} - 3^{n-r-2} + \frac{1}{3}$	$n-r \geq 2$

5. Тип $[0\hat{0}^*]$. К нему относятся 1.3, 2.2 и 3.2, что даёт $2 \cdot 3$ шаблона. Ему удовлетворяет 3^{n-r-1} слов, где $n - r \geq 1$.

6. Тип $[0\hat{0}^*7]$. К нему относится 1.7, которому удовлетворяет 3^{n-r-2} слов, где $n - r \geq 2$.

Все шаблоны столбцов 1–4 табл. 4 рассмотрены. Лемма 4 доказана.

Лемма 5. *Табл. 6 содержит классификацию пересечений шаблонов из столбцов 1–4 с шаблонами столбца 6 табл. 2.*

ДОКАЗАТЕЛЬСТВО. Метки столбцов 1–4 не пересекаются друг с другом, при этом все они удовлетворяют метке $[.*]$ столбца 6, поэтому при классификации можно рассматривать шаблоны независимо друг от друга, как непересекающиеся (что верно с учётом ω' -частей).

Классификацию пересечений проведём по табл. 5, но в некоторых случаях будем обращаться и к табл. 2. Нас интересуют пересечения шаблонов из столбцов 1–4 с шаблонами $[.*d00^*]$ (6.1) и $[.*e11^*]$ (6.2). Отметим также следующие равенства (см. табл. 1):

$$\begin{aligned} \alpha_0 \cap e &= 0_3, & \beta_0 \cap e &= 0_5, & \gamma_0 \cap e &= 0_6, & \gamma_1 \cap e &= 3_5, \\ \alpha_0 \cap d &= 1_2, & \beta_0 \cap d &= 1_4, & \gamma_0 \cap d &= 2_4, & \gamma_1 \cap d &= 1_7. \end{aligned}$$

1. $[2\alpha_0^*]$. Пересечения: $[2\alpha_0^*1_200^*]$, $[2\alpha_0^*0_311^*]$ и $[200^*]$, других пересечений здесь нет. Таким образом, добавляем $8 \cdot 2$ шаблонов в строку 1 и 8 шаблонов в строку 2.

Далее заметим, что в $[.*d00^*]$ и $[.*e11^*]$ имеется смена чётности e/d со стороны младших разрядов: $1 \in d$ и $0 \in e$. При этом все оставшиеся шаблоны из классификации в старших битах содержат подстроку $\hat{tt}^*$, символы которой удовлетворяют либо e , либо d (см. утверждение 2). Это означает, что шаблоны типа 5 табл. 5 не пересекаются с 6.1 и 6.2. Также все шаблоны типа 6, содержащиеся в табл. 2, не заканчиваются ни на 0, ни на 1, т. е. также не имеют пересечений с 6.1 и 6.2. Все оставшиеся шаблоны после $\hat{tt}^*$ имеют смену чётности e/d или d/e , поэтому e/d из 6.1 и 6.2 может соответствовать символ, начиная с последнего в части $\hat{tt}^*$.

2. $[0\hat{0}^*1.*]$. Пересечения: $[0\hat{0}^*1.*d00^*]$, $[0\hat{0}^*1.*e11^*]$ и $[0\hat{0}^*100^*]$, т. е. добавляем $8 \cdot 2$ шаблонов в строку 3 и 8 шаблонов в строку 4. Однако если шаблон типа 2 содержит после $\hat{tt}^*$ символы 0 или 1, то появится ещё один тип пересечения: $[0\hat{0}^*11^*]$. В табл. 2 таких шаблонов два: 1.4 и 1.10, т. е. добавляем 2 шаблона в строку 5.

3. $[0\hat{0}^*2\alpha_0^*]$. Пересечениями в этом случае являются $[0\hat{0}^*2\alpha_0^*0_311^*]$, $[0\hat{0}^*2\alpha_0^*1_200^*]$ и $[0\hat{0}^*200^*]$, что добавляет $8 \cdot 2$ шаблонов в строку 6 и 8 шаблонов в строку 4. Аналогично если шаблон типа 3 содержит после подстроки $\hat{tt}^*$ символы 0 или 1, то появится ещё один тип пересечения: $[2\hat{2}^*00^*]$. В табл. 2 таких шаблонов четыре: 2.4, 2.8, 3.4 и 3.8, т. е. добавляем 4 шаблона в строку 5.

4. $[0\hat{0}*7\gamma_1*1_7]$. Пересечение — $[0\hat{0}*7\gamma_1*3_511*]$, так как он не пересекается с 6.1. Добавляем 8 шаблонов в строку 6. Если шаблон типа 4 после подстроки $\hat{t}\hat{t}*$ содержит символ из e/d и оканчивается на $1/0$, то появится пересечение $[2\hat{2}*511*]$. В табл. 2 таких шаблонов четыре: 2.5, 2.9, 3.5 и 3.9, т. е. добавляем 4 шаблона в строку 4. Ещё один случай дают шаблоны 4.2 и 4.4, в которых после $\hat{t}\hat{t}*$ и в конце стоит 1 (4.2) и 0 (4.4). Это даёт 2 пересечения вида $[6\hat{6}*111*]$, которые добавляются в строку 4.

Тем самым перечислены все типы шаблонов и их число. Они представлены в столбцах 2 и 3 табл. 6. Далее найдём число слов, удовлетворяющих каждому из указанных типов шаблонов. Отметим, что «звёздочки» $*$ в таких словах расставляются однозначно. Действительно, «звёздочка» после $\hat{t}\hat{t}$ всегда оканчивается символом, не принадлежащим $\hat{t}$. «Звёздочка» со стороны младших разрядов также оканчивается символом, отличным от $0/1$. С учётом фиксированной длины слов $n - r$ даже при наличии в шаблоне третьей «звёздочки» её элементы определяются однозначно. Далее будем пользоваться леммой 1: п. 1 в случае двух «звёздочек» и п. 2 в случае трёх.

Шаблон $[2\alpha_0*1_200*]$ разбивается на два непересекающихся подшаблона $[2\alpha_0*100*]$ и $[2\alpha_0*200*]$. Здесь три фиксированных символа, т. е. взяв $m = n - r - 3$, получим, что число слов, удовлетворяющих каждому из шаблонов, равно $\sum_{i=0}^m 4^i \cdot 1^{m-i}$. В силу леммы 1 получаем $\frac{1}{4-1}(4^{m+1} - 1) = \frac{1}{3}(4^{n-r-2} - 1)$ слов на каждый подшаблон. Однако, несмотря на ограничение $n - r \geq 3$ (в шаблоне три фиксированных символа), $\frac{1}{3}(4^{n-r-2} - 1) = 0$ при $n - r = 2$, т. е. указанная формула справедлива при $n - r \geq 2$.

Шаблону $[200*]$ удовлетворяет ровно одно слово с условием $n - r \geq 2$, поскольку здесь имеется два фиксированных символа.

Шаблон $[0\hat{0}*1.*d00*]$ по значению d разделим на 4 непересекающихся подшаблона, в каждом из которых 4 фиксированных символа. При $m = n - r - 4$ получим $\sum_{i=0}^m \sum_{j=0}^{m-i} 8^i \cdot 3^j \cdot 1^{m-i-j}$ слов на каждый подшаблон равно.

По лемме 1 получим следующее число слов на подшаблон:

$$\frac{8^{m+2}}{(8-3)(8-1)} + \frac{3^{m+2}}{(3-8)(3-1)} + \frac{1^{m+2}}{(1-8)(1-3)} = \frac{8^{n-r-2}}{35} - \frac{3^{n-r-2}}{10} + \frac{1}{14}.$$

При $n - r \in \{2, 3\}$ эта сумма обращается в нуль: $\frac{8}{35} - \frac{3}{10} + \frac{1}{14} = \frac{16-21+5}{35 \cdot 2} = 0$ и $\frac{1}{35} - \frac{1}{10} + \frac{1}{14} = \frac{2-7+5}{35 \cdot 2} = 0$, так что ограничение можно ослабить, взяв $n - r \geq 2$.

Аналогично для шаблонов $[0\hat{0}*100*]$ и $[0\hat{0}*11*]$ получаем $\frac{1}{2}(3^{n-r-2} - 1)$ и $\frac{1}{2}(3^{n-r-1} - 1)$ слов с ограничениями $n - r \geq 2$ и $n - r \geq 1$ соответственно.

Шаблон $[0\widehat{0}*2\alpha_0*1_200*]$ разделим на два подшаблона по значению символа 1_2 и аналогично рассмотренному случаю для каждого из них по лемме 1 получим $\frac{1}{3}4^{n-r-2} - \frac{1}{2}3^{n-r-2} + \frac{1}{6}$ слов. Эта сумма также равна нулю при $n - r \in \{2, 3\}$, так что $n - r \geq 2$. Лемма 5 доказана.

3.3. Число невозможных разностей. В следующей теореме приводится точное значение $\mathcal{N}_n^r$ для всех возможных n и r .

Теорема 1. Пусть $n \geq 2$ и $1 \leq r < n$. Тогда

$$\begin{aligned} \mathcal{N}_n^r = & \frac{1}{7}8^n + \frac{3}{7}4^r \cdot 8^{n-r} - \frac{1}{7}8^r + \frac{1}{14}4^r + \\ & + \left(\frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{2}{21} \right) \left(\frac{1}{14}8^r - \frac{1}{12}4^r - \frac{5}{21} \right). \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Пусть $T_{j_1, \dots, j_s}^{i_1, \dots, i_t}(n, r)$, $t \geq 1$, $s \geq 0$, — множество слов (ω, ω') , $\omega = \omega(\alpha, \beta, \gamma)$, $\omega' = \omega(\alpha', \beta', \gamma')$, $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, удовлетворяющих шаблону из столбцов $i_1, \dots, i_t \in \{1, \dots, 7\}$ и не удовлетворяющих шаблону из столбцов $j_1, \dots, j_s \in \{1, \dots, 7\}$ табл. 2. Вычислим $\mathcal{N}_n^r$ по формуле

$$\mathcal{N}_n^r = |T^6(n, r)| + |T^7(n, r)| + |T_{6,7}^5(n, r)| + |T_{5,6,7}^{1-4}(n, r)|.$$

1. $|T^6(n, r)|$. По лемме 1 число различных ω -частей у 6.1 и 6.2 равно $2 \cdot 4 \cdot \frac{1}{8-1}(8^{n-r-1} - 1)$. Это выражение равно нулю при $n - r = 1$, поэтому его можно использовать без ограничений. Таким образом,

$$|T^6(n, r)| = 8^r \cdot \frac{8}{7}(8^{n-r-1} - 1) = \frac{1}{7}8^n - \frac{8}{7}8^r.$$

2. $|T^7(n, r)|$. Заметим, что $|T^7(n, r)| = 4^r \cdot 4 \cdot 8^{n-r-1} = \frac{1}{2}4^r \cdot 8^{n-r}$. Осталось найти пересечение $T^7(n, r)$ и $T^6(n, r)$. Их ω' -части пересекаются по $[\gamma_0*]$, а ω -части — по $[.*e11*]$. Таким образом,

$$\begin{aligned} |T^7(n, r)| &= |T^7(n, r)| - |T^7(n, r) \cap T^6(n, r)| = \\ &= \frac{1}{2}4^r \cdot 8^{n-r} - \frac{4^r \cdot 4}{7}(8^{n-r-1} - 1) = \frac{3}{7}4^r \cdot 8^{n-r} + \frac{4}{7}4^r. \end{aligned}$$

Формула верна при всех $r \geq 1$ и $n - r \geq 1$, т. е. без ограничений.

3. $|T_{6,7}^5(n, r)|$. Поскольку ω -части 5.1 и 5.2 не пересекаются ни с 6.1, ни с 6.2, то $T_{6,7}^5(n, r) = T_7^5(n, r)$. При этом $|T^5(n, r)| = 4 \cdot 8^{r-1} \cdot 2 = 8^r$. Найдём $T^5(n, r) \cap T^7(n, r)$: их ω' -части пересекаются по $[\gamma_0*2_4]$ (пересечение γ_0 и d равно 2_4), а ω -части — по $[1*]$. Тем самым

$$\begin{aligned} |T_{6,7}^5(n, r)| &= |T_7^5(n, r)| = |T^5(n, r)| - |T^5(n, r) \cap T^7(n, r)| = \\ &= 8^r - 2 \cdot 4^{r-1} \cdot 1 = 8^r - \frac{1}{2}4^r. \end{aligned}$$

В итоге получаем первую часть формулы для $\mathcal{N}_n^r$:

$$|T^6(n, r)| + |T_6^7(n, r)| + |T_{6,7}^5(n, r)| = \frac{1}{7}8^n + \frac{3}{7}4^r \cdot 8^{n-r} - \frac{1}{7}8^r + \frac{1}{14}4^r.$$

4. Осталось показать, что $|T_{5,6,7}^{1-4}(n, r)|$ равна произведению скобок в формуле для $\mathcal{N}_n^r$. Обозначим через $P(r)$ число слов, удовлетворяющих метке каждого из столбцов 1–4, т. е. $P(r) = \frac{4}{7}(8^{r-1} - 1)$ по лемме 1. Дополнительных ограничений на r нет, так как $P(1) = 0$. Найдём число $|T_{5,6,7}^{1-4}(n, r)|$ в несколько шагов. Положим $n - r \geq 2$, а случай $r = n - 1$ рассмотрим в конце доказательства.

4.1. $|T^{1-4}(n, r)|$. Применим классификацию из табл. 5 (см. лемму 4):

$$\begin{aligned} |T^{1-4}(n, r)| &= P(r) \left(8 \cdot 4^{n-r-1} + 8 \cdot \frac{1}{5}(8^{n-r-1} - 3^{n-r-1}) + 2 \cdot 3^{n-r-2} + \right. \\ &\quad \left. + 8(4^{n-r-1} - 3^{n-r-1}) + 8 \cdot 2(4^{n-r-2} - 3^{n-r-2}) + 6 \cdot 3^{n-r-1} \right) = \\ &= \frac{1}{5}P(r)(8^{n-r} + 25 \cdot 4^{n-r} - 124 \cdot 3^{n-r-2}). \end{aligned}$$

4.2. $|T_6^{1-4}(n, r)|$. Метки столбцов 1–4 удовлетворяют метке $[.*]$ столбца 6, поэтому применим классификацию пересечений из табл. 6:

$$\begin{aligned} |T_6^{1-4}(n, r)| &= |T^{1-4}(n, r)| - P(r) \left(16 \cdot \frac{2}{3}(4^{n-r-2} - 1) + 8 \cdot 1 + \right. \\ &\quad \left. + 16 \cdot \frac{2}{35}(2 \cdot 8^{n-r-2} - 7 \cdot 3^{n-r-2} + 5) + 22 \cdot \frac{1}{2}(3^{n-r-2} - 1) + \right. \\ &\quad \left. + 6 \cdot \frac{1}{2}(3^{n-r-1} - 1) + 24 \cdot \frac{1}{3}(2 \cdot 4^{n-r-2} - 3^{n-r-1} + 1) \right) = \\ &= P(r) \left(\frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{86}{21} \right). \end{aligned}$$

4.3. $|T_{5,6}^{1-4}(n, r)|$. Метки столбцов 1 и 4 не пересекаются с меткой столбца 5, а метки столбцов 2 и 3 удовлетворяют метке столбца 5. Далее, ω -части $[0*]$ (5.1) и $[1*]$ (5.2) удовлетворяют 2.1 и 3.1. Таким образом,

$$|T_{5,6}^{1-4}(n, r)| = |T^{1-4}(n, r)| - 2 \cdot 2 \cdot P(r) = |T^{1-4}(n, r)| - 4P(r).$$

Более того, шаблоны из столбцов 5 и 6 не пересекаются, поэтому такой способ подсчёта работает и для $|T_{5,6}^{1-4}(n, r)|$:

$$|T_{5,6}^{1-4}(n, r)| = |T_6^{1-4}(n, r)| - 4P(r) = P(r) \left(\frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{2}{21} \right).$$

4.4. $|T_{5,6,7}^{1-4}(n, r)|$. Пусть $|T_{5,6}^{1-4}(n, r)| = P(r)Q(n - r)$, где

$$Q(n - r) = \frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{2}{21}.$$

Метка $[\gamma_0*]$ столбца 7 имеет пересечения типа $[\gamma_0*2_400*]$ с метками любого из столбцов 1–4, что даёт $\frac{2}{3}(4^{r-1} - 1)$ слов (лемма 1).

Далее покажем, что $(\omega, \omega') \in T_{5,6}^{1-4}(n, r)$ тогда и только тогда, когда $(\omega(\alpha, \beta, \bar{\gamma}), \omega') \in T_{5,6}^{1-4}(n, r)$. Действительно, ω -части шаблонов из столбцов 1–6 замкнуты относительно $\bar{\gamma}$ -эквивалентности (лемма 3), а значит, замкнуты и их объединения, пересечения и разности. Пусть $(\omega, \omega') \in T_{5,6}^{1-4}(n, r)$. Тогда одно из слов ω и $\omega(\alpha, \beta, \bar{\gamma})$ будет удовлетворять ω -части $[.*d]$ 7.1, а второе — нет. Действительно, инвертировав γ -бит у символа, удовлетворяющего e , получим символ, удовлетворяющий d (и наоборот). Таким образом, среди всех слов из $T_{5,6}^{1-4}(n, r)$, ω' -части которых удовлетворяют ω' -части 7.1, ω -части 7.1 удовлетворяет ровно $\frac{1}{2}Q(n - r)$ ω -частей слов. Следовательно,

$$\begin{aligned} |T_{5,6,7}^{1-4}(n, r)| &= P(r)Q(n - r) - \frac{2}{3}(4^{r-1} - 1) \cdot \frac{1}{2}Q(n - r) = \\ &= \left(\frac{4}{7}(8^{r-1} - 1) - \frac{1}{3}(4^{r-1} - 1) \right) Q(n - r) = \\ &= \left(\frac{1}{14}8^r - \frac{1}{12}4^r - \frac{5}{21} \right) Q(n - r). \end{aligned}$$

Тем самым получена искомая формула для $\mathcal{N}_n^r$.

Осталось разобрать случай $n - r = 1$, от которого зависит только $|T_{5,6,7}^{1-4}(n, r)|$. Отметим, что в этом случае нет слов, удовлетворяющих шаблонам из столбца 6, т. е. достаточно найти $|T_5^{1-4}(n, n - 1)|$ и далее воспользоваться п. 4.4 для нахождения числа $|T_{5,6,7}^{1-4}(n, n - 1)|$, которое зависит только от мощности $T_{5,6}^{1-4}(n, n - 1) = T_5^{1-4}(n, n - 1)$. Нетрудно видеть, что из ω -частей столбца 1 подходят $[0_1]$, $[2_3]$ и $[4_5]$ (6 слов), а также $[2_3]$ из столбца 2 (2 слова) и $[4_5]$ из столбца 3 (2 слова), исключаем $[0_1]$ из столбцов 2 и 3, потому что они удовлетворяют 5.1 или 5.2. Итого,

$$|T_5^{1-4}(n, n - 1)| = 6P(n - 1) + 2P(n - 1) + 2P(n - 1) = 10P(n - 1),$$

что также совпадает с $|T_{5,6}^{1-4}(n, n - 1)|$. Однако

$$\begin{aligned} Q(1) &= \frac{6}{35} \cdot 8 + \frac{10}{3} \cdot 4 - \frac{8}{5} \cdot 3 + \frac{2}{21} = \frac{48 \cdot 3 + 40 \cdot 35 - 24 \cdot 21 + 10}{3 \cdot 5 \cdot 7} = \\ &= \frac{144 + 1400 - 504 + 10}{105} = \frac{1050}{105} = 10, \end{aligned}$$

так что формула $|T_{5,6}^{1-4}(n, r)| = P(r)Q(n-r)$ справедлива и при $n-r=1$, а значит, 4.4 даст то же самое $|T_{5,6,7}^{1-4}(n, r)|$. Теорема 1 доказана.

Зафиксировав r или $n-r$, можно получить более простое выражение для $\mathcal{N}_n^r$. Напомним также, что $\mathcal{N}_n^1 = \frac{5}{14}8^n - \frac{6}{7}$ (см. (1)).

Следствие 1. *Имеют место следующие равенства:*

$$\begin{aligned}\mathcal{N}_n^2 &= \frac{289}{1120}8^n + \frac{5}{8}4^n - \frac{8}{15}3^n - \frac{54}{7}, \quad n \geq 3, \\ \mathcal{N}_n^3 &= \frac{1853}{8960}8^n + \frac{155}{96}4^n - \frac{248}{135}3^n - \frac{1378}{21}, \quad n \geq 4, \\ \mathcal{N}_n^{n-3} &= \frac{5}{28}8^n + \frac{99}{32}4^n - \frac{430}{7}, \quad n \geq 4, \\ \mathcal{N}_n^{n-2} &= \frac{11}{56}8^n + \frac{35}{24}4^n - \frac{250}{21}, \quad n \geq 3, \\ \mathcal{N}_n^{n-1} &= \frac{3}{14}8^n + \frac{2}{3}4^n - \frac{50}{21}, \quad n \geq 2.\end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Заметим, что

$$\frac{1}{14}8^r - \frac{1}{12}4^r - \frac{5}{21} = \begin{cases} 3 & \text{при } r = 2, \\ 31 & \text{при } r = 3, \end{cases}$$

поэтому

$$\begin{aligned}\mathcal{N}_n^2 &= \left(\frac{1}{7} + \frac{3}{7 \cdot 4} + \frac{9}{7 \cdot 4 \cdot 40}\right)8^n + \frac{5}{8}4^n - \frac{8}{15}3^n - \\ &\quad - \frac{8^2}{7} + \frac{4^2}{7 \cdot 2} + \frac{2}{7} = \frac{289}{1120}8^n + \frac{5}{8}4^n - \frac{8}{15}3^n - \frac{54}{7}, \\ \mathcal{N}_n^3 &= \left(\frac{1}{7} + \frac{3}{7 \cdot 8} + \frac{6 \cdot 31}{5 \cdot 7 \cdot 8^3}\right)8^n + \frac{10 \cdot 31}{3 \cdot 4^3}4^n - \frac{8 \cdot 31}{5 \cdot 27}3^n - \\ &\quad - \frac{8^3}{7} + \frac{4^3}{7 \cdot 2} + \frac{62}{21} = \frac{1853}{8960}8^n + \frac{155}{96}4^n - \frac{248}{135}3^n - \frac{1378}{21}.\end{aligned}$$

Аналогично

$$\frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{2}{21} = \begin{cases} 2 \cdot 3 \cdot 43 = 258 & \text{при } r = n-3, \\ 50 & \text{при } r = n-2, \\ 10 & \text{при } r = n-1, \end{cases}$$

поэтому

$$\begin{aligned}\mathcal{N}_n^{n-3} &= \left(\frac{1}{7} - \frac{1}{7 \cdot 8^3} + \frac{129}{7 \cdot 8^3}\right)8^n + \left(\frac{3 \cdot 8}{7} + \frac{1}{14 \cdot 4^3} - \frac{43}{2 \cdot 4^3}\right)4^n - \frac{430}{7}, \\ \mathcal{N}_n^{n-2} &= \left(\frac{1}{7} - \frac{1}{7 \cdot 8^2} + \frac{25}{7 \cdot 8^2}\right)8^n + \left(\frac{3 \cdot 4}{7} + \frac{1}{14 \cdot 16} - \frac{25}{6 \cdot 16}\right)4^n - \frac{250}{21},\end{aligned}$$

$$\mathcal{N}_n^{n-1} = \left(\frac{1}{7} - \frac{1}{7 \cdot 8} + \frac{5}{7 \cdot 8}\right) 8^n + \left(\frac{6}{7} + \frac{1}{14 \cdot 4} - \frac{5}{6 \cdot 4}\right) 4^n - \frac{50}{21}.$$

Следствие 1 доказано.

Значения $\mathcal{N}_n^r$ при $n \leq 9$ представлены в табл. 7, где также приведены доли невозможных разностей по отношению ко всем 8^n разностям.

Таблица 7

Значения $\mathcal{N}_n^r$ при $2 \leq n \leq 9$

n	r	$\mathcal{N}_n^r$	$\mathcal{N}_n^r/8^n, \%$	n	r	$\mathcal{N}_n^r$	$\mathcal{N}_n^r/8^n, \%$
2	1	22	34,37500	7	4	425118	20,27121
3	1	182	35,54688		5	435822	20,78161
	2	150	29,29688		6	460310	21,94929
4	1	1462	35,69336	8	1	5991862	35,71428
	2	1166	28,46680		2	4366574	26,02681
	3	1046	25,53711		3	3563358	21,23927
5	1	11702	35,71167		4	3232014	19,26430
	2	8958	27,33765		5	3198622	19,06527
	3	7918	24,16382		6	3391086	20,21245
	4	7702	23,50464		7	3638806	21,68897
6	1	93622	35,71396	9	1	47934902	35,71429
	2	69806	26,62888		2	34786302	25,91781
	3	59422	22,66769		3	28144334	20,96916
	4	57454	21,91696		4	25110494	18,70878
	5	58902	22,46933		5	24182542	18,01740
7	1	748982	35,71424		6	24778398	18,46135
	2	550206	26,23587		7	26746478	19,92768
	3	456078	21,74749		8	28935702	21,55878

4. Свойства чисел невозможных разностей

В этом разделе найдём оценки для чисел $\mathcal{N}_n^r$ и проведём их сравнительный анализ при фиксированном n . П. 4.1 содержит нижнюю оценку, асимптотически точную при $n, n-r \rightarrow \infty$. В п. 4.2 числа $\mathcal{N}_n^1, \dots, \mathcal{N}_n^{n-1}$ упорядочены по величине. Как следствие, получен ряд точных оценок для $\mathcal{N}_n^r$ при некоторых ограничениях на r . Результаты также показывают примерное отношение числа невозможных разностей к числу всех разностей. Например, в табл. 8 приведены такие отношения при самом востребованном с точки зрения практических приложений значении $n = 32$.

4.1. Нижняя оценка $\mathcal{N}_n^r$. Для доказательства дальнейших утверждений преобразуем формулу из теоремы 1.

Таблица 8

Отношение $\mathcal{N}_{32}^r/8^{32}$ при $1 \leq r < 32$

r	$\mathcal{N}_{32}^r/8^{32}, \%$	r	$\mathcal{N}_{32}^r/8^{32}, \%$	r	$\mathcal{N}_{32}^r/8^{32}, \%$
1	35,71429	12	15,52034	23	15,55504
2	25,80357	13	15,51531	24	15,59874
3	20,68080	14	15,51282	25	15,68429
4	18,09849	15	15,51165	26	15,85039
5	16,80472	16	15,51120	27	16,16909
6	16,15751	17	15,51124	28	16,76897
7	15,83386	18	15,51180	29	17,85714
8	15,67204	19	15,51316	30	19,64286
9	15,59112	20	15,51597	31	21,42857
10	15,55067	21	15,52161		
11	15,53044	22	15,53284		

Следствие 2. Пусть $n \geq 2$ и $1 \leq r < n$. Тогда

$$\begin{aligned} \mathcal{N}_n^r = & \frac{38}{245}8^n - \frac{5}{18}4^n + \frac{5}{21}8^r \cdot 4^{n-r} + \frac{29}{70}4^r \cdot 8^{n-r} - \frac{4}{35}8^r \cdot 3^{n-r} + \\ & + \frac{2}{15}4^r \cdot 3^{n-r} - \frac{20}{147}8^r + \frac{4}{63}4^r - \frac{6}{147}8^{n-r} - \frac{50}{63}4^{n-r} + \frac{8}{21}3^{n-r} - \frac{10}{441}. \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Раскрывая скобки в формуле для $\mathcal{N}_n^r$ из теоремы 1, получаем

$$\begin{aligned} & \left(\frac{6}{35}8^{n-r} + \frac{10}{3}4^{n-r} - \frac{8}{5}3^{n-r} + \frac{2}{21} \right) \left(\frac{1}{14}8^r - \frac{1}{12}4^r - \frac{5}{21} \right) = \\ & = \frac{3}{35 \cdot 7}8^n + \frac{5}{3 \cdot 7}8^r \cdot 4^{n-r} - \frac{4}{5 \cdot 7}8^r \cdot 3^{n-r} + \frac{1}{21 \cdot 7}8^r - \\ & - \frac{1}{35 \cdot 2}4^r \cdot 8^{n-r} - \frac{5}{3 \cdot 6}4^n + \frac{2}{5 \cdot 3}4^r \cdot 3^{n-r} - \frac{1}{21 \cdot 6}4^r - \\ & - \frac{6}{7 \cdot 21}8^{n-r} - \frac{10 \cdot 5}{3 \cdot 21}4^{n-r} + \frac{8}{21}3^{n-r} - \frac{10}{21 \cdot 21}, \end{aligned}$$

поэтому

$$\begin{aligned} \mathcal{N}_n^r = & \frac{35+3}{35 \cdot 7}8^n - \frac{5}{6 \cdot 3}4^n + \frac{5}{3 \cdot 7}8^r \cdot 4^{n-r} + \frac{30-1}{2 \cdot 35}4^r \cdot 8^{n-r} - \\ & - \frac{4}{5 \cdot 7}8^r \cdot 3^{n-r} + \frac{2}{15}4^r \cdot 3^{n-r} - \frac{21-1}{21 \cdot 7}8^r + \frac{9-1}{2 \cdot 7 \cdot 9}4^r - \\ & - \frac{6}{7 \cdot 21}8^{n-r} - \frac{10 \cdot 5}{3 \cdot 21}4^{n-r} + \frac{8}{21}3^{n-r} - \frac{10}{21 \cdot 21}. \end{aligned}$$

Следствие 2 доказано.

Используя полученную формулу, несложно доказать следующую нижнюю оценку для чисел $\mathcal{N}_n^r$.

Следствие 3. Пусть $n \geq 2$ и $1 \leq r < n$. Тогда

$$\mathcal{N}_n^r > \frac{38}{245} 8^n.$$

ДОКАЗАТЕЛЬСТВО. Ясно, что $\mathcal{N}_n^{n-1}, \mathcal{N}_n^{n-2} > \frac{38}{245} 8^n$ (см. следствие 1). Далее положим $n - r \geq 3$ и воспользуемся следствием 2:

$$\begin{aligned} \mathcal{N}_n^r &= \frac{38}{245} 8^n - \frac{5}{18} 4^n + \frac{5}{21} 8^r \cdot 4^{n-r} + \frac{29}{70} 4^r \cdot 8^{n-r} - \frac{4}{35} 8^r \cdot 3^{n-r} + \\ &+ \frac{2}{15} 4^r \cdot 3^{n-r} - \frac{20}{147} 8^r + \frac{4}{63} 4^r - \frac{6}{147} 8^{n-r} - \frac{50}{63} 4^{n-r} + \frac{8}{21} 3^{n-r} - \frac{10}{441}. \end{aligned}$$

Докажем оценку в три шага. Во-первых, отметим, что

$$\frac{5}{21} 8^r \cdot 4^{n-r} > \frac{4}{35} 8^r \cdot 3^{n-r} + \frac{20}{147} 8^r,$$

поскольку

$$\frac{4}{35} 8^r \cdot 3^{n-r} + \frac{20}{147} \cdot \frac{4^{n-r}}{4^{n-r}} 8^r \leq \left(\frac{4}{35} + \frac{20}{4^3 \cdot 147} \right) 8^r \cdot 4^{n-r} < \frac{5}{21} 8^r \cdot 4^{n-r}.$$

Во-вторых,

$$\frac{29}{70} 4^r \cdot 8^{n-r} > \frac{5}{18} 4^n + \frac{6}{147} 8^{n-r} + \frac{50}{63} 4^{n-r},$$

так как

$$\begin{aligned} \frac{5}{18} \cdot \frac{8^{n-r}}{2^{n-r}} 4^r + \frac{6}{147} \cdot \frac{4^r}{4^r} 8^{n-r} + \frac{50}{63} \cdot \frac{4^r \cdot 8^{n-r}}{4^r \cdot 2^{n-r}} &\leq \\ &\leq \left(\frac{5}{8 \cdot 18} + \frac{6}{4 \cdot 147} + \frac{50}{2^5 \cdot 63} \right) 4^r \cdot 8^{n-r} = \\ &= \left(\frac{5 \cdot 2 \cdot 49 + 8 \cdot 3 \cdot 6 + 7 \cdot 50}{32 \cdot 9 \cdot 49} \right) 4^r \cdot 8^{n-r} = \\ &= \frac{984}{14112} 4^r \cdot 8^{n-r} < \frac{29}{70} 4^r \cdot 8^{n-r}. \end{aligned}$$

В-третьих,

$$\frac{8}{21} 3^{n-r} > \frac{10}{441}.$$

Суммируя сказанное, получаем требуемое неравенство для числа $\mathcal{N}_n^r$. Утверждение 2 доказано.

Доказанная оценка лучше, чем $\mathcal{N}_n^r \geq \frac{1}{7} 8^n - \frac{1}{7} 8^r$, которая была предложена в [19]. Более того, она асимптотически точна при удалении от крайних значений r . Непосредственно из следствия 2 вытекает

Следствие 4. Справедливо равенство $\lim_{\substack{r \rightarrow \infty \\ n-r \rightarrow \infty}} \frac{\mathcal{N}_n^r}{8^n} = \frac{38}{245}$, т. е.

$$\lim_{\substack{r \rightarrow \infty \\ n-r \rightarrow \infty}} \frac{\mathcal{N}_n^r}{\frac{38}{245} 8^n} = 1.$$

Отметим также, что $\frac{38}{245} \approx 15,51020\%$. В то же время доля невозможных разностей по модулю 2^n для преобразования $(x, y) \mapsto x \oplus y$ примерно равна $\frac{4}{7}$ (см. [15]), что составляет около $57,14286\%$. Следовательно, добавление циклического сдвига с далёким от крайних значений в несколько раз снижает число невозможных разностей.

4.2. Соотношения чисел $\mathcal{N}_n^r$ и точные оценки. Докажем неравенства для полученных в следствии 1 формул.

Лемма 6. При $n \geq 7$ имеем $\mathcal{N}_n^1 > \mathcal{N}_n^2 > \mathcal{N}_n^3$ и $\mathcal{N}_n^{n-3} < \mathcal{N}_n^{n-2} < \mathcal{N}_n^{n-1}$.

ДОКАЗАТЕЛЬСТВО. Пользуясь формулами из следствия 1, получаем

$$\frac{5}{14} 8^n > \frac{289}{1120} 8^n > \frac{1853}{8960} 8^n, \quad \frac{5}{28} 8^n < \frac{11}{56} 8^n < \frac{3}{14} 8^n.$$

Несложно проверить, что при $n \geq 7$ другие слагаемые не нарушают этих неравенств. Лемма 6 доказана.

Теперь можем упорядочить числа $\mathcal{N}_n^1, \dots, \mathcal{N}_n^{n-1}$.

Теорема 2. Если $n \geq 2$, $r = n/2 + 1$ при $n \in \{4, 6, 8, 10, 12\}$ и $r = \lceil n/2 \rceil$ иначе, то

$$\mathcal{N}_n^1 > \mathcal{N}_n^2 > \dots > \mathcal{N}_n^r < \mathcal{N}_n^{r+1} < \dots < \mathcal{N}_n^{n-1}.$$

ДОКАЗАТЕЛЬСТВО. Из табл. 7 вытекает утверждение теоремы при $n \leq 6$. Далее пусть $n \geq 7$. Положим $\Delta_{n,r} = \mathcal{N}_n^r - \mathcal{N}_n^{r+1}$ для любого r , $1 \leq r \leq n-2$. Для доказательства необходимо показать, что

- 1) $\Delta_{n,r} > 0$ при $r < n/2$,
- 2) $\Delta_{n,r} < 0$ при $r > n/2$,
- 3) $\Delta_{n,n/2} < 0$ для чётного $n \geq 14$,
- 4) $\Delta_{n,n/2} > 0$ для $n \in \{8, 10, 12\}$.

Из леммы 6 следует, что $\Delta_{n,1}$ и $\Delta_{n,2}$ положительны, а $\Delta_{n,n-3}$ и $\Delta_{n,n-2}$ отрицательны. Таким образом, можем полагать, что $3 \leq r \leq n-4$.

Все значения $\mathcal{N}_n^r$ и $\mathcal{N}_n^{r+1}$ могут быть вычислены с помощью следствия 2:

$$\begin{aligned} \mathcal{N}_n^r = & \frac{38}{245} 8^n - \frac{5}{18} 4^n + \frac{5}{21} 8^r \cdot 4^{n-r} + \frac{29}{70} 4^r \cdot 8^{n-r} - \frac{4}{35} 8^r \cdot 3^{n-r} + \\ & + \frac{2}{15} 4^r \cdot 3^{n-r} - \frac{20}{147} 8^r + \frac{4}{63} 4^r - \frac{6}{147} 8^{n-r} - \frac{50}{63} 4^{n-r} + \frac{8}{21} 3^{n-r} - \frac{10}{441}. \end{aligned}$$

Отметим, что для любых чисел a и b справедливо равенство

$$a^r b^{n-r} - a^{r+1} b^{n-(r+1)} = a^r b^{n-r-1} (b - a),$$

поэтому

$$\begin{aligned} \Delta_{n,r} = & -\frac{4 \cdot 5}{21} 8^r \cdot 4^{n-r-1} + \frac{4 \cdot 29}{70} 4^r \cdot 8^{n-r-1} + \\ & + \frac{5 \cdot 4}{35} 8^r \cdot 3^{n-r-1} - \frac{2}{15} 4^r \cdot 3^{n-r-1} + \frac{7 \cdot 20}{147} 8^r - \frac{3 \cdot 4}{63} 4^r - \\ & - \frac{7 \cdot 6}{147} 8^{n-r-1} - \frac{3 \cdot 50}{63} 4^{n-r-1} + \frac{2 \cdot 8}{21} 3^{n-r-1}. \end{aligned}$$

Следовательно,

$$\begin{aligned} \frac{21}{8^n} \Delta_{n,r} = & -5 \cdot 8^{-n} \cdot 8^r \cdot 4^{n-r} + \frac{87}{20} 8^{-n} \cdot 4^r \cdot 8^{n-r} + 4 \cdot 8^{-n} \cdot 8^r \cdot 3^{n-r} + \\ & + T_{n,r} - S_{n,r} = \frac{87}{20} 2^{-r} - S_{n,r} - \left(5 \cdot 2^{r-n} - T_{n,r} - 4 \left(\frac{8}{3} \right)^{r-n} \right), \quad (2) \end{aligned}$$

где

$$\begin{aligned} T_{n,r} = & 20 \cdot 8^{-n} \cdot 8^r - 4 \cdot 8^{-n} \cdot 4^r = \frac{20}{4^{n-r}} 2^{r-n} - \frac{4}{2^r \cdot 4^{n-r}} 2^{r-n}, \\ S_{n,r} = & \frac{14}{15} 8^{-n} \cdot 4^r \cdot 3^{n-r} + 6 \cdot 8^{-n} \cdot 8^{n-r-1} + \\ & + 50 \cdot 8^{-n} \cdot 4^{n-r-1} - 16 \cdot 8^{-n} \cdot 3^{n-r-1} = \\ = & \frac{14}{15} \left(\frac{3}{8} \right)^{n-r} 2^{-r} + \frac{3}{4^{r+1}} 2^{-r} + \frac{25}{4^r \cdot 2^{n-r+1}} 2^{-r} - \frac{16}{3 \cdot 4^r} \left(\frac{3}{8} \right)^{n-r} 2^{-r}. \end{aligned}$$

В то же время $r \geq 3$ и $n - r \geq 4$, поэтому

$$\begin{aligned} 0 < T_{n,r} & < \frac{20}{4^4} 2^{r-n} = \frac{5}{64} 2^{r-n}, \\ 0 < S_{n,r} & < \frac{14}{15} \left(\frac{3}{8} \right)^4 2^{-r} + \frac{3}{4^4} 2^{-r} + \frac{25}{4^3 \cdot 2^5} 2^{-r} = \\ & = \frac{7 \cdot 27 + 3 \cdot 5 \cdot 2^3 + 25 \cdot 5}{5 \cdot 2^{11}} 2^{-r} = \frac{189 + 120 + 125}{5 \cdot 2^{11}} 2^{-r} = \frac{217}{5120} 2^{-r}. \end{aligned}$$

Для удобства можем считать, что

$$0 < T_{n,r} < \frac{1}{10} 2^{r-n} < \frac{13}{40} 2^{r-n}, \quad (3)$$

$$0 < S_{n,r} < \frac{1}{20} 2^{-r}. \quad (4)$$

Будем использовать эти неравенства совместно с (2) для $\Delta'_{n,r} = \frac{21}{8^n} \Delta_{n,r}$. Отметим, что $\Delta_{n,r}$ и $\Delta'_{n,r}$ имеют одинаковый знак.

1. Докажем, что $\Delta'_{n,r} > 0$ при $r < n/2$. В силу (3) и (4) достаточно проверить, что $5 \cdot 2^{r-n} \leq \frac{87-1}{20} 2^{-r}$. Действительно, имеем $n-r > n/2$, или $n-r \geq r+1$, что влечёт

$$5 \cdot 2^{r-n} \leq 5 \cdot 2^{-r-1} = \frac{5}{2} 2^{-r} = \frac{50}{20} 2^{-r} < \frac{86}{20} 2^{-r}.$$

2. Докажем, что $\Delta'_{n,r} < 0$ при $r > n/2$. В силу (4) достаточно показать, что

$$5 \cdot 2^{r-n} - T_{n,r} - 4 \left(\frac{8}{3} \right)^{r-n} \geq \frac{87}{20} 2^{-r}. \quad (5)$$

Аналогично п. 1 справедливо $r \geq n-r+1$. Это означает, что

$$\frac{87}{20} 2^{-r} \leq \frac{87}{20} 2^{r-n-1} = \frac{87}{40} 2^{r-n}.$$

Таким образом, (3) обеспечивает оценку

$$5 \cdot 2^{r-n} - T_{n,r} - \frac{87}{20} 2^{-r} > \frac{200 - 13 - 87}{40} 2^{r-n} = \frac{5}{2} 2^{r-n}.$$

Осталось заметить, что $\frac{5}{2} 2^{r-n} \geq 4 \left(\frac{8}{3} \right)^{r-n}$. Это эквивалентно неравенству $\frac{5}{8} 8^{n-r} \geq 3^{n-r} \cdot 2^{n-r}$, которое имеет место, поскольку $5 \cdot 8^{n-r-1} \geq 6^{n-r}$ при $n-r \geq 2$.

3. Докажем, что $\Delta'_{n,n/2} < 0$ для чётных $n \geq 14$. Аналогично п. 2, покажем, что справедливо неравенство (5). Поскольку $r = n-r = n/2$, оно эквивалентно

$$\left(5 - \frac{87}{20} \right) \cdot 2^{-n/2} - T_{n,r} \geq 4 \left(\frac{8}{3} \right)^{-n/2}.$$

В то же время в силу (3) достаточно доказать, что

$$\frac{11}{20} 2^{-n/2} \geq 4 \left(\frac{8}{3} \right)^{-n/2}.$$

Это эквивалентно неравенству $11 \cdot 4^{n/2-2} \geq 5 \cdot 3^{n/2}$, которое имеет место при $n/2 \geq 7$. Действительно, $11 \cdot 4^5 = 11264$ и $5 \cdot 3^7 = 10935$.

Таблица 9

Значения $\mathcal{N}_n^{n/2}$ и $\mathcal{N}_n^{n/2+1}$ при чётных n , $4 \leq n \leq 12$

n	$\mathcal{N}_n^{n/2}$	$\mathcal{N}_n^{n/2}/8^n, \%$	$\mathcal{N}_n^{n/2+1}$	$\mathcal{N}_n^{n/2+1}/8^n, \%$
4	1166	28,46680%	1046	25,53711%
6	59422	22,66769%	57454	21,91696%
8	3232014	19,26430%	3198622	19,06527%
10	187255262	17,43951%	186758926	17,39328%
12	11332868174	16,49149%	11329597918	16,48673%

4. Табл. 9 показывает, что $\mathcal{N}_n^{n/2} > \mathcal{N}_n^{n/2+1}$ при $n \in \{8, 10, 12\}$. Теорема 2 доказана.

Прямым следствием теоремы 2 является точная нижняя оценка для числа $\mathcal{N}_n^r$.

Следствие 5. При $1 \leq r < n$ справедливо неравенство

$$\mathcal{N}_n^r \geq \begin{cases} \mathcal{N}_n^{n/2+1}, & \text{если } n \in \{4, 6, 8, 10, 12\}, \\ \mathcal{N}_n^{\lceil n/2 \rceil} & \text{иначе.} \end{cases}$$

Рассмотрим верхние оценки чисел $\mathcal{N}_n^r$. Согласно (1)

$$\mathcal{N}_n^r < \mathcal{N}_n^1, \quad 2 \leq r < n.$$

Однако $\mathcal{N}_n^1$ заметно больше всех других $\mathcal{N}_n^2, \dots, \mathcal{N}_n^{n-1}$. При этом теорема 2 позволяет получить другие оценки при дополнительных ограничениях на r .

Следствие 6. При $3 \leq r < n$

$$\mathcal{N}_n^r \leq \mathcal{N}_n^{n-1} = \frac{3}{14}8^n + \frac{2}{3}4^n - \frac{50}{21} < \mathcal{N}_n^2 < \mathcal{N}_n^1.$$

ДОКАЗАТЕЛЬСТВО. Из следствия 1 вытекает, что $\mathcal{N}_n^3 < \mathcal{N}_n^{n-1} < \mathcal{N}_n^2$. Действительно, $\frac{1853}{8960} < \frac{3}{14} < \frac{289}{1120}$, остальные слагаемые не нарушат это неравенство (при малых значениях n можно воспользоваться табл. 7). Для $4 \leq r \leq n-2$ неравенства следуют из теоремы 2. Следствие 6 доказано.

Рассмотренные оценки можно назвать точными, так как они представляют собой одно из чисел $\mathcal{N}_n^1, \dots, \mathcal{N}_n^{n-1}$.

5. Невозможные разности с точностью до симметрий

Наиболее многочисленны в табл. 2 шаблоны столбцов 1–4. При этом согласно доказательству теоремы 1 и следствию 4 доля слов, удовлетворяющих шаблону столбцов 1–4 и не удовлетворяющих шаблону столбцов 5–7, составляет $\frac{6}{35 \cdot 14} \cdot \frac{245}{38} = \frac{3}{38}$ при $r, n-r \rightarrow \infty$. Таким образом, эти столбцы нельзя игнорировать. Однако можно сократить число шаблонов за счёт преобразований аргументов, сохраняющих значение adr^{XR} .

Через $\mathcal{S}$ обозначим множество всех композиций преобразований

- $(\alpha, \beta, \gamma) \mapsto (\beta, \alpha, \gamma)$,
- $(\alpha, \beta, \gamma) \mapsto (-\alpha, \beta, \gamma)$, $(\alpha, \beta, \gamma) \mapsto (\alpha, -\beta, \gamma)$, $(\alpha, \beta, \gamma) \mapsto (\alpha, \beta, -\gamma)$,

где $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$.

Отметим, что $\mathcal{S}$ образует группу относительно композиции преобразований. Известно, что перечисленные преобразования сохраняют значение adr^{XR} (см. [19, теорема 5]). Для множества $V \subseteq (\mathbb{Z}_2^n)^3$ положим $\mathcal{S}(V) = \{s(v) \mid s \in \mathcal{S}, v \in V\}$.

Таблица 10

Невозможные разности с точностью
до преобразований из $\mathcal{S}$

№	ω	ω'	№	ω	ω'
1	$[\alpha_0*]$	$[.*d00*]$	6	$[0\hat{0}*7\gamma_1*1_7]$	$[.*d00*]$
2	$[0\hat{0}*]$	$[.*d00*]$	7	$[12_4*0_6\gamma_0*]$	$[\gamma_0*2_400*]$
3	$[0\hat{0}*1.*]$	$[.*d00*]$	8	$[.*d]$	$[\gamma_0*]$
4	$[0\hat{0}*2\alpha_0*]$	$[.*d00*]$	9	$[0*]$	$[.*d]$
5	$[0\hat{0}*7]$	$[.*d00*]$	10	$[.*d00*]$	$[.*]$

Теорема 3. Пусть множество $I \subseteq (\mathbb{Z}_2^n)^3$ содержит все тройки $((\alpha', \alpha), (\beta', \beta), (\gamma, \gamma'))$, где $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, такие, что $\omega = \omega(\alpha, \beta, \gamma)$ и $\omega' = \omega(\alpha', \beta', \gamma')$ удовлетворяют некоторой паре из табл. 10. Для векторов $x, y, z \in \mathbb{Z}_2^n$ имеет место $\text{adr}^{\text{XR}}(x, y \xrightarrow{r} z) = 0$ тогда и только тогда, когда $(x, y, z) \in \mathcal{S}(I)$.

ДОКАЗАТЕЛЬСТВО. Для шаблона ξ через $\xi^{-\alpha}$ обозначим множество шаблонов таких, что $\omega(\alpha, \beta, \gamma)$ удовлетворяет ξ тогда и только тогда, когда $\omega(-\alpha, \beta, \gamma)$ удовлетворяет некоторому шаблону из $\xi^{-\alpha}$. Аналогично определим $\xi^{-\gamma}$. Напомним, что

$$-(\alpha', \alpha) = \begin{cases} (\bar{\alpha}', -\alpha), & \text{если } \alpha \neq 0, \\ (-\alpha', 0) & \text{иначе,} \end{cases}$$

$$-0 = 0, \quad -(\alpha_1, \dots, \alpha_k, 1, 0, \dots, 0) = (\bar{\alpha}_1, \dots, \bar{\alpha}_k, 1, 0, \dots, 0),$$

где $k \in \{0, \dots, n-r-1\}$ можно подобрать для любого $\alpha \neq 0$. Другими словами, чтобы найти $\xi^{-\alpha}$ для некоторого ξ , сначала пересекаем ξ с шаблоном $[.*\alpha_1\alpha_0*]$, при необходимости разделив ξ на несколько подшаблонов, а затем к части ξ , соответствующей $.*$ в $[.*\alpha_1\alpha_0*]$, применяем преобразование $\bar{\alpha}$. Если ξ имеет непустое пересечение с $[\alpha_0*]$ (случай $\alpha = 0$), то оно тоже будет частью $\xi^{-\alpha}$. Например,

$$[.*e11*]^{-\alpha} = [.*e11*], \quad [.*d00*]^{-\alpha} = [.*d00*], \quad (6)$$

так как при $\alpha \neq 0$ будут инвертированы биты внутри $.*$, а при $\alpha = 0$ изменений не будет.

Начнём с преобразования шаблонов столбцов 2–4 в шаблоны из столбцов 1, 5 и 6, используя преобразование $-(\alpha', \alpha)$, а также меняя местами (α', α) и (β', β) . Обратим внимание, что ω -части из столбцов 1–6 замкнуты относительно $\bar{\gamma}$ -эквивалентности, при этом применяемые преобразования не зависят от γ . Таким образом, на данном этапе достаточно рассматривать только шаблоны из табл. 4 (см. лемму 3).

Столбец 4. Таким шаблонам не удовлетворяют слова с $\alpha = 0$, так как наиболее значимый символ в ω -части всегда $6 = (1, 1, 0)$. Таким образом, $-(\alpha', \alpha) = (\bar{\alpha}', -\alpha)$ и $\omega(\bar{\alpha}', \beta', \gamma')$ будет соответствовать метке столбца 2. Пересечём шаблоны с $[.*\alpha_1\alpha_0*]$. Далее будем часто использовать табл. 3 (см. утверждение 2).

4.1	$[\widehat{66}*7.*]$	4.2	$[\widehat{66}*1\gamma_1*1_7]$
4.1a	$[\widehat{66}*7.*\alpha_1\alpha_0*]$	4.2a	$[\widehat{66}*1\gamma_1*7]$
4.1b	$[\widehat{66}*7\alpha_0*]$	4.2b	$[\widehat{66}*1\gamma_1*5_71_3*1]$
		4.2c	$[\widehat{66}*11_3*1]$

Шаблон $4.1a^{-\alpha} = [2\widehat{2}*3.*\alpha_1\alpha_0*]$ удовлетворяет 2.3. Шаблон $4.1b^{-\alpha} = [2\widehat{2}*7\alpha_0*]$ удовлетворяет $[2\widehat{2}*\alpha_0*]$ (7 удовлетворяет $\widehat{2}$ согласно табл. 3). Значит, по лемме 2 он удовлетворяет одному из шаблонов столбца 2. Аналогично $4.2a^{-\alpha}$ удовлетворяет $[2\widehat{2}*5\gamma_1*7]$, а значит, и 2.5; $4.2b^{-\alpha} = [2\widehat{2}*5\gamma_1*5_71_3*1]$ также удовлетворяет 2.5. Поскольку 4.2c удовлетворяет $[.*e11*]$, в силу (6) все слова, удовлетворяющие $4.2c^{-\alpha}$, также будут удовлетворять $[.*e11*]$, т. е. 6.2 (с учётом и ω' -части).

Столбец 3. Здесь $\alpha = 0$ могут иметь только слова, удовлетворяющие $[0\beta_1*]$ (3.1), ω -части которых удовлетворяют $[00_1*]$. Такие слова удовлетворяют 5.1 (ω' -часть столбца 4 удовлетворяет $[.*d]$), 6.1 или 6.2.

Пусть $\alpha \neq 0$. Таким образом, $\omega(\bar{\alpha}', \beta', \gamma')$ будет удовлетворять метке столбца 1. Пересечём шаблоны с $[.*\alpha_1\alpha_0*]$:

3.1	$[0\beta_0*]$	3.2	$[\widehat{44}*]$	3.4	$[\widehat{44}*0\beta_0*]$
3.1a	$[0\beta_0*4_50_1*]$	3.2a	$[\widehat{44}*4_71*]$	3.4a	$[\widehat{44}*0\beta_0*4_50_1*]$
3.1b	$\emptyset$	3.2b	$[\widehat{41}*]$	3.4b	$[\widehat{44}*00_1*]$

Шаблон $3.1a^{-\alpha} = [4\beta_0*4_50_1*]$ удовлетворяет шаблону 1.2. Шаблон $3.2a^{-\alpha} = [0\widehat{0}*4_71*]$ удовлетворяет либо шаблону $[0\widehat{0}*4\beta_0*]$ (1.6), либо одному из двух: $[0\widehat{0}*7]$ или $[0\widehat{0}*7\gamma_1*1_7]$ (1.7 или 1.8). Шаблон $3.4a^{-\alpha} = [0\widehat{0}*4\beta_0*4_50_1*]$ удовлетворяет $[0\widehat{0}*4\beta_0*]$ (1.6). Шаблон 3.4b удовлетворяет $[.*d00*]$ или $[.*e11*]$, а значит, $3.4b^{-\alpha}$ удовлетворяет 6.1 или 6.2 в силу (6). Шаблоны 3.3 и 3.5 рассматриваются аналогично 4.1 и 4.2.

Столбец 2 можно не рассматривать, так как и его метка, и его элементы $\alpha\beta$ -эквивалентны шаблонам столбца 3.

Заполним табл. 10. Здесь возвращаемся к табл. 2, в которой достаточно рассмотреть столбцы 1, 5, 6 и 7. Добавим шаблоны 7.1 (строка 8), 6.1 (строка 9), 5.1 (строка 10), а также 1.1, 1.3–5, 1.7, 1.8 (строки 1–6). При этом не добавляем 1.2 и 1.6, так как они $\alpha\beta$ -эквивалентны 1.1 и 1.5 (с учётом ω' -части). Таким образом, осталось разобраться с 1.9–14, 5.2 и 6.2. Используем подстановку $-(\gamma, \gamma')$ вместо (γ, γ') .

Пусть $\gamma' \neq 0$, тогда $-(\gamma, \gamma') = (\bar{\gamma}, -\gamma')$. Отметим также, что метки столбцов 1, 5 и 6 симметричны относительно подстановки $-\gamma'$, т. е. слово $\omega(\alpha', \beta', \gamma')$ удовлетворяет одной из этих меток тогда и только тогда, когда слово $\omega(\alpha', \beta', -\gamma')$ удовлетворяет той же самой метке (используем (6) с $-\gamma$ вместо $-\alpha$). Поскольку ω -части 1.9–14, 5.2 и 6.2 $\bar{\gamma}$ -эквивалентны 1.3–8, 5.1 и 6.1, в табл. 10 ничего добавлять не требуется.

Пусть $\gamma' = 0$, тогда $-(\gamma, \gamma') = (-\gamma, 0)$. Значит, ω' -часть слов удовлетворяет $[\gamma_0^*]$. Таким образом, нужно понять, какие шаблоны из 1.9, 1.10, 1.11, 1.13 и 1.14 (1.12 $\alpha\beta$ -эквивалентен 1.11) удовлетворяют 1.3–1.8 после преобразования $\omega(\alpha, \beta, -\gamma)$. Рассматриваем только ω -части, удовлетворяющие $[*e]$, так как иначе получится подслучай 7.1. Например, в 7.1 укладываются исключённые из списка выше 5.2 и 6.2. Получаем, что ω' -части всех оставшихся слов должны удовлетворять и $[*d00^*]$, и $[\gamma_0^*]$, т. е. $[\gamma_0^*2_400^*]$. Как и в случае с α , пересечём все шаблоны с $[*\gamma_1\gamma_0^*]$. Также можем заменить шаблоны 1.13 и 1.14 «расширенным» $[1\hat{1}*6\gamma_0^*]$ (1.14). Действительно, добавили только $[1\hat{1}*6\gamma_0^*2_4]$, который удовлетворяет 7.1. Также можно не рассматривать 1.9: его ω -часть удовлетворяет $[*d]$, а значит, $1.9^{-\gamma}$ удовлетворяет 7.1.

1.10	$[1\hat{1}*0.*]$	1.11	$[1\hat{1}*3\alpha_0^*]$	$\widetilde{1.14}$	$[1\hat{1}*6\gamma_0^*]$
1.10a	$[1\hat{1}*0.*\gamma_1\gamma_0^*]$	1.11a	$[1\hat{1}*3\alpha_0^*1_30_2^*]$	$\widetilde{1.14a}$	$[1\hat{1}*12_4*6\gamma_0^*]$
1.10b	$[1\hat{1}*12_4*0\gamma_0^*]$	1.11b	$[1\hat{1}*30_2^*]$	$\widetilde{1.14b}$	$[12_4*6\gamma_0^*]$
1.10c	$[12_4*0\gamma_0^*]$				

Шаблон $1.10a^{-\gamma} = [0\hat{0}*1.*\gamma_1\gamma_0^*]$ удовлетворяет 1.4, как и $1.10b^{-\gamma} = [0\hat{0}*12_4*0\gamma_0^*]$. Далее, шаблон $1.11a^{-\gamma} = [0\hat{0}*2\alpha_0^*1_30_2^*]$ удовлетворяет 1.5, а $1.11b^{-\gamma} = [0\hat{0}*30_2^*] - [0\hat{0}*\alpha_0^*]$ ($3 \in \hat{0}$), слова которого по лемме 2 удовлетворяют 1.3, 1.4 или 1.5. Шаблон $\widetilde{1.14a}^{-\gamma} = [0\hat{0}*12_4*6\gamma_0^*]$ удовлетворяет 1.4. Шаблоны $1.10c^{-\gamma} = 1.10c$ и $\widetilde{1.14b}^{-\gamma} = \widetilde{1.14b}$ объединены в один шаблон и добавлены в табл. 10 (строка 7).

Табл. 10 заполнена. Указанные в ней шаблоны порождают невозможные разности, и с помощью преобразований из $\mathcal{S}$ к ним сводятся все шаблоны табл. 2. При этом преобразования из $\mathcal{S}$ сохраняют значение разностной характеристики adr^{XR} . Теорема 3 доказана.

Таким образом, применяя преобразования из $\mathcal{S}$ (перестановка первых двух аргументов adp^{XR} и замена любого аргумента противоположным по модулю 2^n), невозможные разности XR-преобразования можно описать 10-ю парами шаблонов. Отметим, что одновременная инверсия старших битов первых двух аргументов adp^{XR} также не изменяет её значения, при этом все шаблоны табл. 2 (учитывая метки) изначально замкнуты относительно этого преобразования.

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. **Beaulieu R., Shors D., Smith J., Treatman-Clark S., Weeks B., Wingers L.** The SIMON and SPECK families of lightweight block ciphers. San Diego: Univ. California, 2013. 45 p. (Cryptol. ePrint Archive; Paper ID 2013/404). URL: eprint.iacr.org/2013/404 (accessed: Dec. 9, 2024).
2. **Roh D., Koo B., Jung Y., Jeong I. W., Lee D.-G., Kwon D., Kim W.-H.** Revised version of block cipher CHAM // Information security and cryptology — ICISC 2019. Rev. Sel. Pap. 22th Int. Conf. (Seoul, South Korea, Dec. 4–6, 2019). Cham: Springer, 2020. P. 1–19. (Lect. Notes Comput. Sci.; V. 11975). DOI: 10.1007/978-3-030-40921-0_1.
3. **Bernstein D. J.** The Salsa20 family of stream ciphers. Chicago: Univ. Ill. Chic., 2007. 15 p. URL: cr.yp.to/snuffle/salsafamily-20071225.pdf (accessed: Dec. 9, 2024).
4. **Bernstein D. J.** ChaCha, a variant of Salsa20. Chicago: Univ. Ill. Chic., 2008. 6 p. URL: cr.yp.to/chacha/chacha-20080128.pdf (accessed: Dec. 9, 2024).
5. **Beierle C., Biryukov A., Cardoso dos Santos L. [et al.]** Lightweight AEAD and hashing using the sparkle permutation family // IACR Trans. Symmetric Cryptol. 2020. V. 2020, Suppl. 1. P. 208–261. DOI: 10.13154/tosc.v2020.is1.208-261.
6. **Mouha N., Mennink B., Van Herrewege A., Watanabe D., Preneel B., Verbauwhede I.** Chaskey: An efficient MAC algorithm for 32-bit microcontrollers // Selected areas in cryptography — SAC 2014. Rev. Sel. Pap. 21th Int. Conf. (Montreal, Canada, Aug. 14–15, 2014). Cham: Springer, 2014. P. 306–323. (Lect. Notes Comput. Sci.; V. 8781). DOI: 10.1007/978-3-319-13051-4_19.
7. **Biham E., Shamir A.** Differential cryptanalysis of DES-like cryptosystems // J. Cryptol. 1991. V. 4. P. 3–72.

8. **Biryukov A., Velichkov V.** Automatic search for differential trails in ARX ciphers // Topics in cryptology — CT-RSA 2014. Proc. Cryptographer's Track at the RSA Conf. (San Francisco, USA, Feb. 25–28, 2014). Cham: Springer, 2014. P. 227–250. (Lect. Notes Comput. Sci.; V. 8366). DOI: 10.1007/978-3-319-04852-9_12.
9. **Leurent G.** Analysis of differential attacks in ARX constructions // Advances in cryptology — ASIACRYPT 2012. Proc. 18th Int. Conf. Theory and Application of Cryptology and Information Security (Beijing, China, Dec. 2–6, 2012). Heidelberg: Springer, 2012. P. 226–243. (Lect. Notes Comput. Sci.; V. 7658).
10. **Leurent G.** Construction of differential characteristics in ARX designs application to Skein // Advances in cryptology — CRYPTO 2013. Proc. 33rd Annu. Cryptology Conf. (Santa Barbara, USA, Aug. 18–22, 2013). Heidelberg: Springer, 2013. P. 241–258. (Lect. Notes Comput. Sci.; V. 8042). DOI: 10.1007/978-3-642-40041-4_14.
11. **Малышев Ф. М.** Вероятностные характеристики разностных и линейных соотношений для неоднородной линейной среды // Мат. вопрос. криптографии. 2019. Т. 10, № 1. С. 41–72. DOI: 10.4213/mvk276.
12. **Малышев Ф. М.** Разностные характеристики основных операций ARX-шифров // Мат. вопрос. криптографии. 2020. Т. 11, № 4. С. 97–105. DOI: 10.4213/mvk342.
13. **Berson T. A.** Differential cryptanalysis mod 2^{32} with applications to MD5 // Advances in cryptology — EUROCRYPT'92. Proc. Workshop Theory and Application of Cryptographic Techniques (Balatonfüred, Hungary, May 24–28, 1992). Heidelberg: Springer, 1992. P. 71–80. (Lect. Notes Comput. Sci.; V. 658). DOI: 10.1007/3-540-47555-9_6.
14. **Daum M. A.** Cryptanalysis of hash functions of the MD4-family: PhD Thesis. Bochum: Ruhr-Univ. Bochum, 2005.
15. **Lipmaa H., Wallén J., Dumas P.** On the additive differential probability of exclusive-or // Fast software encryption. Rev. Pap. 11th Int. Workshop (Delhi, India, Feb. 5–7, 2004). Heidelberg: Springer, 2004. P. 317–331. (Lect. Notes Comput. Sci.; V. 3017). DOI: 10.1007/978-3-540-25937-4_20.
16. **Mouha N., Velichkov V., De Cannière C., Preneel B.** The differential analysis of S-functions // Selected areas in cryptography. Rev. Sel. Pap. 17th Int. Workshop (Waterloo, Canada, Aug. 12–13, 2010). Heidelberg: Springer, 2011. P. 36–56. (Lect. Notes Comput. Sci.; V. 6544). DOI: 10.1007/978-3-642-19574-7_3.
17. **Mouha N., Kolomeec N. A., Akhtiamov D.** [et al.]. Maximums of the additive differential probability of exclusive-or // IACR Trans. Symmetric Cryptol. 2021. V. 2021, No. 2. P. 292–313. DOI: 10.46586/tosc.v2021.i2.292-313.
18. **Velichkov V., Mouha N., De Cannière C., Preneel B.** The additive differential probability of ARX // Fast software encryption. Rev. Sel. Pap. 18th Int. Workshop (Lyngby, Denmark, Feb. 13–16, 2011). Heidelberg: Springer, 2011. P. 342–358. (Lect. Notes Comput. Sci.; V. 6733). DOI: 10.1007/978-3-642-21702-9_20.

19. Kolomeec N. A., Sutormin I. A., Bykov D. A., Panferov M. A., Bonich T. A. On additive differential probabilities of the composition of bitwise exclusive-or and a bit rotation. Ithaca, NY: Cornell Univ., 2023. 35 p. (Cornell Univ. Libr. e-Print Archive; arXiv:2303.04097). DOI: 10.48550/arXiv.2303.04097.
20. Мокроусов А. С., Коломеец Н. А. Разности по модулю 2^n для ARX-преобразований, вероятность которых больше $1/4$ // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 2. С. 108–135. DOI: 10.33048/daio.2024.31.769.
21. Sutormin I. A., Kolomeec N. A. On additive differential probabilities of a composition of bitwise XORs // Прикл. дискрет. математика. 2023. № 60. С. 59–75. DOI: 10.17223/20710410/60/5.
22. Knudsen L. DEAL — A 128-bit block cipher. Tech. Rep. No. 151. Bergen: Univ. Bergen, 1998. 9 p.
23. Biham E., Biryukov A., Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials // Advances in cryptology — EUROCRYPT'99. Proc. Int. Conf. Theory and Application of Cryptographic Techniques (Prague, Czech Republic, May 2–6, 1999). Heidelberg: Springer, 1999. P. 12–23. (Lect. Notes Comput. Sci.; V. 1592). DOI: 10.1007/3-540-48910-X_2.
24. Agievich S. V., Gorodilova A. A., Kolomeec N. A. [et al.]. Problems, solutions and experience of the First International Student's Olympiad in Cryptography // Прикл. дискрет. математика. 2015. № 3. С. 41–62. DOI: 10.17223/20710410/29/4.
25. Gorodilova A. A., Tokareva N. N., Agievich S. V. [et al.]. An overview of the Eight International Olympiad in Cryptography «Non-Stop University Crypto» // Сиб. электрон. мат. изв. 2022. Т. 19, № 1. С. A9–A37. DOI: 10.33048/semi.2022.19.023.

Коломеец Николай Александрович

Статья поступила

9 апреля 2024 г.

После доработки —

10 мая 2024 г.

Принята к публикации

22 июня 2024 г.

THE NUMBER OF IMPOSSIBLE ADDITIVE DIFFERENTIALS FOR THE COMPOSITION OF XOR AND BIT ROTATION

N. A. Kolomeec

Sobolev Institute of Mathematics,
4 Acad. Koptug Avenue, 630090 Novosibirsk, Russia
E-mail: nkolomeec@gmail.com

Abstract. Additive differentials of the function $(x \oplus y) \lll r$ whose probability is 0 are considered, where $x, y \in \mathbb{Z}_2^n$ and $1 \leq r < n$. They are called impossible differentials and are interesting in the context of differential cryptanalysis of ciphers whose schemes consist of additions modulo 2^n , bitwise XORs ($\oplus$), and bit rotations ($\lll r$). The number of all such differentials is calculated for all possible r and n . It is also shown that this number is greater than $\frac{38}{245}8^n$. Moreover, the estimate is asymptotically tight for $r, n-r \rightarrow \infty$. For any fixed n the number of all impossible differentials decreases as r goes from 1 to $\lceil n/2 \rceil$ (to $n/2 + 1$ in the case of $n \in \{4, 6, 8, 10, 12\}$) and then increases monotonically as r goes to $n - 1$. A simplified description of all impossible differentials is obtained up to known symmetries. Tab. 10, bibliogr. 25.

Keywords: ARX, differential probability, XOR, modular addition, bit rotation, impossible differential.

References

1. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, The SIMON and SPECK families of lightweight block ciphers (Univ. California, San Diego, 2013) (Cryptol. ePrint Archive; ID 2013/404), URL: eprint.iacr.org/2013/404 (accessed: Dec. 9, 2024).
2. D. Roh, B. Koo, Y. Jung, I. W. Jeong, D.-G. Lee, D. Kwon, and W.-H. Kim, Revised version of block cipher CHAM, in *Information Security and Cryptology — ICISC 2019* (Rev. Sel. Pap. 22th Int. Conf., Seoul, South Korea, Dec. 4–6, 2019) (Springer, Cham, 2020), pp. 1–19 (Lect. Notes Comput. Sci., Vol. 11975), DOI: 10.1007/978-3-030-40921-0_1.

3. **D. J. Bernstein**, The Salsa20 family of stream ciphers (Univ. Ill. Chic., Chicago, 2007), URL: cr.yp.to/snuffle/salsafamily-20071225.pdf (accessed: Dec. 9, 2024).
4. **D. J. Bernstein**, ChaCha, a variant of Salsa20 (Univ. Ill. Chic., Chicago, 2008), URL: cr.yp.to/chacha/chacha-20080128.pdf (accessed: Dec. 9, 2024).
5. **C. Beierle, A. Biryukov, L. Cardoso dos Santos**, [et al.]. Lightweight AEAD and hashing using the sparkle permutation family, *IACR Trans. Symmetric Cryptol.* **2020** (Suppl. 1), 208–261 (2020), DOI: 10.13154/tosc.v2020.iS1.208-261.
6. **N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel**, and **I. Verbauwhede**, Chaskey: An efficient MAC algorithm for 32-bit microcontrollers, in *Selected Areas in Cryptography — SAC 2014* (Rev. Sel. Pap. 21th Int. Conf., Montreal, Canada, Aug. 14–15, 2014) (Springer, Cham, 2014), pp. 306–323 (Lect. Notes Comput. Sci., Vol. 8781), DOI: 10.1007/978-3-319-13051-4_19.
7. **E. Biham** and **A. Shamir**, Differential cryptanalysis of DES-like cryptosystems, *J. Cryptol.* **4**, 3–72 (1991).
8. **A. Biryukov** and **V. Velichkov**, Automatic search for differential trails in ARX ciphers, in *Topics in Cryptology — CT-RSA 2014* (Proc. Cryptographer’s Track at the RSA Conf., San Francisco, USA, Feb. 25–28, 2014) (Springer, Cham, 2014), pp. 227–250 (Lect. Notes Comput. Sci., Vol. 8366), DOI: 10.1007/978-3-319-04852-9_12.
9. **G. Leurent**, Analysis of differential attacks in ARX constructions, in *Advances in Cryptology — ASIACRYPT 2012* (Proc. 18th Int. Conf. Theory and Application of Cryptology and Information Security, Beijing, China, Dec. 2–6, 2012) (Springer, Heidelberg, 2012), pp. 226–243 (Lect. Notes Comput. Sci., Vol. 7658).
10. **G. Leurent**, Construction of differential characteristics in ARX designs application to Skein, in *Advances in Cryptology — CRYPTO 2013* (Proc. 33rd Annu. Cryptology Conf., Santa Barbara, USA, Aug. 18–22, 2013) (Springer, Heidelberg, 2013), pp. 241–258 (Lect. Notes Comput. Sci., Vol. 8042), DOI: 10.1007/978-3-642-40041-4_14.
11. **F. M. Malyshev**, Probabilistic characteristics of differential and linear relations for nonhomogeneous linear medium, *Mat. Vopr. Kriptogr.* **10** (1), 41–72 (2019), DOI: 10.4213/mvk276 [Russian].
12. **F. M. Malyshev**, Differential characteristics of base operations in ARX-ciphers, *Mat. Vopr. Kriptogr.* **11** (4), 97–105 (2020), DOI: 10.4213/mvk342 [Russian].
13. **T. A. Berson**, Differential cryptanalysis mod 2^{32} with applications to MD5, in *Advances in Cryptology — EUROCRYPT’92* (Proc. Workshop Theory and Application of Cryptographic Techniques, Balatonfüred, Hungary, May 24–28, 1992) (Springer, Heidelberg, 1992), pp. 71–80 (Lect. Notes Comput. Sci., Vol. 658), DOI: 10.1007/3-540-47555-9_6.

14. **M. A. Daum**, Cryptanalysis of hash functions of the MD4-family, *PhD Thesis* (Ruhr-Univ. Bochum, Bochum, 2005).
15. **H. Lipmaa, J. Wallén and P. Dumas**, On the additive differential probability of exclusive-or, in *Fast Software Encryption* (Rev. Pap. 11th Int. Workshop, Delhi, India, Feb. 5–7, 2004) (Springer, Heidelberg, 2004), pp. 317–331 (Lect. Notes Comput. Sci., Vol. 3017), DOI: 10.1007/978-3-540-25937-4_20.
16. **N. Mouha, V. Velichkov, C. De Cannière and B. Preneel**, The differential analysis of S-functions, in *Selected Areas in Cryptography* (Rev. Sel. Pap. 17th Int. Workshop, Waterloo, Canada, Aug. 12–13, 2010) (Springer, Heidelberg, 2011), pp. 36–56 (Lect. Notes Comput. Sci., Vol. 6544), DOI: 10.1007/978-3-642-19574-7_3.
17. **N. Mouha, N. A. Kolomeec, D. Akhtiamov**, [et al.]. Maximums of the additive differential probability of exclusive-or, *IACR Trans. Symmetric Cryptol.* **2021** (2), 292–313 (2021), DOI: 10.46586/tosc.v2021.i2.292-313.
18. **V. Velichkov, N. Mouha, C. De Cannière and B. Preneel**, The additive differential probability of ARX, in *Fast Software Encryption* (Rev. Sel. Pap. 18th Int. Workshop, Lyngby, Denmark, Feb. 13–16, 2011) (Springer, Heidelberg, 2011), pp. 342–358 (Lect. Notes Comput. Sci., Vol. 6733), DOI: 10.1007/978-3-642-21702-9_20.
19. **N. A. Kolomeec, I. A. Sutormin, D. A. Bykov, M. A. Panferov**, and **T. A. Bonich**, On additive differential probabilities of the composition of bitwise exclusive-or and a bit rotation (Cornell Univ., Ithaca, NY, 2023) (Cornell Univ. Libr. e-Print Archive, arXiv:2303.04097), DOI: 10.48550/arXiv.2303.04097.
20. **A. S. Mokrousov and N. A. Kolomeec**, Additive differentials for ARX mappings with probability exceeding $1/4$, *Diskretn. Anal. Issled. Oper.* **31** (2), 108–135 (2024), DOI: 10.33048/daio.2024.31.769 [Russian] [*J. Appl. Ind. Math.* **18** (2), 294–311 (2024), DOI: 10.1134/S199047892402011X].
21. **I. A. Sutormin and N. A. Kolomeec**, On additive differential probabilities of a composition of bitwise XORs, *Prikl. Diskretn. Mat.*, No. 60, 59–75 (2023), DOI: 10.17223/20710410/60/5.
22. **L. Knudsen**, DEAL — A 128-bit block cipher, *Tech. Rep. No. 151* (Univ. Bergen, Bergen, 1998).
23. **E. Biham, A. Biryukov**, and **A. Shamir**, Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials, in *Advances in Cryptology — EUROCRYPT’99* (Proc. Int. Conf. Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999) (Springer, Heidelberg, 1999), pp. 12–23 (Lect. Notes Comput. Sci., Vol. 1592), DOI: 10.1007/3-540-48910-X_2.
24. **S. V. Agievich, A. A. Gorodilova, N. A. Kolomeec**, [et al.]. Problems, solutions, and experience of the First International Student’s Olympiad in Cryptography, *Prikl. Diskretn. Mat.*, No. 3, 41–62 (2015), DOI: 10.17223/20710410/29/4.

-
- 25. A. A. Gorodilova, N. N. Tokareva, S. V. Agievich**, [et al.]. An overview of the Eight International Olympiad in Cryptography “Non-Stop University Crypto”, *Sib. Elektron. Mat. Izv.* **19** (1), A9–A37 (2022), DOI: 10.33048/semi.2022.19.023.

Nikolay A. Kolomeec

Received April 9, 2024

Revised May 10, 2024

Accepted June 22, 2024

УТОЧНЁННЫЕ ОЦЕНКИ ДЛЯ АЛГОРИТМОВ УПАКОВКИ 2-ГИСТОГРАММ В ПОЛОСУ

С. А. Назаренко

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

E-mail: s.nazarenko@g.nsu.ru

Аннотация. Рассматривается NP-трудная задача, в которой требуется упаковать гистограммы из двух столбиков в полосу минимальной длины единичной высоты. Высота каждого столбика положительная и не превосходит 1. Эта задача, называемая в англоязычной литературе Two-Bar Charts Packing Problem, является обобщением задачи упаковки в контейнеры и задачи двумерной векторной упаковки. Доказываются уточнённые оценки точности и трудоёмкости для некоторых ранее разработанных приближённых полиномиальных алгоритмов для Two-Bar Charts Packing Problem и её частных случаев. Показана достижимость этих оценок. Кроме того, рассматривается задача упаковки неограниченного числа гистограмм k различных типов. При этом оценивается не длина упаковки, а её плотность, и предлагается полиномиальный алгоритм решения последней задачи в случае $k = \text{const}$. Ил. 12, библиогр. 26.

Ключевые слова: гистограмма, упаковка в полосу, NP-трудная задача, априорная оценка, достижимость.

Введение

Задача упаковки гистограмм в полосу (bar charts packing problem, ВСРР) возникла при оптимизации инвестиционного портфеля в нефтегазовой сфере [1]. Предположим, что территория месторождения разбита на кластеры, для каждого из которых известен проект его разработки. Проект характеризуется, в частности, ежегодными объёмами добычи нефти. Таким образом, если известен год начала проекта, то известны и объёмы добычи в первый и все последующие годы реализации проекта. График добычи нефти по каждому проекту может быть представлен

Описанная проблема сводится к поиску упаковки гистограмм в часть полубесконечной полосы минимальной длины. Ранее исследовался частный случай задачи, называемый 2-ВСРР, когда каждая гистограмма состоит из двух столбиков. Для удобства будем обозначать гистограмму с двумя столбиками через 2-Г. Задача 2-ВСРР впервые сформулирована в [2], а затем изучена в [3–7]. Пусть имеется полубесконечная горизонтальная полоса единичной высоты и множество, состоящее из n 2-Г. Разобьём полосу на одинаковые ячейки единичной высоты и ширины. Все столбики гистограмм имеют единичную ширину и положительную высоту, не превышающую 1. В допустимой упаковке столбики каждой гистограммы могут перемещаться по вертикали, но неразрывны по горизонтали, и их нельзя переставлять местами. Более того, в допустимой упаковке сумма высот столбиков, попавших в одну ячейку, не превосходит 1. Задача 2-ВСРР заключается в нахождении допустимой упаковки n 2-Г в минимальное число ячеек. На рис. 1 представлен пример допустимой упаковки трёх гистограмм в 4 ячейки, т. е. длина упаковки равна 4.

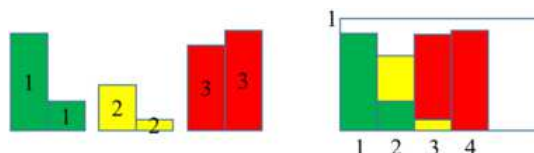


Рис. 1. Пример допустимой упаковки трёх 2-Г

В классической одномерной задаче упаковки в контейнеры ВРР задано множество предметов L , размер каждого предмета, а также множество одинаковых контейнеров единичной ёмкости. Требуется разместить все предметы в минимальное число контейнеров. ВРР является частным случаем задачи 2-BCPP, когда каждая 2-Г состоит из столбиков одинаковой высоты. Если $P \neq NP$, то задача ВРР $(\frac{3}{2} - \varepsilon)$ -неаппроксимируема для любого $\varepsilon > 0$ [8, 9]. Как показано в [10], $\frac{3}{2}$ -приближённое решение может быть найдено алгоритмом First Fit Decreasing (FFD). В алгоритме FFD

предметы сначала нумеруются в порядке невозрастания размеров, затем просматриваются в порядке нумерации, и текущий предмет кладётся в первый контейнер, куда он помещается. В 2007 г. доказано, что алгоритм FFD использует не более $\frac{11}{9} \text{OPT}(L) + \frac{6}{9}$ контейнеров, где $\text{OPT}(L)$ — минимальное число контейнеров для упаковки предметов множества L , и найден пример достижимости этой оценки [11]. Модифицированная версия алгоритма FFD (modified first fit decreasing, MFFD) группирует предметы по размеру и пакует отдельно предметы разных групп. Джонсон и Гэри, построившие алгоритм MFFD, показали, что он использует не более $\frac{71}{60} \text{OPT}(L) + \frac{31}{6}$ контейнеров [12]. В дальнейшем аддитивная константа уменьшена до 1 [13]. В [14] для задачи ВРР предложена асимптотическая полиномиальная аппроксимационная схема (APTAS) и показано, что для любого $\varepsilon > 0$ существует алгоритм линейной трудоёмкости, который использует не более $(1 + \varepsilon) \text{OPT}(L)$ контейнеров. Также для задачи ВРР построена асимптотическая вполне полиномиальная аппроксимационная схема [15] и аддитивный приближённый алгоритм, который строит упаковку длины не более $\text{OPT}(L) + O(\log \text{OPT}(L))$ [16].

Задача 2-DVPP представляет собой обобщение задачи ВРР и частный случай 2-BCPP. В этом случае каждая пара 2-Г может располагаться в контейнере одна над другой. При этом предметы и контейнеры характеризуются двумя параметрами. Требуется упаковать все предметы в минимальное число контейнеров, не нарушая ограничений по обоим параметрам. В 2003 г. представлен 2-приближённый алгоритм для 2-DVPP, трудоёмкость которого равна $O(n \log n)$ [17]. Подробный обзор приближённых алгоритмов для 2-DVPP можно найти в [18]. Лучший алгоритм строит $(\frac{3}{2} + \varepsilon)$ -приближённое решение для любого $\varepsilon > 0$ [19]. В 1997 г. доказано, что не существует схемы APTAS для задачи 2-DVPP, если $P \neq NP$ [20].

С другой стороны, 2-BCPP является частным случаем задачи построения расписания для выполнения проектов с ограниченным однородным возобновляемым ресурсом (resource-constrained project scheduling problem, RCPSP) [21]. В 2-BCPP каждый проект состоит из двух работ единичной длительности, которые выполняются одна за другой без прерывания. Такой проект можно представить в виде 2-Г, в которой высота столбика соответствует величине потребляемого возобновляемого ресурса. Требуется определить момент начала каждого проекта таким образом, чтобы выполнить все проекты за минимальное время, учитывая ограничение на ресурс. Задача RCPSP NP-трудна, и для неё не известны полиномиальные алгоритмы с гарантированными оценками точности. Как правило, для построения приближённого решения разрабатываются эвристические алгоритмы, и проводится их апостериорный анализ [22–25].

Задача 2-ВСРР достаточно новая и рассматривается в небольшом числе работ [2–7, 26]. В [2] предложен алгоритм линейной трудоёмкости, который строит упаковку длины не более $2 \text{OPT} + 1$, где OPT — минимальная длина упаковки 2-Г. В [3–5] исследовались частные случаи задачи 2-ВСРР, когда каждая 2-Г содержит хотя бы один столбик выше $\frac{1}{2}$ и — дополнительно — когда все 2-Г либо невозрастающие, либо неубывающие. В [3] предлагаются два полиномиальных $\frac{3}{2}$ -приближённых алгоритма для решения этих задач. Авторы статьи [5] доказали, что 2-ВСРР, когда каждая 2-Г содержит столбик высоты более $\frac{1}{2}$, остаётся NP-трудной в сильном смысле. В этой же работе предложен алгоритм трудоёмкости $O(n^{2.5})$, который строит упаковку длины не более $\frac{4}{3} \text{OPT} + \frac{4}{3}$. В [6] предложен алгоритм трудоёмкости $O(n \log n)$ для поиска линейно упорядоченной упаковки, в которой первые столбики всех 2-Г находятся в разных ячейках длины не более $\text{OPT} + 1$. В [7] рассматриваются классы задач, которые являются как обобщениями 2-ВСРР, так и её частными случаями. Используя известные алгоритмы, авторы находят верхнюю и нижнюю границы значения целевой функции для задачи 2-ВСРР, а также предлагают две модели целочисленного линейного программирования и проводят численный эксперимент на шести различных базах данных. В [26] исследуется задача упаковки гистограмм, состоящих из трёх столбиков. Предлагается алгоритм линейной трудоёмкости, который строит упаковку длины не более $3 \text{OPT} + 2$. Если каждая гистограмма содержит столбик выше $\frac{1}{2}$, то доказано, что задача остаётся NP-трудной.

Результатами данной работы являются уточнённые оценки для ранее разработанных алгоритмов как для общего, так и для частных случаев задачи 2-ВСРР. Найдены примеры достижимости оценок. Для частного случая, когда высота каждого столбика не превышает $h \leq \frac{1}{2}$, предлагается алгоритм A_h линейной трудоёмкости, который строит упаковку длины не более $\frac{1}{1-h} \text{OPT} + 2$. Приводится пример асимптотической достижимости этой оценки. Кроме того, рассматривается задача упаковки неограниченного числа 2-Г, принадлежащих к одному из $k = \text{const}$ типов. В этой задаче необходимо найти упаковку максимальной плотности. Для этого разработан алгоритм K , который строит оптимальную упаковку с трудоёмкостью $O(v_k^3)$, где v_k — верхняя оценка на число вариантов заполнения одной ячейки первыми столбиками 2-Г k типов.

1. Постановка задачи

Пусть заданы горизонтальная полубесконечная (ограниченная слева) полоса единичной высоты и множество гистограмм S , $|S| = n$, состоящих из двух столбиков. Для каждой 2-Г $i \in S$ обозначим высоту первого

(левого) столбика через $a_i \in (0, 1]$, а высоту второго (правого) — через $b_i \in (0, 1]$. Разобьём полосу на одинаковые прямоугольники единичной высоты и ширины, которые назовём *ячейками*, и пронумеруем их натуральными числами, начиная с левого края полосы.

Определение 1. 2-Г i *невозрастающая* (*неубывающая*), если $a_i \geq b_i$ ($a_i \leq b_i$).

Определение 2. Упаковкой множества гистограмм S назовём функцию $p: S \rightarrow \mathbb{Z}^+$, которая каждой гистограмме i ставит в соответствие целое число $p(i)$, соответствующее номеру ячейки полосы, в которую попадает её первый столбик. Таким образом, в упаковке p столбики 2-Г i занимают ячейки $p(i)$ и $p(i) + 1$.

Определение 3. Упаковка *допустима*, если сумма высот столбиков, попавших в каждую ячейку полосы, не превосходит 1.

Далее рассматриваются только допустимые упаковки.

Определение 4. Длиной $L(p)$ упаковки p назовём число ячеек полосы, в которых расположен хотя бы один столбик.

Можно считать, что любая упаковка p начинается с первой ячейки и в каждой из $L(p)$ ячеек находится хотя бы один столбик. Если это не так, то всю упаковку или её часть можно сдвинуть влево.

Задача 2-ВСРР состоит в построении допустимой упаковки множества гистограмм S минимальной длины (т. е. с использованием минимального числа ячеек).

Впервые задача 2-ВСРР сформулирована в виде задачи булева линейного программирования в [2]. Ниже повторим эту формулировку для удобства читателя. Введём переменные

$$x_{ij} = \begin{cases} 1, & \text{если первый столбик 2-Г } i \text{ находится в ячейке } j, \\ 0 & \text{иначе,} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{если ячейка } j \text{ содержит хотя бы один столбик,} \\ 0 & \text{иначе.} \end{cases}$$

Тогда 2-ВСРР запишется в следующем виде:

$$\sum_j y_j \rightarrow \min_{x_{ij}, y_j \in \{0,1\}}, \quad (1)$$

$$\sum_j x_{ij} = 1, \quad i \in S, \quad (2)$$

$$\sum_i a_i x_{ij} + \sum_k b_k x_{kj-1} \leq y_j, \quad 1 \leq j \leq L(p). \quad (3)$$

Целевая функция (1) — это число ячеек, занятых хотя бы одним столбиком. Условия (2) гарантируют, что все 2-Г из множества S упакованы в полосу. Из неравенств (3) следует, что в каждой ячейке сумма высот столбиков не превышает высоты полосы.

2. Уточнённые оценки

2.1. Алгоритм A .

Жадный алгоритм G . В [2] предложен жадный алгоритм GA трудоёмкости $O(n^2)$, который используется в качестве процедуры в алгоритме A [2]. В настоящей работе предлагается использовать другую версию жадного алгоритма, которая называется G [1]. Применение этого алгоритма на втором этапе алгоритма A позволит реализовать его с меньшей трудоёмкостью. Работа G заключается в следующем. Пусть P — произвольно упорядоченное множество элементов из S . Первый элемент из P размещается в ячейках 1 и 2 и удаляется из P . Удалённые из P элементы уже упакованы, и они в дальнейшем не перемещаются. Далее выполняется следующая типовая процедура. Для первой гистограммы из P ищется самое левое положение, не нарушающее допустимости упаковки, и порядка 2-Г. Тем самым для произвольной 2-Г j справедливы неравенства $p(j) \geq p(i)$ для всех $i \in \{1, 2, \dots, j-1\}$ (см. рис. 1). Расположение гистограммы в упаковке фиксируется, а сама 2-Г удаляется из P . Алгоритм останавливается, когда $P = \emptyset$. Трудоёмкость G равна $O(n)$.

Очевидно, что в зависимости от порядка гистограмм в множестве P алгоритм G может строить разные решения.

Уточнённая оценка точности для алгоритма A .

Определение 5. 2-Г назовём *большой*, если хотя бы один её столбик выше $\frac{1}{2}$. Также *большим* назовём сам столбик выше $\frac{1}{2}$; иначе он *маленький*.

Напомним алгоритм A для решения задачи 2-BCPP из [2]. Он состоит из трёх этапов. Первый подготовительный этап заключается в объединении гистограмм таким образом, чтобы по возможности каждая 2-Г стала большой. Для этого пара гистограмм i и j , у которых $a_i, b_i, a_j, b_j \leq \frac{1}{2}$, объединяются в одну новую 2-Г с высотой столбиков $a_i + a_j$ и $b_i + b_j$. В результате множество гистограмм S преобразуется: из него удаляются две гистограммы i и j и добавляется одна новая 2-Г. За один просмотр гистограмм у всех 2-Г, кроме, возможно, одной, хотя бы один столбик станет большим. На втором этапе алгоритма обновлённое множество 2-Г разбивается на два подмножества S_1 и S_2 . В S_1 помещаются невозрастающие, а в S_2 — неубывающие гистограммы (рис. 2b). Множество элементов

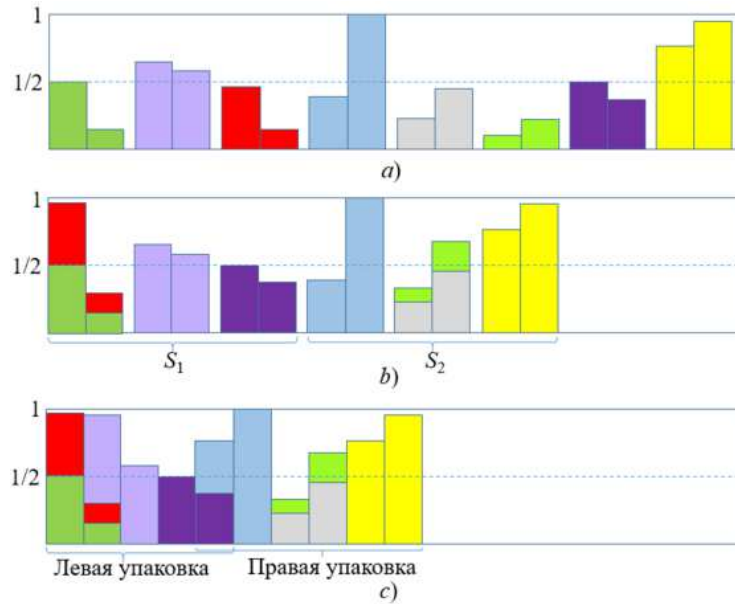


Рис. 2. Иллюстрация работы алгоритма A : а) множество S ; б) множества объединённых гистограмм; в) упаковка, построенная алгоритмом A

из S_1 пакуется с помощью алгоритма G слева направо, а множество элементов из S_2 пакуется аналогом алгоритма G справа налево. Отметим, что для упаковки множеств S_1 и S_2 в [2] использовался алгоритм GA квадратичной трудоёмкости. Получаются две упаковки: левая и правая. На третьем этапе алгоритма A правая упаковка элементов множества S_2 сдвигается максимально влево без нарушения допустимости упаковки (рис. 2с).

Теорема 1. Алгоритм A с трудоёмкостью $O(n)$ строит 2-приближённое решение для задачи 2-BCPP, и эта оценка достижима.

ДОКАЗАТЕЛЬСТВО. В [2] доказано, что в результате работы алгоритма A плотность упаковки всех ячеек, кроме, возможно, одной, больше $\frac{1}{2}$. Таким образом, алгоритм строит упаковку для 2-BCPP, длина которой не более $2 \text{OPT} + 1$. Оценка на длину построенной упаковки остаётся справедливой, если на втором этапе упаковка элементов множеств S_1 и S_2 осуществляется алгоритмом G . Доказательство этого утверждения совпадает с доказательством теоремы 1 из [2], если заменить алгоритм GA на G .

Обозначим через b высоту последнего столбика в левой упаковке, а через a — высоту первого столбика в правой упаковке. Кроме того, пусть

$X = \sum_{i \in S} a_i$, $Y = \sum_{i \in S} b_i$, а $L_A(n)$ — длина упаковки, построенной алгоритмом A . Так как после сдвига правой упаковки максимально влево, в худшем случае плотность всех ячеек кроме одной больше $\frac{1}{2}$, справедливы неравенства

$$\frac{X + Y}{L_A(n) - 1} > \frac{X + Y - a - b}{L_A(n) - 1} > \frac{1}{2}.$$

Помимо этого, очевидно, что $\text{OPT} \geq X + Y$, откуда получаем

$$\frac{\text{OPT}}{L_A(n) - 1} > \frac{1}{2}$$

или $L_A(n) < 2\text{OPT} + 1$. Однако, так как длина упаковки — это целое число, окончательно имеем $L_A(n) \leq 2\text{OPT}$.

Первый этап алгоритма A может быть реализован с трудоёмкостью $O(n)$ следующим образом. Положим $M = \emptyset$. Просматриваем гистограммы в порядке нумерации. Если у очередной 2-Г первый столбик большой и больше второго, то помещаем её в множество S_1 . Если у очередной 2-Г второй столбик большой и больше первого, то помещаем её в множество S_2 . Если оба столбика маленькие и $M = \emptyset$, то помещаем текущую гистограмму в M и продолжаем просмотр. Если оба столбика маленькие и $M \neq \emptyset$, то объединяем текущую гистограмму с гистограммой, находящейся в M . Если полученная новая объединённая гистограмма имеет хотя бы один столбик больше $\frac{1}{2}$, то исключаем её из M и помещаем в S_1 , если её первый столбик больше второго, иначе помещаем её в S_2 . Если же после объединения текущей гистограммы с гистограммой из M получается новая гистограмма, в которой оба столбика маленькие, то оставляем новую объединённую гистограмму в M и переходим к рассмотрению следующей гистограммы из P . Таким образом, в результате одного просмотра всех гистограмм будут сформированы множества S_1 и S_2 .

На втором этапе с помощью алгоритма G независимо строятся упаковки гистограмм из множеств S_1 и S_2 . На каждом шаге алгоритма G к уже построенной упаковке добавляется одна гистограмма. В силу того, что большие 2-Г невозрастающие (неубывающие) в соответствующих упаковках, в каждой ячейке упаковок множеств S_1 и S_2 могут находиться не более двух столбиков. Значит, для произвольной гистограммы f достаточно проверить выполнение неравенства $a_f + b_{f-1} \leq 1$ для определения её лучшего допустимого положения. Если неравенство выполняется, то первый столбик 2-Г f размещается в последней непустой

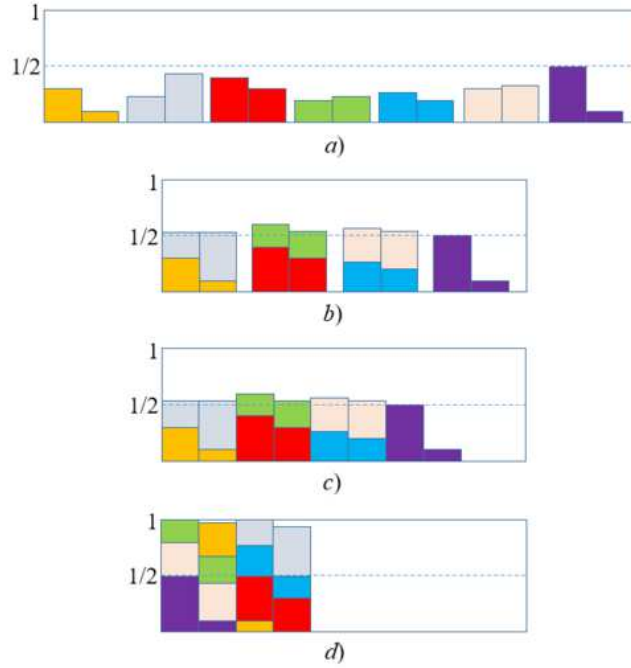


Рис. 3. Пример достижимости оценки $L_A(n) = 2 \text{ OPT}$: а) множество S ; б) множество S^* объединённых невозрастающих 2-Г; в) упаковка, построенная алгоритмом A ; д) оптимальная упаковка

ячейке упаковки, иначе — в первой пустой ячейке. Таким образом, трудоёмкость процесса нахождения лучшего положения для текущей гистограммы ограничена величиной $O(1)$. Число шагов алгоритма G не превосходит n . Следовательно, трудоёмкость второго этапа алгоритма равна $O(n)$. Трудоёмкость третьего этапа равна $O(1)$. Таким образом, трудоёмкость алгоритма A равна $O(n)$.

Рассмотрим следующий пример (рис. 3а). Пусть $|S| = 7$ и гистограммы имеют следующие высоты столбиков: $a_1 = 0,3$, $b_1 = 0,1$, $a_2 = \frac{7}{30}$, $b_2 = \frac{13}{30}$, $a_3 = 0,4$, $b_3 = 0,3$, $a_4 = 0,2$, $b_4 = \frac{7}{30}$, $a_5 = \frac{4}{15}$, $b_5 = 0,2$, $a_6 = 0,3$, $b_6 = \frac{1}{3}$, $a_7 = 0,5$, $b_7 = 0,1$. На первом этапе алгоритм A просматривает 2-Г в порядке нумерации и объединяет их, пока все они, кроме последней, не станут большими. В результате объединяются 2-Г 1 и 2, 3 и 4, 5 и 6, формируя новый обновлённый список гистограмм S^* , где $a_1^* = 0,3 + \frac{7}{30} = \frac{8}{15}$, $b_1^* = 0,1 + \frac{13}{30} = \frac{8}{15}$, $a_2^* = 0,6$, $b_2^* = \frac{8}{15}$, $a_3^* = \frac{17}{30}$, $b_3^* = \frac{8}{15}$, $a_4^* = 0,5$, $b_4^* = 0,1$ (рис. 3б).

Все 2-Г в S^* невозрастающие, поэтому правая упаковка пуста. На втором этапе 2-Г пакуются алгоритмом G . Первая 2-Г размещается в ячейках 1 и 2 и удаляется из S^* . Так как для второй 2-Г сумма $a_2^* + b_1^* = \frac{17}{15}$ больше единицы, алгоритм для неё ставит в соответствие ячейку полосы с номером 3 и удаляет её из S^* . Аналогично алгоритм для третьей и четвёртой 2-Г ставит в соответствие ячейки полосы с номерами 5 и 7 соответственно и завершает свою работу (рис. 3с). Длина построенной упаковки $L_A(n) = 8$. Однако 2-Г из множества S могут быть упакованы оптимально в четыре ячейки (рис. 3d). При этом сумма высот столбиков в первой ячейке равна $0,5 + 0,3 + 0,2 = 1$, во второй ячейке $0,1 + \frac{1}{3} + \frac{7}{30} + 0,3 = \frac{29}{30}$, в третьей ячейке $0,1 + 0,4 + \frac{4}{15} + \frac{7}{30} = 1$, а в последней ячейке оптимальной упаковки $0,3 + 0,2 + \frac{13}{30} = \frac{28}{30}$. Таким образом, в рассмотренном примере $L_A(n) = 2 \text{ OPT}$. Теорема 1 доказана.

После первого этапа алгоритма A несколько гистограмм объединяются в одну гистограмму, что уменьшает число допустимых упаковок. Кроме того, длина упаковки, строящейся алгоритмом G , существенно зависит от порядка элементов в списке P . В [2] проведён следующий численный эксперимент. Для оценки влияния процедуры объединения столбиков в алгоритме A реализован алгоритм, который назовём A_h , состоящий только из второго и третьего этапов алгоритма A . Алгоритм A_h за один просмотр разбивает все гистограммы на два множества — невозрастающих и неубывающих 2-Г. При этом соответствующие множества S_1 и S_2 содержат произвольные 2-Г (в алгоритме A после первого этапа эти множества содержали большие 2-Г). Затем алгоритм A_h строит левую и правую упаковки с помощью алгоритма G и сдвигает правую упаковку максимально влево, не нарушая допустимости.

2.2. Упаковка маленьких гистограмм. Через $2\text{-BCPP}|h$ обозначим частный случай задачи 2-BCPP, когда высота обоих столбиков каждой гистограммы не превосходит $h \leq \frac{1}{2}$, и докажем следующую теорему.

Теорема 2. Для задачи $2\text{-BCPP}|h$ алгоритм A_h с трудоёмкостью $O(n)$ строит упаковку длины не более $\frac{1}{1-h} \text{ OPT} + 2$, и эта оценка асимптотически достижима.

ДОКАЗАТЕЛЬСТВО. Пусть высота всех столбиков не превосходит h . Сначала A_h строит левую упаковку невозрастающих гистограмм, размещая 2-Г одну над другой в первых двух ячейках, пока возможно. После этого возможны два случая:

- 1) во вторую ячейку не помещается первый столбик очередной 2-Г;
- 2) во вторую ячейку можно поместить первый столбик очередной 2-Г.

В первом случае плотность упаковки первых двух ячеек больше $1 - h$. Во втором случае в результате дальнейшей работы алгоритма во вторую

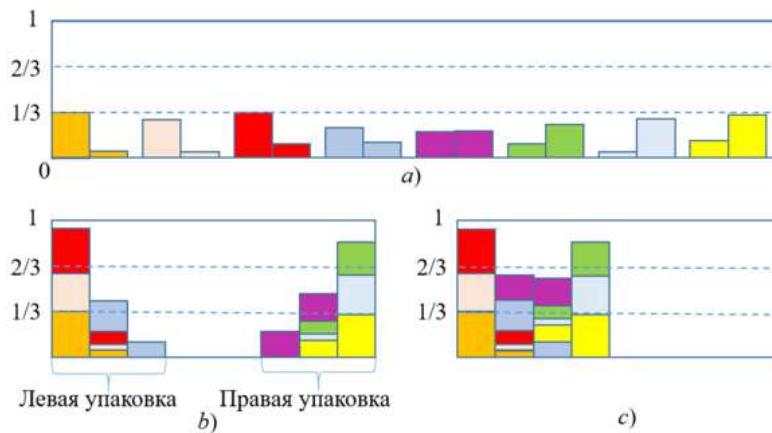


Рис. 4. Пример работы алгоритма A_h при $h = \frac{1}{3}$: а) множество 2-Г; б) левая и правая упаковки; в) упаковка, построенная алгоритмом A_h

ячейку попадают первые столбики других 2-Г, пока незаполненное пространство ячейки не станет меньше h . Значит, и в этом случае плотность первых двух ячеек больше $1 - h$.

Далее A_h аналогично заполняет следующие ячейки. При размещении последней гистограммы в левой упаковке ячейки, в которые попадут её столбики, могут иметь плотность не более $1 - h$ (рис. 4б). Таким образом, в каждой ячейке левой упаковки, кроме, возможно, двух последних, плотность больше $1 - h$.

Правая упаковка неубывающих гистограмм строится аналогичным способом зеркально (справа налево), обеспечивая плотность всех ячеек правой упаковки, кроме, возможно, первых двух, больше $1 - h$.

На последнем этапе алгоритма A_h правая упаковка сдвигается максимально влево, не нарушая допустимости упаковки. Обозначим через a_1 и b_1 суммарную высоту столбиков в предпоследней и последней ячейках левой упаковки соответственно, а через a_2 и b_2 — суммарную высоту столбиков во второй и первой ячейках правой упаковки соответственно. После сдвига правой упаковки максимально влево получим один из следующих случаев:

- 1) $b_1 + b_2 > 1$;
- 2) $a_1 + b_2 \leq 1$, $b_1 + a_2 \leq 1$;
- 3) $b_1 + b_2 \leq 1$.

В первом случае левая и правая упаковки не могут объединиться ни в одной ячейке. Это означает, что совокупная плотность последней ячейки левой упаковки и первой ячейки правой упаковки больше $\frac{1}{2}$. Однако в силу неубывания и невозрастания гистограмм в разных упаковках

предпоследняя ячейка левой упаковки и вторая ячейка правой упаковки также имеют совокупную плотность больше $\frac{1}{2}$. Таким образом, четыре ячейки получившейся упаковки имеют плотность больше $\frac{1}{2}$, а все остальные ячейки имеют плотность более $1 - h$, что даёт оценку $\frac{1}{1-h} \text{OPT} + 2$.

Во втором случае упаковки объединились в двух ячейках, плотность которых может быть менее $1 - h$. Плотность же всех остальных ячеек, кроме общих двух, более $1 - h$, что также позволяет оценить длину упаковки как $\frac{1}{1-h} \text{OPT} + 2$.

В третьем случае упаковки объединились только в одной ячейке. Без ограничения общности положим, что упаковки нельзя объединить в двух ячейках из-за невыполнения неравенства $a_1 + b_2 \leq 1$. Тогда общая ячейка двух упаковок и ячейка с предыдущим номером имеют совокупную плотность больше $\frac{1}{2}$. Плотность ячейки со следующим номером после общей ячейки может быть менее $1 - h$. Плотность же всех остальных ячеек более $1 - h$, что также даёт оценку $\frac{1}{1-h} \text{OPT} + 2$. Оценка $\frac{1}{1-h} \text{OPT} + 2$ получена во всех рассмотренных случаях.

В результате одного просмотра всех гистограмм будут сформированы множества S_1 и S_2 . Затем строится упаковка элементов множеств S_1 и S_2 с помощью алгоритма G . На каждом шаге алгоритма G к уже построенной упаковке добавляется одна гистограмма. Трудоёмкость процесса нахождения лучшего положения для текущей гистограммы ограничена функцией $O(1)$. Число шагов алгоритма G не более n . Следовательно, трудоёмкость построения левой и правой упаковки равна $O(n)$. Трудоёмкость сдвига правой упаковки равна $O(1)$. Таким образом, алгоритм A_h имеет трудоёмкость, равную $O(n)$.

Рассмотрим следующий пример. Пусть 1 (высота полосы) делится на h без остатка. Имеем $\frac{3}{h}$ гистограмм, из которых $\frac{2}{h} - 1$ невозрастающие с высотами столбиков $a_i = \frac{h}{2}$, $b_i = \frac{h}{2} - \varepsilon$, $i \in \{1, 2, \dots, \frac{2}{h} - 1\}$, $\frac{1}{h}$ невозрастающие с высотами столбиков $a_i = h$, $b_i = h$, $i \in \{\frac{2}{h}, \frac{2}{h} + 1, \dots, \frac{3}{h} - 1\}$, и одна неубывающая с высотами столбиков $a_{3/h} = \frac{h}{2}$, $b_{3/h} = \frac{h}{2} + \varepsilon(\frac{2}{h} - 1)$. Для любого $0 < \varepsilon < \frac{h^2}{4-2h}$ выполняются неравенства $0 < \varepsilon(\frac{2}{h} - 1) < \frac{h}{2}$. Значит, после размещения алгоритмом первых $\frac{2}{h} - 1$ гистограмм незанятым окажется пространство $\frac{h}{2}$ в первой ячейке, более $\frac{h}{2}$ и менее h — во второй. Затем будут упакованы $\frac{1}{h}$ невозрастающих гистограмм с высотой обоих столбиков h , начиная с третьей ячейки, так как размещение хотя бы одной ещё не упакованной невозрастающей 2-Г в первых двух ячейках нарушит допустимость. Заметим, что после сдвига правой упаковки, состоящей из единственной неубывающей 2-Г, общих ячеек у левой и правой упаковок нет. Длина приближённой упаковки, построенной

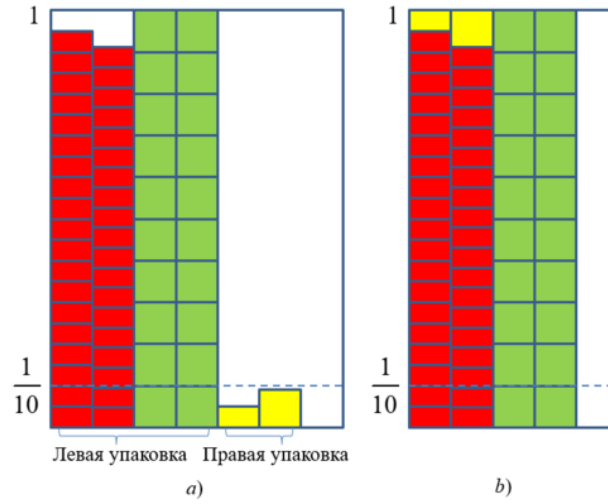


Рис. 5. Пример асимптотической достижимости оценки для алгоритма A_h , $h = 0,1$, $\varepsilon = 0,002$: а) упаковка, построенная алгоритмом A_h ; б) оптимальная упаковка

алгоритмом A_h для данного примера, равна $L_{A_h}(n) = 6$ (рис. 5а). Оптимальным образом гистограммы упаковываются в 4 ячейки (рис. 5б). Таким образом, при $h \rightarrow 0$ достигается оценка $\text{OPT} + 2$ на длину упаковки, построенной алгоритмом. Теорема 2 доказана.

2.3. Уточнённая оценка точности для алгоритма M . В работе [3] впервые предложен алгоритм M . Приведём его краткое описание для решения задачи 2-ВСРР в случае, когда каждая 2-Г большая и невозрастающая.

Определение 6. Две 2-Г образуют t -объединение, если t ячеек упаковки содержат столбики обеих 2-Г.

Пусть каждая 2-Г большая и невозрастающая. Тогда в каждой ячейке может находиться не более двух столбиков, а две 2-Г могут образовать только 1-объединение. Очевидно, что n 2-Г до упаковки занимают $2n$ ячеек полосы. Каждое 1-объединение уменьшает длину упаковки на 1.

По множеству гистограмм S построим граф $G_1 = (V_1, E_1)$, в котором вершины соответствуют гистограммам, $|V_1| = |S| = n$, при этом $(i, j) \in E_1$, если 2-Г i и j могут образовать 1-объединение. На первом шаге алгоритма M в графе G_1 строится максимальное паросочетание мощности m_1 . Результатом являются $n - m_1$ 2- и 3-Г, которые соответствуют

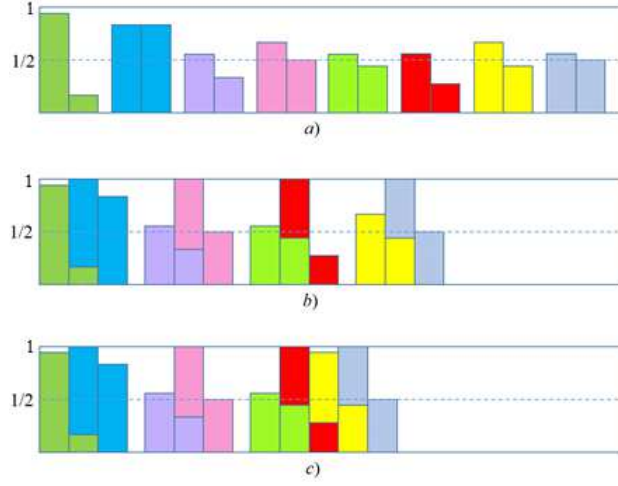


Рис. 6. Пример работы алгоритма M : а) множество больших невозрастающих 2-Г; б) первое паросочетание, $m_1 = 4$; в) второе паросочетание, $m_2 = 1$

множеству вершин V_2 нового графа $G_2 = (V_2, E_2)$. В графе G_2 имеется ребро (i, j) , если гистограммы i и j могут образовать 1-объединение. На произвольном шаге k строится очередное максимальное паросочетание мощности m_k в соответствующем графе G_k . Алгоритм останавливается, если следующий граф G_{k+1} пустой, $E_{k+1} = \emptyset$. На рис. 6 изображён пример работы алгоритма M .

Если в графе G_{k+1} нет рёбер, то длина упаковки, построенной алгоритмом M , равна

$$L_M(n) = 2n - m_1 - m_2 - \dots - m_k. \quad (4)$$

В [3] доказано, что оптимальную упаковку можно также построить путём последовательного нахождения паросочетаний. Пусть в оптимальной упаковке на шаге k строится паросочетание мощности m_k^* . Тогда длина оптимальной упаковки равна

$$\text{ОПТ} = 2n - m_1^* - m_2^* - \dots - m_q^*, \quad q \geq 1. \quad (5)$$

Кроме того, в [3] приводится доказательство неравенств

$$m_2^* + \dots + m_q^* \leq m_1^* \leq m_1, \quad (6)$$

а также того факта, что алгоритм M строит $3/2$ -приближённое решение для задачи 2-ВСПР, если все 2-Г большие невозрастающие. Оказывается, эту оценку можно уменьшить, что показано в следующей теореме.

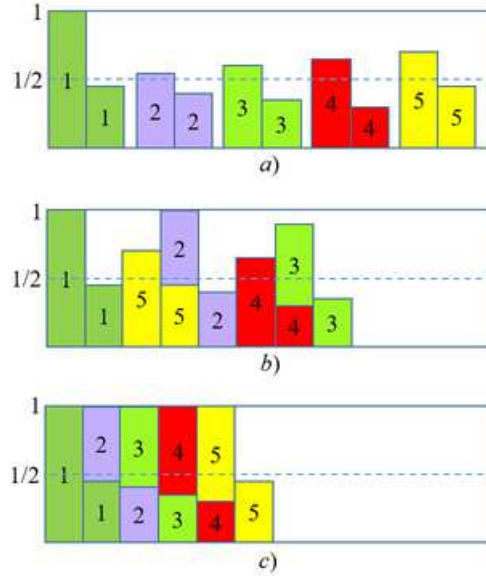


Рис. 7. Пример достижимости $L_M(n) = \frac{3}{2} \text{OPT} - 1$: а) множество S ; б) упаковка, построенная алгоритмом M ; в) оптимальная упаковка.

Теорема 3. Если все 2-Г большие и невозрастающие, то для 2-ВСРР алгоритм M строит упаковку, длина которой не превосходит $\frac{3}{2} \text{OPT} - 1$, и эта оценка достижима.

ДОКАЗАТЕЛЬСТВО. Из (4)–(6) следует, что $L_M(n) \leq 2n - m_1$, $\text{OPT} \geq 2n - 2m_1$, или $m_1 \geq n - \frac{1}{2} \text{OPT}$. Тогда

$$L_M(n) \leq 2n - m_1 \leq \frac{1}{2} \text{OPT} + n.$$

Очевидно, что n больших неубывающих 2-Г могут образовать не более $n - 1$ 1-объединений. Значит, $\text{OPT} \geq 2n - (n - 1) = n + 1$. Окончательно имеем

$$L_M(n) \leq \frac{1}{2} \text{OPT} + n \leq \frac{3}{2} \text{OPT} - 1.$$

Рассмотрим следующий пример. Пусть $|S| = 5$ и гистограммы имеют следующие высоты столбиков: $a_1 = 1$, $b_1 = 0,45$, $a_2 = 0,55$, $b_2 = 0,4$, $a_3 = 0,6$, $b_3 = 0,35$, $a_4 = 0,65$, $b_4 = 0,3$, $a_5 = 0,7$, $b_5 = 0,45$ (рис. 7а). Очевидно, что мощность первого паросочетания m_1 не более 2, так как множество S состоит из пяти 2-Г. Если в первом паросочетании, построенном алгоритмом, 1-объединение образуют 2-Г 2 и 5, 3 и 4, то другие 1-объединения невозможны, и алгоритм останавливается. Длина приближённой упаковки $L_M(n) = 8$ (рис. 7б). Оптимальная упаковка может

быть получена, если 2-Г последовательно 1-объединятся в порядке нумерации, её длина $\text{OPT} = 6$ (рис. 7с). Таким образом, $L_M(n) = \frac{3}{2} \text{OPT} - 1$. Теорема 3 доказана.

3. Упаковка 2-Г ограниченного числа типов

Пусть имеется неограниченное число 2-Г, которые принадлежат к одному из k типов, $k \geq 1$, $k \in \mathbb{N}$. В частности, если $k = 1$, то все 2-Г одинаковые. Так как гистограмм бесконечно много, длина упаковки не имеет смысла, поэтому будем оценивать её плотность. Если $k = \text{const}$, то число различных комбинаций заполнения каждой ячейки конечно. Следовательно, в любой упаковке рано или поздно встретится ячейка, комбинация первых столбиков 2-Г в которой уже встречалась в некоторой ячейке с меньшим номером (рис. 8). Очевидно, что если упаковка строится детерминированным алгоритмом, то начиная с этой ячейки в упаковке будет бесконечно повторяться заполнение некоторой последовательности ячеек. Такую последовательность назовём *регулярной частью* упаковки.

Определение 7. В каждой упаковке бесконечного числа 2-Г k типов встречается последовательность ячеек, в которой первая и последняя ячейки заполнены одинаковой комбинацией первых столбиков 2-Г. *Регулярной частью* упаковки назовём первую такую последовательность без последней ячейки.

Задача 2-BCPP $|k$ состоит в построении допустимой упаковки неограниченного числа 2-Г k типов максимальной плотности.

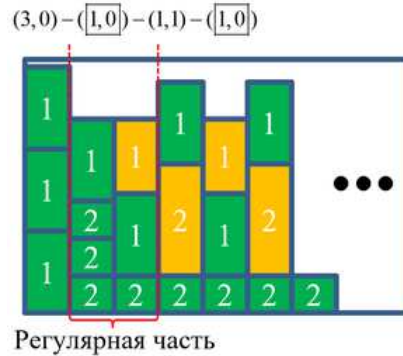


Рис. 8. Пример регулярной части в упаковке 2-Г двух типов

3.1. 2-Г одного типа. В этом случае $k = 1$. В [1] доказано следующее утверждение. Если задан порядок упаковки невозрастающих (неубывающих) гистограмм, то жадный алгоритм G (или его аналог, пакующий

гистограммы справа налево) строит упаковку минимальной длины. Воспользуемся этим утверждением. Так как 2-Г принадлежат к единственному типу, все они либо неубывающие, либо невозрастающие. Кроме того, можно считать, что для упаковки одинаковых 2-Г задан любой порядок. Таким образом, справедливо

Утверждение 1. *Если все 2-Г одного типа, то алгоритм G строит упаковку максимальной плотности для задачи 2-BCPP[1].*

3.2. 2-Г $k \geq 2$ типов. Рассмотрим случай $k = 2$ и затем обобщим результат. Имеем неограниченное число 2-Г двух типов с высотами столбиков a_1, b_1 и a_2, b_2 соответственно. Обозначим через s_1 и s_2 число 2-Г первого и второго типов соответственно, которые могут 2-объединиться, не нарушая допустимости, т. е. $s_1 = \lfloor \max^{-1}\{a_1, b_1\} \rfloor$, $s_2 = \lfloor \max^{-1}\{a_2, b_2\} \rfloor$. Предположим, что $s_1, s_2 \geq 1$, иначе либо ни одна 2-Г не может быть упакована в полосу, либо получим случай, рассмотренный в п. 3.1. Обозначим через x_i (y_i) число первых столбиков 2-Г первого (второго) типа в ячейке i . Будем присваивать код (x_i, y_i) каждому варианту заполнения ячейки i . Число допустимых комбинаций x_i и y_i ограничено величиной $v = (s_1 + 1)(s_2 + 1)$. Так как гистограммы состоят из двух столбиков, каждому коду (x_i, y_i) соответствует величина $h_{x_i, y_i} = b_1 x_i + b_2 y_i$, отражающая сумму высот столбиков, которые попадут в следующую ячейку. Любая упаковка может быть представлена последовательностью кодов до нахождения регулярной части.

Определение 8. *Доминирующей упаковкой для кода (x_i, y_i) назовём последовательность кодов, оканчивающуюся на (x_i, y_i) , с максимальной плотностью.*

В этом пункте предлагается алгоритм K для решения 2-BCPP[2]. Работа алгоритма заключается в последовательном нахождении доминирующей упаковки для каждого кода заполнения текущей ячейки. Сначала алгоритм K рассматривает всевозможные коды заполнения первой ячейки (x_1, y_1) , $x_1 \in \{0, 1, \dots, s_1\}$, $y_1 \in \{0, 1, \dots, s_2\}$. Если для очередного кода нарушается допустимость в первой или второй ячейке, то алгоритм удаляет код. Затем для каждого значения h_{x_1, y_1} находятся все допустимые комбинации заполнения второй ячейки (x_2, y_2) . Заметим, что в третьей ячейке всегда будет высота h_{x_2, y_2} , которая уже появлялась во второй ячейке при каких-то значениях x_1 и y_1 . Из последовательностей кодов, оканчивающихся на (x_2, y_2) , выбирается доминирующая упаковка (рис. 9). Если среди доминирующих упаковок встречается повторение кода в последовательности или последним является код $(0, 0)$,

то соответствующая упаковка с найденной регулярной частью запоминается. Для остальных доминирующих упаковок рассматриваются всевозможные (x_3, y_3) (рис. 10) и т. д., пока для очередной ячейки во всех доминирующих упаковках не будет найдена регулярная часть (рис. 11). Затем все упаковки, в которых была найдена регулярная часть, сравниваются, и выбирается упаковка с максимальной плотностью (рис. 12).

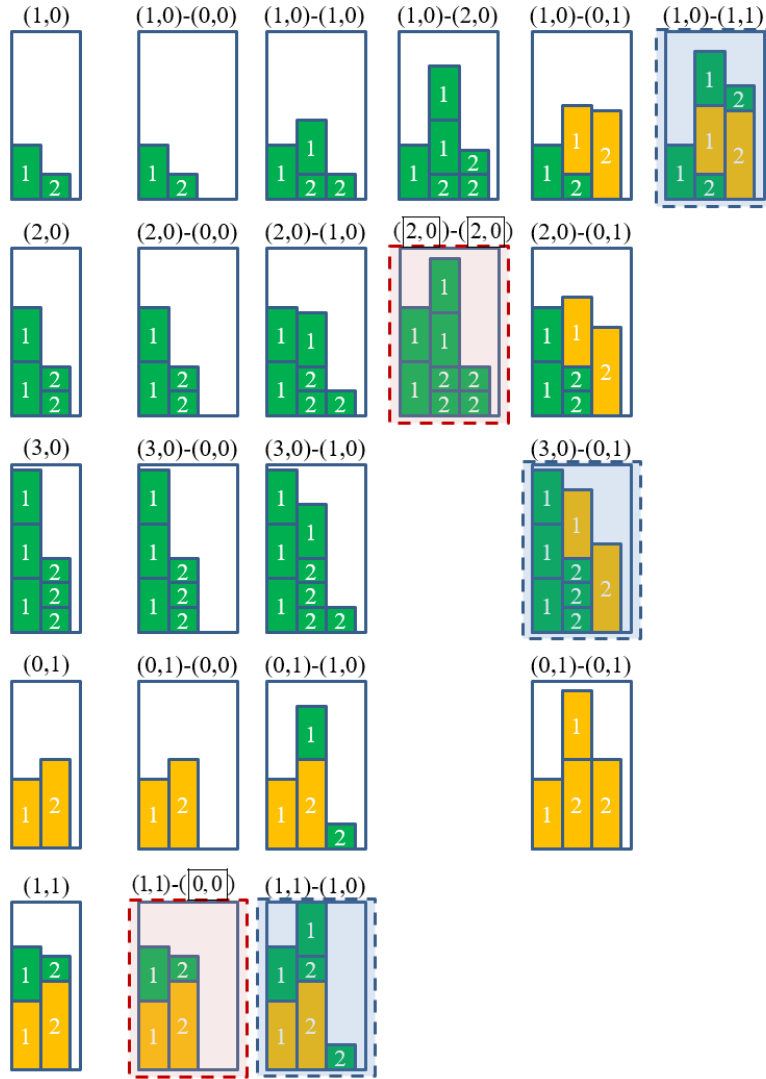


Рис. 9. Пример нахождения доминирующих упаковок для всех комбинаций заполнения второй ячейки

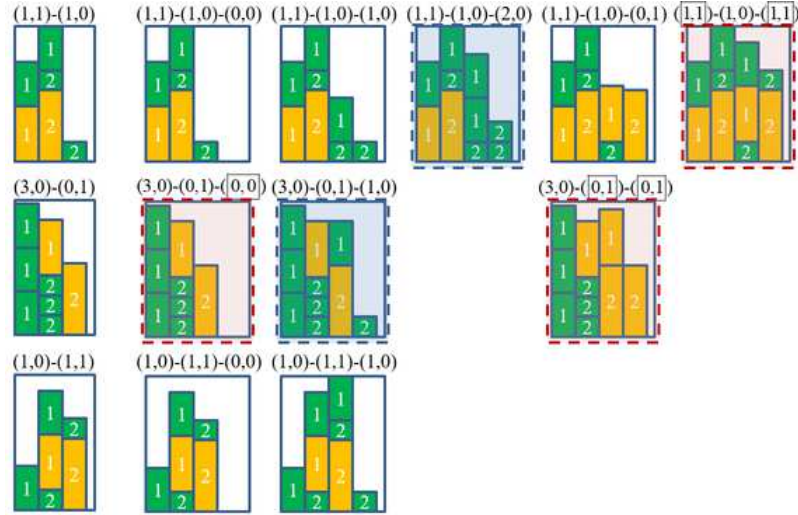


Рис. 10. Пример нахождения доминирующих упаковок для всех комбинаций заполнения третьей ячейки

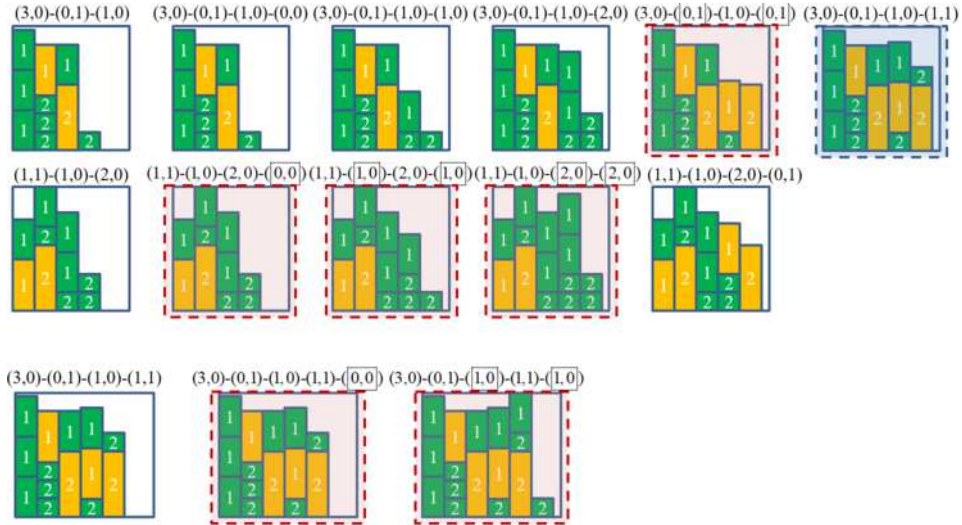
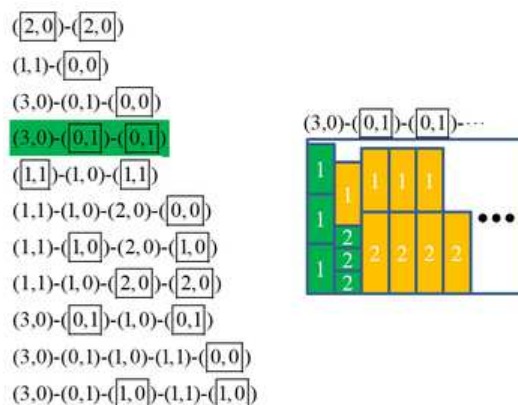


Рис. 11. Пример нахождения доминирующих упаковок для всех комбинаций заполнения четвёртой и пятой ячеек

На рис. 9–11 в упаковках, выделенных красной штриховой линией, имеется регулярная часть (код в прямоугольнике); упаковки, выделенные синей штриховой линией, доминирующие без регулярной части.



Теорема 4. Алгоритм K строит упаковку максимальной плотности с трудоёмкостью $O(v^3)$ для задачи 2-BCPP|2.

В ходе работы алгоритма ищется одна доминирующая упаковка для каждого кода заполнения текущей ячейки. Число возможных кодов заполнения каждой ячейки не более $O(v)$. Для каждого кода рассматриваются не более $O(v)$ вариантов упаковки. Следовательно, трудоёмкость нахождения всех доминирующих упаковок для одной ячейки равняется $O(v^2)$. Количество итераций не более $O(v)$, так как далее гарантированно встретится повторение кода в последовательности. Таким образом, трудоёмкость алгоритма K равна $O(v^3)$. Теорема 4 доказана.

Замечание 1. Алгоритм K может быть применён для нахождения точного решения задачи 2-BCPP $|k$ для любого $k \geq 2$, $k \in \mathbb{N}$. При этом число допустимых комбинаций первых столбиков 2-Г k типов ограничено числом $v_k = (s_1 + 1)(s_2 + 1) \dots (s_k + 1)$. Трудоемкость алгоритма, когда имеется k типов 2-Г, равна $O(v_k^3)$.

Заключение

Рассмотрена задача упаковки гистограмм, состоящих из двух столбиков, в полосу минимальной длины. Предложены уточнённые априорные оценки точности для ранее разработанных приближённых алгоритмов. Показана достижимость этих оценок. Кроме того, впервые исследовался частный случай задачи 2-ВСРР, когда высота обоих столбиков каждой гистограммы не превосходит h . Доказано, что алгоритм A_h с трудоёмкостью $O(n)$ строит упаковку для этого частного случая задачи длины не более $\frac{1}{1-h} \text{OPT} + 2$, где OPT — минимальная длина упаковки. Приведён пример асимптотической достижимости этой оценки.

Кроме того, впервые рассматривается задача упаковки неограниченного числа 2-Г, принадлежащих к одному из k типов. Предложен алгоритм, строящий оптимальную упаковку для этой задачи, трудоёмкость которого равна $O(v_k^3)$, где v_k — верхняя оценка на число вариантов заполнения одной ячейки первыми столбиками 2-Г k типов, которые не нарушают допустимости упаковки.

Автор выражает благодарность А. И. Ерзину за ценные советы и конструктивную помощь в представлении результатов.

Финансирование работы

Исследование выполнено за счёт бюджета Новосибирского государственного университета. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Erzin A. I., Plotnikov R. V., Korobkin A. P., Melidi G. E., Nazarenko S. A.** Optimal investment in the development of oil and gas field // Mathematical optimization theory and operations research. Rev. Sel. Pap. 19th Int. Conf. MOTOR 2020 (Novosibirsk, Russia, July 6–10, 2020). Cham: Springer, 2020. P. 336–349. (Commun. Comput. Inf. Sci.; V. 1275).
2. **Erzin A. I., Melidi G. E., Nazarenko S. A., Plotnikov R. V.** Two-bar charts packing problem // Optim. Lett. 2021. V. 15, No. 6. P. 1955–1971.
3. **Erzin A. I., Melidi G. E., Nazarenko S. A., Plotnikov R. V.** A 3/2-approximation for big two-bar charts packing // J. Comb. Optim. 2021. V. 42, No. 1. P. 71–84.

4. **Erzin A. I., Melidi G. E., Nazarenko S. A., Plotnikov R. V.** A posteriori analysis of the algorithms for two-bar charts packing problem // Advances in optimization and applications. Rev. Sel. Pap. 12th Int. Conf. OPTIMA 2021 (Petrovac, Montenegro, Sep. 27–Oct. 1, 2021). Cham: Springer, 2021. P. 201–216. (Commun. Comput. Inf. Sci.; V. 1514.)
5. **Erzin A. I., Kononov A. V., Melidi G. E., Nazarenko S. A.** A $4/3 \times \text{OPT} + 2/3$ approximation for big two-bar charts packing problem // J. Math. Sci. 2023. V. 269, No. 6. P. 813–823.
6. **Erzin A. I., Kononov A. V., Nazarenko S. A., Sharankhaev K. I.** An $O(n \log n)$ -time algorithm for linearly ordered packing of 2-bar charts into $\text{OPT} + 1$ bins // Mathematical optimization theory and operations research. Rev. Sel. Pap. 22th Int. Conf. MOTOR 2023 (Yekaterinburg, Russia, July 2–8, 2023). Cham: Springer, 2023. P. 122–133. (Commun. Comput. Inf. Sci.; V. 1881).
7. **Barkel M., Delorme M.** Arcflow formulations and constraint generation frameworks for the two bar charts packing problem // INFORMS J. Comput. 2023. V. 35, No. 2. P. 475–494.
8. **Garey M. R., Johnson D. S.** Computers and intractability. A guide to the theory of NP-completeness. San Francisco: Freeman, 1979. 340 p.
9. **Vazirani V. V.** Approximation algorithms. Heidelberg: Springer, 2001. 396 p.
10. **Simchi-Levi D.** New worst-case results for the bin-packing problem // Nav. Res. Logist. 1994. V. 41, No. 4. P. 579–585.
11. **Dósa G.** The tight bound of first fit decreasing bin-packing algorithm is $\text{FFD}(I) \leq 11/9 \text{OPT}(I) + 6/9$ // Combinatorics, algorithms, probabilistic and experimental methodologies. Rev. Sel. Pap. 1st Int. Symp. (Hangzhou, China, Apr. 7–9, 2007). Heidelberg: Springer, 2007. P. 1–11. (Lect. Notes Comput. Sci.; V. 4614).
12. **Johnson D. S., Garey M. R.** A $71/60$ theorem for bin packing // J. Complex. 1985. V. 1, No. 1. P. 65–106.
13. **Yue M., Zhang L.** A simple proof of the inequality $\text{MFFD}(L) \leq 71/60 \times \text{OPT}(L) + 1$ for the MFFD bin-packing algorithm // Acta Math. Appl. Sin. 1995. V. 11, No. 3. P. 318–330.
14. **Fernandez de la Vega W., Lueker G. S.** Bin packing can be solved within $1 + \epsilon$ in linear time // Combinatorica. 1981. V. 1. P. 349–355.
15. **Karmarkar N., Karp R. M.** An efficient approximation scheme for the one dimensional bin-packing problem // 23rd Annu. Symp. Foundations of Computer Science (Chicago, USA, Nov. 3–5, 1982). Piscataway: IEEE, 1982. P. 312–320.
16. **Hoberg R., Rothvoss T.** A logarithmic additive integrality gap for bin packing // Proc. 28th Annu. ACM-SIAM Symp. Discrete Algorithms (Barcelona, Spain, Jan. 16–19, 2017). Philadelphia, PA: SIAM, 2017. P. 2616–2625.
17. **Kellerer H., Kotov V.** An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing // Oper. Res. Lett. 2003. V. 31. P. 35–41.

-
18. **Christensen H. I., Khanb A., Pokutta S., Tetali P.** Approximation and online algorithms for multidimensional bin packing: A survey // *Comput. Sci. Rev.* 2017. V. 24. P. 63–79.
 19. **Bansal N., Eliáš M., Khan A.** Improved approximation for vector bin packing // *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms* (Arlington, VA, USA, Jan. 10–12, 2016). Philadelphia, PA: SIAM, 2016. P. 1561–1579.
 20. **Woeginger G. J.** There is no asymptotic PTAS for two-dimensional vector packing // *Inf. Process. Lett.* 1997. V. 64, No. 6. P. 293–297.
 21. **Brucker P., Knust S.** *Complex scheduling*. Heidelberg: Springer, 2006. 286 p.
 22. **Goncharov E. N.** A greedy heuristic approach for the resource-constrained project scheduling problem // *Stud. Inform. Univers.* 2011. V. 9, No. 3. P. 79–90.
 23. **Гончаров Е. Н.** Стохастический жадный алгоритм для задачи календарного планирования с ограниченными ресурсами // *Дискрет. анализ и исслед. операций*. 2014. Т. 21, № 3. С. 11–24.
 24. **Hartmann S.** A self-adapting genetic algorithm for project scheduling under resource constraints // *Nav. Res. Logist.* 2002. V. 49. P. 433–448.
 25. **Kolisch R., Hartmann S.** Experimental investigation of heuristics for resource-constrained project scheduling: An update // *Eur. J. Oper. Res.* 2006. V. 174. P. 23–37.
 26. **Erzin A. I., Sharankhaev K. I.** Three-bar charts packing problem // *Commun. Comput. Inf. Sci.* 2023. V. 1739. P. 61–75.

Назаренко Степан Андреевич

Статья поступила
15 декабря 2023 г.
После доработки —
23 апреля 2024 г.
Принята к публикации
22 июня 2024 г.

UPDATED ESTIMATES FOR ALGORITHMS
FOR PACKING 2-BAR CHARTS IN A STRIP

S. A. Nazarenko

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

Abstract. We consider a two-bar charts packing problem in which it is necessary to pack bar charts consisting of two bars in a unit-height strip of minimum length. Each bar has a height of at most 1 and unit length. The problem under consideration is NP-hard and generalizes the bin packing problem and two-dimensional vector packing problem. This paper proves updated accuracy estimates and time complexity for several previously developed polynomial approximation algorithms for the two-bar charts packing problem and particular cases of the problem. We show the attainability of the estimates. Furthermore, we consider a problem of packing an unlimited number of bar charts belonging to k different types and propose a polynomial algorithm to solve the problem in case $k = \text{const}$. Illustr. 12, bibliogr. 26.

Keywords: bar chart, strip packing, NP-hard problem, a priori estimate, attainability.

References

1. A. I. Erzin, R. V. Plotnikov, A. P. Korobkin, G. E. Melidi, and S. A. Nazarenko, Optimal investment in the development of oil and gas field, in *Mathematical Optimization Theory and Operations Research* (Rev. Sel. Pap. 19th Int. Conf. MOTOR 2020, Novosibirsk, Russia, July 6–10, 2020) (Springer, Cham, 2020), pp. 336–349 (Commun. Comput. Inf. Sci., Vol. 1275).
2. A. I. Erzin, G. E. Melidi, S. A. Nazarenko, and R. V. Plotnikov, Two-bar charts packing problem, *Optim. Lett.* **15** (6), 1955–1971 (2021).
3. A. I. Erzin, G. E. Melidi, S. A. Nazarenko, and R. V. Plotnikov, A $3/2$ -approximation for big two-bar charts packing, *J. Comb. Optim.* **42** (1), 71–84 (2021).

4. **A. I. Erzin, G. E. Melidi, S. A. Nazarenko, and R. V. Plotnikov**, A posteriori analysis of the algorithms for two-bar charts packing problem, in *Advances in Optimization and Applications* (Rev. Sel. Pap. 12th Int. Conf. OPTIMA 2021, Petrovac, Montenegro, Sep. 27–Oct. 1, 2021) (Springer, Cham, 2021), pp. 201–216 (Commun. Comput. Inf. Sci., Vol. 1514.)
5. **A. I. Erzin, A. V. Kononov, G. E. Melidi, and S. A. Nazarenko**, A $4/3 \times \text{OPT} + 2/3$ approximation for big two-bar charts packing problem, *J. Math. Sci.* **269** (6), 813–823 (2023).
6. **A. I. Erzin, A. V. Kononov, S. A. Nazarenko, and K. I. Sharankhaev**, An $O(n \log n)$ -time algorithm for linearly ordered packing of 2-bar charts into $\text{OPT} + 1$ bins, in *Mathematical Optimization Theory and Operations Research* (Rev. Sel. Pap. 22th Int. Conf. MOTOR 2023, Yekaterinburg, Russia, July 2–8, 2023) (Springer, Cham, 2023), pp. 122–133 (Commun. Comput. Inf. Sci., Vol. 1881).
7. **M. Barkel and M. Delorme**, Arcflow formulations and constraint generation frameworks for the two bar charts packing problem, *INFORMS J. Comput.* **35** (2), 475–494 (2023).
8. **M. R. Garey and D. S. Johnson**, *Computers and Intractability. A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).
9. **V. V. Vazirani**, *Approximation Algorithms* (Springer, Heidelberg, 2001).
10. **D. Simchi-Levi**, New worst-case results for the bin-packing problem, *Nav. Res. Logist.* **41** (4), 579–585 (1994).
11. **G. Dósa**, The tight bound of first fit decreasing bin-packing algorithm is $\text{FFD}(I) \leq 11/9 \text{OPT}(I) + 6/9$, in *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies* (Rev. Sel. Pap. 1st Int. Symp., Hangzhou, China, Apr. 7–9, 2007) (Springer, Heidelberg, 2007), pp. 1–11 (Lect. Notes Comput. Sci., Vol. 4614).
12. **D. S. Johnson and M. R. Garey**, A $71/60$ theorem for bin packing, *J. Complex.* **1** (1), 65–106 (1985).
13. **M. Yue and L. Zhang**, A simple proof of the inequality $\text{MFFD}(L) \leq 71/60 \times \text{OPT}(L) + 1$ for the MFFD bin-packing algorithm, *Acta Math. Appl. Sin.* **11** (3), 318–330 (1995).
14. **W. Fernandez de la Vega and G. S. Lueker**, Bin packing can be solved within $1 + \varepsilon$ in linear time, *Combinatorica* **1**, 349–355 (1981).
15. **N. Karmarkar and R. M. Karp**, An efficient approximation scheme for the one dimensional bin-packing problem, in *23rd Annu. Symp. Foundations of Computer Science, Chicago, USA, Nov. 3–5, 1982* (IEEE, Piscataway, 1982), pp. 312–320.
16. **R. Hoberg and T. Rothvoss**, A logarithmic additive integrality gap for bin packing, in *Proc. 28th Annu. ACM-SIAM Symp. Discrete Algorithms, Barcelona, Spain, Jan. 16–19, 2017* (SIAM, Philadelphia, PA, 2017), pp. 2616–2625.
17. **H. Kellerer and V. Kotov**, An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing, *Oper. Res. Lett.* **31**, 35–41 (2003).

18. **H. I. Christensen, A. Khanb, S. Pokutta, and P. Tetali**, Approximation and online algorithms for multidimensional bin packing: A survey, *Comput. Sci. Rev.* **24**, 63–79 (2017).
19. **N. Bansal, M. Eliáš, and A. Khan**, Improved approximation for vector bin packing, in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms, Arlington, VA, USA, Jan. 10–12, 2016* (SIAM, Philadelphia, PA, 2016), pp. 1561–1579.
20. **G. J. Woeginger**, There is no asymptotic PTAS for two-dimensional vector packing, *Inf. Process. Lett.* **64** (6), 293–297 (1997).
21. **P. Brucker and S. Knust**, *Complex Scheduling* (Springer, Heidelberg, 2006).
22. **E. N. Goncharov**, A greedy heuristic approach for the resource-constrained project scheduling problem, *Stud. Inform. Univers.* **9** (3), 79–90 (2011).
23. **E. N. Goncharov**, A stochastic greedy algorithm for the resource-constrained project scheduling problem, *Diskretn. Anal. Issled. Oper.* **21** (3), 11–24 (2014) [Russian].
24. **S. Hartmann**, A self-adapting genetic algorithm for project scheduling under resource constraints, *Nav. Res. Logist.* **49**, 433–448 (2002).
25. **R. Kolisch and S. Hartmann**, Experimental investigation of heuristics for resource-constrained project scheduling: An update, *Eur. J. Oper. Res.* **174**, 23–37 (2006).
26. **A. I. Erzin and K. I. Sharankhaev**, Three-bar charts packing problem, *Commun. Comput. Inf. Sci.* **1739**, 61–75 (2023).

Stepan A. Nazarenko

Received December 15, 2023

Revised April 23, 2024

Accepted June 22, 2024

СПУСК С ЧЕРЕДУЮЩИМИСЯ ОКРЕСТНОСТЯМИ
ДЛЯ ПОИСКА РАДИУСА ПОРОГОВОЙ УСТОЙЧИВОСТИ
В ЗАДАЧЕ РАЗМЕЩЕНИЯ И ДИСКРИМИНАЦИОННОГО
ЦЕНООБРАЗОВАНИЯ

А. А. Панин^a, Д. А. Пискеева^b, А. В. Плясунов^c

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

E-mail: ^aaapanin1988@gmail.com,
^bd.piskeeva@g.nsu.ru, ^capljas@math.nsc.ru

Аннотация. Рассматривается новая проблема пороговой устойчивости при размещении предприятий и дискриминационном ценообразовании. В задаче размещения и ценообразования производитель принимает решение об открытии предприятий и назначении цен для каждого потребителя на каждом предприятии. Дискриминация в ценообразовании приводит к ситуации, когда каждый потребитель вынужден тратить максимум своих финансовых ресурсов, тем самым гарантируя максимальный доход производителю. В проблеме пороговой устойчивости финансовые ресурсы или бюджет каждого потребителя является параметром с известным ожидаемым значением. Цель — максимизировать отклонение параметров от ожидаемого значения при условии, что доход производителя не меньше заданного порога.

Для решения проблемы пороговой устойчивости предлагается алгоритм, основанный на спуске с чередующимися окрестностями (VND). Численное исследование алгоритма проводится на известных примерах и случайно сгенерированных. Исследуются различные способы построения стартового размещения и различные критерии сравнения векторов размещения предприятий. Табл. 3, ил. 6, библиогр. 12.

Ключевые слова: пороговая устойчивость, размещение и ценообразование, спуск с чередующимися окрестностями.

Введение

Пороговая устойчивость [5] является относительно новым направлением исследований в дискретной оптимизации и исследовании операций. В определённом смысле пороговая устойчивость является развитием робастной оптимизации [1]: оба подхода направлены на решение оптимизационных задач в условиях неточности входных данных. В робастной оптимизации задано множество сценариев, описывающих возможные и ожидаемые значения входных параметров примера оптимизационной задачи, а значение целевой функции опирается на худший сценарий для допустимого решения. В подходе пороговой устойчивости множество сценариев не задано, но известно ожидаемое значение целевой функции, определяемое как некоторый порог, ограничивающий это значение. Цель — найти такое решение, при котором значение целевой функции не преодолевает заданного порога на окрестности множества всевозможных сценариев максимального радиуса. Такой радиус называется *радиусом пороговой устойчивости*. Впервые подход пороговой устойчивости в дискретной оптимизации был предложен в работах [2, 3].

В данной работе исследуется пороговая устойчивость NP-трудной задачи размещения производства и ценообразования [9]. Производитель определяет, в каких местах и сколько предприятий ему нужно открыть и какие цены на однородную продукцию установить, чтобы максимизировать прибыль от обслуживания потребителей (потенциальных клиентов). В каждом возможном месте открытия (размещения) известна стоимость открытия предприятия. Каждый потребитель имеет ограниченный бюджет, который он готов потратить на покупку и транспортировку продукта в единичном экземпляре. Потребители действуют рационально, т. е. минимизируют свои затраты. Считаем, что транспортные затраты известны и неизменны, а информация о бюджетах неточна. В таком случае множество сценариев — это множество отклонений бюджетов от текущего ожидаемого значения.

Результат принятия решения в задачах размещения производства и ценообразования зависит от выбора политики назначения цен. Можно выделить три классических политики: равномерная, фабричная и дискриминационная [4]. При равномерной политике назначается единая цена на всю продукцию. При фабричной политике на каждом пункте обслуживания (предприятие, магазин) устанавливается своя цена. В данной работе ограничимся рассмотрением дискриминационной политики, когда на каждом предприятии для каждого потребителя назначается своя цена обслуживания.

Пороговая устойчивость для задач размещения производства и ценообразования исследовалась в работах [5–7]. В [6] исследовалась задача

с медианным ограничением на открытие предприятий и равномерной стратегией назначения цен. В [5] можно найти обзор работ, посвящённых проблемам размещения производства и ценообразования и пороговой устойчивости. В [7] впервые исследована пороговая устойчивость задач ценообразования с различными ценовыми политиками и задач конкурентного ценообразования.

В разд. 1 приводится математическая модель исследуемой проблемы. Приближённые алгоритмы для её решения предлагаются в разд. 2. В разд. 3 описывается численный эксперимент для сравнения разработанных алгоритмов с решателем Gurobi.

1. Математическая модель

Для описания математической модели введём следующие обозначения входных данных:

- I — конечное множество предприятий, предлагаемых к размещению или открытию;
- J — конечное множество потребителей;
- b_j — бюджет потребителя j ;
- c_{ij} — стоимость транспортировки товара от предприятия i к потребителю j ;
- f_i — стоимость открытия предприятия i ;
- V — желаемая прибыль производителя (порог).

Кроме того, введём следующие переменные:

- ρ — радиус пороговой устойчивости, отклонение от ожидаемых значений бюджетов потребителей;
- p_{ij} — цена товара на предприятии i для потребителя j ;
- $x_{ij} = \begin{cases} 1, & \text{если клиент } j \text{ обслуживается на предприятии } i, \\ 0 & \text{иначе;} \end{cases}$
- $y_i = \begin{cases} 1, & \text{если предприятие } i \text{ открыто,} \\ 0 & \text{иначе.} \end{cases}$

Используя данные обозначения, запишем задачу пороговой устойчивости для размещения и дискриминационного ценообразования в виде следующей модели квадратичного программирования:

$$\rho \rightarrow \max_{\rho, p, y, x}, \quad (1)$$

$$\sum_{i \in I} \sum_{j \in J} p_{ij} x_{ij} - \sum_{i \in I} f_i y_i \geq V, \quad (2)$$

$$x_{ij} \leq y_i, \quad i \in I, j \in J, \quad (3)$$

$$\sum_{i \in I} x_{ij} \leq 1, \quad j \in J, \quad (4)$$

$$\sum_{i \in I} (b_j - \rho - c_{ij} - p_{ij})x_{ij} \geq 0, \quad j \in J, \quad (5)$$

$$\sum_{i \in I} (c_{ij} + p_{ij})x_{ij} \leq c_{kj} + p_{kj}, \quad k \in I, j \in J, \quad (6)$$

$$x_{ij}, y_i \in \{0, 1\}, \quad \rho, p_{ij} \geq 0, \quad i \in I, j \in J. \quad (7)$$

В целевой функции (1) максимизируется радиус пороговой устойчивости. Пороговое ограничение (2) гарантирует, что прибыль производителя будет не меньше заданного порога V . Ограничение (3) запрещает обслуживать потребителей в закрытых (не размещённых) предприятиях. Условие (4) говорит о том, что потребители заинтересованы только в одном экземпляре продукта. Бюджетное ограничение (5) и условие (6) гарантируют, что каждый потребитель обслуживается на предприятии с минимальными суммарными затратами и совершает покупку только в том случае, если затраты не превышают бюджета.

Заметим, что мы не рассматриваем сценарии увеличения значения бюджетов потребителей и не проверяем допустимость решения при всех значениях радиуса, не превышающих максимального, в силу очевидности последнего. При увеличении значения бюджетов левая часть ограничения (5) не убывает.

2. Эвристические алгоритмы решения

В [6] рассматривалась задача пороговой устойчивости при равномерной стратегии ценообразования и медианном ограничении на размещение предприятий. Предложенный авторами алгоритм решения использует идеи подходов, представленных в [8, 9], для максимизации прибыли производителя в условиях фабричной политики назначения цен и медианного ограничения на размещение производства. В основе алгоритма из [6] лежит метод спуска с чередующимися окрестностями (VND). Для решения исследуемой в данной статье задачи мы предлагаем алгоритмы, являющиеся модификацией этого эвристического подхода.

В [6] для поиска вектора размещения использовалась окрестность k -swap, соответствующая всевозможным вариантам одновременного закрытия k предприятий и открытия k новых. Другими словами, k -swap(y) содержит векторы с таким же весом Хэмминга, что и y , но находящиеся на расстоянии Хэмминга $2k$ от него.

В постановке, предлагаемой в настоящей работе, таких окрестностных структур недостаточно. Невозможно предугадать заранее оптимальное число открытых предприятий, поэтому рассмотрим ещё одну классическую окрестность k -flip [9], соответствующую тем размещениям, при которых либо k предприятий закрываются, либо открываются k новых.

Другими словами, окрестность $k\text{-flip}(y)$ содержит размещения, находящиеся на расстоянии Хэмминга k от y , вес Хэмминга которых отличается от веса y на k . Для простоты: $\text{swap} = 1\text{-swap}$, $\text{flip} = 1\text{-flip}$.

Определим также процедуру встряски $k\text{-shake}(y)$, которая понадобится для описания VND-алгоритмов. Эта процедура состоит из последовательного просмотра окрестностей $2\text{-swap}(y) \cup 2\text{-flip}(y)$, $\dots$, $k\text{-swap}(y) \cup k\text{-flip}(y)$ до первого улучшения размещения y .

Общая схема разработанных алгоритмов представлена алгоритмом 1. Приведённая схема VND-алгоритмов основана на идеях VNS- и VND-метаэвристик [10]. В нашей интерпретации окрестности строятся на части решения, соответствующей размещению производства. Такой подход связан с простотой задачи при фиксированном размещении. Задача дискриминационного ценообразования полиномиально разрешима [11].

Алгоритм 1. Схема алгоритмов VND

Вход: k — параметр процедуры встряски $k\text{-shake}(y)$.

Выход: радиус пороговой устойчивости ρ , вектор размещения y , матрица цен p , матрица назначения клиентов x .

- 1: задать начальный вектор размещения y ;
 - 2: применить локальный спуск по окрестности $\text{swap}(y) \cup \text{flip}(y)$ и найти локальный оптимум y^* ;
 - 3: $y := k\text{-shake}(y^*)$;
 - 4: **if** $y = y^*$ **then** **СТОП**; ▷ Улучшения не произошло.
 - 5: **else goto** 2.
-

Для сравнения векторов размещения предприятий требуется сперва определить цены, назначения клиентов и радиус. Дискриминационное ценообразование позволяет перейти от матрицы цен к вектору, каждый элемент которого соответствует цене обслуживания клиента на ближайшем предприятии производителя. Алгоритм поиска радиуса ρ и матриц p , x приведён ниже (алгоритм 2).

Для вычисления матрицы цен достаточно для любого клиента j положить цену обслуживания на всех предприятиях равной p_j . При повторении шага 6 алгоритма 2 число обслуживаемых клиентов сокращается. Следовательно, трудоёмкость алгоритма равна $O(|I||J| + |J|^2)$ и верно

Утверждение 1. При фиксированном размещении $y \in \{0, 1\}^{|I|}$ задача (1)–(7) разрешима за $O(|I||J| + |J|^2)$ операций.

Ограничимся рассмотрением двух процедур поиска стартового размещения: случайная генерация и жадный алгоритм. В обеих процедурах сперва задаётся число открываемых предприятий. Схема жадного алгоритма отражена в алгоритме 3.

Алгоритм 2. Вычисление цен и радиуса пороговой устойчивости**Вход:** y — вектор размещения предприятий.**Выход:** радиус пороговой устойчивости ρ , вектор цен p , матрица назначения клиентов x .

- 1: $I(y) := \{i \mid y_i = 1\}$; $\triangleright$ Множество индексов открытых предприятий.
- 2: **for** $j \in J$ **do** $p_j := \max \left\{ 0, \max_{i \in I(y)} \{b_j - c_{ij}\} \right\}$; $\triangleright$ Цена обслуживания клиента j на ближайшем открытом предприятии.
- 3: $c(p) := |\{j \in J \mid p_j \neq 0\}|$; $\triangleright$ Число обслуживаемых клиентов.
- 4: $d(p) := \sum_{j \in J} p_j - \left(V + \sum_{i \in I} y_i f_i \right)$; $\triangleright$ Излишек прибыли, доступный для перераспределения.
- 5: $\rho := 0$;
- 6: **while** $d(p) > 0$ **do**
- 7: $\rho := \rho + \frac{d(p)}{c(p)}$;
- 8: **for** $j \in J$ **do** $p_j := \max \left\{ 0, p_j - \frac{d(p)}{c(p)} \right\}$;
- 9: $c(p) := |\{j \in J \mid p_j \neq 0\}|$;
- 10: $d(p) := \sum_{j \in J} p_j - \left(V + \sum_{i \in I} y_i f_i \right)$;
- 11: **for** $i \in I, j \in J$ **do** $\triangleright$ Матрица назначения клиентов x .
- 12: **if** $p_j > 0$ и предприятие i ближайшее к потребителю j
 then $x_{ij} := 1$;
- 13: **else** $x_{ij} := 0$.

Алгоритм 3. Жадный алгоритм поиска стартового размещения y **Вход:** r — число открываемых предприятий.**Выход:** вектор размещения предприятий y .

- 1: **for** $i \in I$ **do** $v_i := \sum_{j \in J} (b_j - \min_{i \in I} c_{ij})$; $\triangleright$ Максимальный доход предприятия i от обслуживания всех клиентов.
- 2: упорядочить предприятия по убыванию дохода;
- 3: открыть r наиболее доходных предприятий.

В [6] рассмотрены два критерия для сравнения векторов размещения предприятий: 1) максимизация прибыли производителя; 2) максимизация радиуса пороговой устойчивости. Использование первого критерия приводит к алгоритму решения исходной задачи с последующим вычислением (ρ, p, x) . При втором критерии (ρ, p, x) определяются для каждого вектора размещения. Так как на каждой итерации алгоритма спуска

с чередующимися окрестностями решение не ухудшается, для каждого рекордного радиуса ρ можем уменьшить на его значение все бюджеты клиентов, тем самым упростив задачу.

3. Численный эксперимент

Для проведения вычислительного эксперимента использованы примеры из библиотеки «Дискретные задачи размещения» [12], а также случайно сгенерированные примеры из статьи [6]. Все примеры разбиты на шесть групп:

- 1) FLPr-40-100, 10 примеров;
- 2) FLPr-100-100, 10 примеров;
- 3) gen-20-20, 2 примера;
- 4) gen-40-40, 2 примера;
- 5) gen-60-60, 2 примера;
- 6) gen-100-100, 3 примера.

Префикс «FLPr» означает примеры из библиотеки, «gen» — случайно сгенерированные примеры. Первое число в названии группы — число мест для размещения предприятий, а второе — число потребителей. Все примеры содержат информацию об исходных бюджетах и транспортных затратах.

В качестве порога рассматривались три уровня (значения): 30, 50, 70% от наибольшей прибыли производителя, найденной решателем Gurobi для исходной задачи размещения и ценообразования. Стоимости открытия предприятий, обозначенные на рисунках charge, брались одинаковыми и равными 5, 15, 30, 50, 100, 300. Так как в зависимости от числа открываемых предприятий r в стартовом размещении y разработанный алгоритм спуска с чередующимися окрестностями может приводить

Таблица 1

Среднее относительное отклонение (%) и время счёта (с)
для алгоритмов VND и Gurobi

Алгоритм	VND								Gurobi
Критерий	px		rho		px-rho		rho-px		
Примеры	gap	time	gap	time	gap	time	gap	time	time
gen-20-20	0,0295	0,004	0,0256	0,004	0,0256	0,004	0,0256	0,004	1,76
gen-40-40	0,0284	0,027	0,0255	0,029	0,0255	0,039	0,0255	0,040	113,73
gen-60-60	0,0761	0,088	0,0632	0,103	0,0757	0,139	0,0632	0,141	845,81
FLPr-40-100	2,4275	0,120	0,1396	0,120	0,1004	0,189	0,1388	0,183	25,63
gen-100-100	−0,0907	0,688	−0,0928	0,733	−0,0929	1,007	−0,0937	1,020	8203,2
FLPr-100-100	2,9078	2,929	0,1505	2,252	0,1075	3,980	0,1503	3,496	157,34
Все примеры	0,8964	0,64	0,0519	0,54	0,0403	0,89	0,0516	0,81	1557,9

Таблица 2

**Среднее относительное отклонение (%)
по группе тестов FLPr-40-100**

	r	Уровень порога V от наибольшей прибыли											
		30%				50%				70%			
		Стоимость открытия предприятий											
		5	50	100	300	5	50	100	300	5	50	100	300
px	0	1,21	3,06	3,49	5,69	0,77	2,08	1,87	3,58	0,64	1,30	0,92	1,60
		1,21	3,06	3,49	5,69	0,77	2,08	1,87	3,58	0,64	1,30	0,92	1,60
	5	1,23	2,98	3,51	8,93	0,83	2,07	1,92	6,09	0,64	1,29	0,86	3,56
		1,22	3,06	4,36	8,15	0,79	2,08	2,57	5,61	0,67	1,30	1,38	3,40
	10	1,27	3,07	3,81	8,34	0,85	2,15	2,20	5,56	0,67	1,41	1,26	3,08
		1,24	3,27	4,29	7,64	0,83	2,28	2,53	4,99	0,71	1,46	1,45	2,52
	15	1,31	3,29	4,06	7,55	0,91	2,36	2,49	4,89	0,77	1,57	1,52	2,22
		1,27	3,43	4,16	8,17	0,86	2,35	2,41	5,34	0,76	1,57	1,30	2,81
	20	1,28	3,28	4,39	7,55	0,87	2,35	2,69	4,89	0,70	1,56	1,69	2,22
		1,24	3,53	4,19	8,17	0,84	2,44	2,55	5,34	0,75	1,69	1,52	2,81
	25	1,30	3,18	4,19	7,47	0,90	2,26	2,55	5,10	0,75	1,44	1,52	2,89
		1,25	3,53	4,33	7,68	0,85	2,44	2,69	4,92	0,77	1,69	1,70	2,56
	opt	1,28	2,98	3,51	5,42	0,87	2,15	1,92	3,27	0,74	1,41	1,08	1,04
		1,27	3,06	4,39	5,69	0,84	2,08	2,57	4,11	0,75	1,46	1,12	2,97
rho	0	0	0	0,26	1,44	0	0	0,02	0,07	0	0,03	0,02	0,91
		0	0	0,26	1,44	0	0	0,02	0,07	0	0,03	0,02	0,91
	5	0	0,01	0	0,40	0	0	0	0,56	0	0	0	0,84
		0	0	0	0,59	0	0,06	0,01	0,99	0	0,03	0,45	1,25
	10	0	0,01	0	0,30	0	0,04	0,01	1,42	0	0	0	0,19
		0	0,01	0	0,30	0	0,07	0,02	1,17	0	0,03	0,48	0,19
	15	0	0,01	0	0,40	0	0	0,01	1,42	0	0	0	0,44
		0	0,01	0	0,40	0	0,03	0,02	1,17	0	0,03	0,17	0,25
	20	0	0,01	0	0,40	0	0	0,01	0,56	0	0	0	1,87
		0	0	0	0,30	0	0	0,01	0,30	0	0,03	0,48	1,16
	25	0	0	0	0,40	0	0,04	0,01	0,56	0	0	0	1,87
		0	0	0	0,30	0	0	0,01	0,44	0	0,03	0	1,16
	opt	0	0,01	0	0,74	0	0	0	0,38	0	0	0,03	1,81
		0	0	0	0,50	0	0,04	0,01	0,00	0	0,03	0,31	1,46

к разным результатам, эксперимент проводился на 7 значениях: 0, 5, 10, 15, 20, 25 и opt — число открытых предприятий в решении, найденном решателем Gurobi. Для применения Gurobi математическая модель линейаризована, как в [6]. Параметр k процедуры встряски shake равен 2.

VND-алгоритм применён с критериями максимизации прибыли (px) и максимизации радиуса пороговой устойчивости (rho). Вместе с тем исследовано чередование этих критериев на последовательных итерациях алгоритма (одна итерация включает локальный спуск и встряску). Группа запусков с критерием прибыли на первой итерации обозначена через px-rho, а с критерием максимизации радиуса — через rho-px.

Таблица 2 (продолжение)

	r	Уровень порога V от наибольшей прибыли											
		30%				50%				70%			
		Стоимость открытия предприятий											
		5	50	100	300	5	50	100	300	5	50	100	300
px-rho	0	0	0,02	0	0,40	0	0,03	0,01	0,44	0	0,03	0,02	0,24
		0	0,02	0	0,40	0	0,03	0,01	0,44	0	0,03	0,02	0,24
	5	0	0,01	0	0,40	0	0	0,01	0,56	0	0,03	0,02	0,24
		0	0,02	0	0,30	0	0,01	0	1,12	0	0,03	0,31	1,25
	10	0	0,03	0	0,40	0	0,04	0,01	0,85	0	0,03	0,03	0,49
		0	0,02	0	0,40	0	0,01	0,01	0,41	0	0,03	0,18	0,25
	15	0	0	0	0,40	0	0,01	0,01	0,85	0	0	0,02	0,24
		0	0,02	0	0,40	0	0,01	0,01	0,18	0	0,03	0,17	0,24
	20	0	0,02	0	0,40	0	0	0,01	0,30	0	0,03	0,18	0,49
		0	0,02	0	0,40	0	0,03	0,01	0,18	0	0,03	0,18	0,24
	25	0	0,02	0	0,40	0	0,04	0	0,30	0	0,03	0,18	0,49
		0	0,02	0	0,40	0	0,04	0	0,44	0	0,03	0,18	0,24
	opt	0	0,03	0	0,21	0	0,04	0,02	1,42	0	0	0,03	0,00
0		0,02	0	0,90	0	0	0	0,87	0	0,03	0	1,25	
rho-px	0	0	0	0,26	1,44	0	0	0,02	0,07	0	0,03	0,02	0,91
		0	0	0,26	1,44	0	0	0,02	0,07	0	0,03	0,02	0,91
	5	0	0,01	0	0,40	0	0	0	0,56	0	0	0	0,44
		0	0	0	0,59	0	0,03	0,01	0,99	0	0,03	0,45	1,25
	10	0	0,01	0	0,30	0	0,04	0,01	1,42	0	0	0	0,19
		0	0,01	0	0,30	0	0,04	0,02	1,17	0	0,03	0,48	0,19
	15	0	0,01	0	0,40	0	0	0,01	1,42	0	0	0	0,44
		0	0,01	0	0,40	0	0,03	0,02	1,17	0	0,03	0,17	0,25
	20	0	0,01	0	0,40	0	0	0,01	0,56	0	0	0	1,87
		0	0	0	0,30	0	0	0,01	0,30	0	0,03	0,48	1,16
	25	0	0	0	0,40	0	0,04	0,01	0,56	0	0	0	1,87
		0	0	0	0,30	0	0	0,01	0,44	0	0,03	0	1,16
	opt	0	0,01	0	0,74	0	0	0	0,38	0	0	0,03	1,81
0		0	0	0,50	0	0,04	0,01	0,00	0	0,03	0,31	1,46	

В табл. 1 приведены средние результаты по группам примеров для всех значений порога, стоимости открытия и всех стартовых векторов размещения предприятий, построенных жадным алгоритмом. В частности, приведено относительное отклонение $\text{gap} = \frac{\rho_{\text{gurobi}} - \rho_{\text{VND}}}{\rho_{\text{gurobi}}}$ в процентах и время вычисления time в секундах. Здесь ρ_{gurobi} — радиус пороговой устойчивости, найденный решателем Gurobi, а ρ_{VND} — радиус, найденный алгоритмом VND. Время работы решателя Gurobi ограничивалось 10800 секундами. В группе примеров gen-100-100 он не смог достичь оптимального решения за это время. Отрицательное относительное отклонение говорит о том, что Gurobi за отведённое время нашёл решение хуже, чем VND-алгоритм.

Таблица 3

**Среднее относительное отклонение (%)
по группе тестов FLPr-100-100**

	r	Уровень порога V от наибольшей прибыли											
		30%				50%				70%			
		Стоимость открытия предприятий											
		5	50	100	300	5	50	100	300	5	50	100	300
px	0	2,02	2,96	4,97	7,84	1,36	1,77	3,23	4,35	1,01	1,32	3,16	2,54
		2,02	2,96	4,97	7,84	1,36	1,77	3,23	4,35	1,01	1,32	3,16	2,54
	5	2,04	2,86	4,36	10,01	1,46	1,78	2,71	6,66	1,11	1,56	2,51	5,17
		1,99	2,93	4,52	7,84	1,35	1,74	2,91	4,35	1,00	1,30	2,93	2,54
	10	2,03	3,30	5,39	9,36	1,47	1,98	3,58	6,06	1,15	1,56	3,05	4,64
		2,02	3,03	5,26	8,57	1,43	1,76	3,53	5,11	1,10	1,36	3,40	3,64
	15	2,11	3,29	5,62	9,36	1,53	1,99	3,71	6,06	1,19	1,67	3,03	4,64
		2,02	2,89	5,11	8,57	1,44	1,60	3,42	5,11	1,11	1,20	3,01	3,64
	20	2,07	3,56	5,29	7,84	1,46	2,31	3,55	4,35	1,10	1,97	3,39	2,54
		2,03	3,38	5,27	8,57	1,44	2,06	3,42	5,11	1,07	1,68	3,00	3,64
	25	2,02	3,46	5,21	7,84	1,40	2,28	3,52	4,35	1,03	1,98	2,98	2,54
		2,01	3,40	4,89	8,57	1,42	2,00	3,18	5,11	1,08	1,65	3,02	3,64
	opt	2,05	3,02	4,33	6,76	1,45	1,59	2,47	3,30	1,07	1,27	2,30	1,02
		2,00	2,97	4,34	6,28	1,41	1,54	2,68	3,25	1,07	1,36	2,10	1,90
rho	0	0,01	0,10	0,14	0,43	0	0,16	0,31	1,55	0,01	0,33	0,85	0,90
		0,01	0,10	0,14	0,43	0	0,16	0,31	1,55	0,01	0,33	0,85	0,90
	5	0,01	0,09	0,16	0,05	0,01	0,28	0,18	0,50	0,03	0,16	0,89	1,30
		0	0,24	0,21	0,05	0	0,22	0,06	0,26	0,01	0,18	1,19	0,05
	10	0,01	0,07	0,06	0,05	0	0,04	0,19	0,57	0,02	0,25	0,42	1,08
		0	0,16	0,17	0,77	0	0	0,21	0,57	0,01	0,02	0,54	1,18
	15	0,01	0,05	0,05	0,05	0,01	0,22	0,18	0,57	0,02	0,06	0,49	1,76
		0	0,08	0,04	0,17	0,01	0	0,11	0,57	0,01	0,07	0,68	1,15
	20	0,01	0,07	0,05	0,05	0	0,16	0,13	0,57	0,02	0,17	0,61	1,15
		0,01	0,04	0,07	0,05	0,01	0,06	0,04	0,13	0,01	0,07	0,28	0,00
	25	0,01	0,05	0,05	0,77	0	0,06	0,19	0,77	0,02	0,40	0,65	1,15
		0,01	0,04	0,07	0,05	0,01	0,06	0,20	0,13	0	0,07	0,69	0,00
	opt	0,01	0,04	0,03	0,00	0,01	0,10	0,38	0,00	0,01	0,26	0,87	0,91
		0,01	0,11	0	0,00	0,01	0,04	0,06	0,00	0	0,49	0,40	0,46

Как видно из табл. 1, VND-алгоритм практически всегда находит оптимальное или близкое к оптимальному решение. Относительное отклонение превосходит 0,15% только для критерия максимизации прибыли и только на примерах FLPr, на которых среднее значение гар превышает 2,4%. При этом время работы алгоритма VND на 2–3 порядка меньше времени работы Gurobi.

Критерий максимизации прибыли существенно проигрывает критерию максимизации радиуса пороговой устойчивости на примерах FLPr. Причина кроется в том, что в исходной задаче размещения производства и ценообразования оптимальные или близкие к оптимальным решения

Таблица 3 (продолжение)

	r	Уровень порога V от наибольшей прибыли											
		30%				50%				70%			
		Стоимость открытия предприятий											
		5	50	100	300	5	50	100	300	5	50	100	300
px-rho	0	0,01	0,13	0,27	0,05	0	0	0,24	0,64	0	0,02	0,49	0,05
		0,01	0,13	0,27	0,05	0	0	0,24	0,64	0	0,02	0,49	0,05
	5	0,01	0,05	0,18	0,77	0,01	0,10	0,23	0,77	0,01	0,04	0,50	0,05
		0	0,01	0,10	0,05	0,01	0,10	0,15	0,77	0	0,17	0,80	0,05
	10	0	0,14	0,10	0,05	0,01	0	0,08	0,63	0,02	0,09	0,13	0,05
		0	0,06	0,18	0,05	0,01	0,04	0,18	0,63	0,01	0,02	0,43	0,05
	15	0,01	0,17	0,10	0,05	0,01	0,10	0,09	0,57	0,02	0,04	0,14	0,05
		0	0,01	0,08	0,05	0,01	0	0,18	0,64	0,01	0,09	0,15	0,05
	20	0,01	0,09	0,10	0,05	0,01	0,10	0,20	0,64	0,02	0,02	0,71	1,20
		0	0,19	0,08	0,05	0,01	0,06	0,03	0,77	0,01	0,09	0,18	0,05
	25	0,01	0,19	0,30	0,05	0,01	0	0,24	0,77	0,02	0,04	0,77	0,05
		0	0,19	0,28	0,05	0,01	0,12	0,20	0,77	0	0,12	0,36	0,05
	opt	0,01	0	0,11	0,05	0,01	0,04	0,23	0,26	0,01	0,09	0,75	0,05
		0	0,01	0,05	0,05	0,01	0,04	0,15	0,20	0	0,02	0,29	0,05
rho-px	0	0,01	0,10	0,14	0,43	0	0,16	0,31	1,55	0,01	0,33	0,85	0,90
		0,01	0,10	0,14	0,43	0	0,16	0,31	1,55	0,01	0,33	0,85	0,90
	5	0,01	0,09	0,16	0,05	0,01	0,28	0,18	0,50	0,03	0,16	0,89	1,30
		0	0,24	0,21	0,05	0	0,22	0,06	0,26	0,01	0,18	1,19	0,05
	10	0,01	0,07	0,06	0,05	0	0,04	0,19	0,57	0,02	0,25	0,42	1,08
		0	0,16	0,17	0,77	0	0	0,21	0,57	0,01	0,02	0,54	1,18
	15	0,01	0,05	0,05	0,05	0,01	0,22	0,18	0,57	0,02	0,06	0,49	1,76
		0	0,08	0,04	0,17	0,01	0	0,11	0,57	0,01	0,07	0,68	1,15
	20	0,01	0,07	0,05	0,05	0	0,16	0,13	0,57	0,02	0,17	0,61	1,15
		0,01	0,03	0,07	0,05	0,01	0,06	0,04	0,13	0,01	0,07	0,28	0,00
	25	0,01	0,05	0,05	0,77	0	0,06	0,19	0,77	0,02	0,40	0,65	1,15
		0,01	0,04	0,07	0,05	0,01	0,06	0,20	0,13	0	0,07	0,69	0,00
	opt	0,01	0,04	0,03	0,00	0,01	0,10	0,38	0,00	0,01	0,26	0,87	0,91
		0,01	0,11	0	0,00	0,01	0,04	0,06	0,00	0	0,02	0,40	0,46

могут сильно отличаться по числу открытых предприятий и числу обслуживаемых потребителей. При высокой стоимости открытия предприятий условно лишние предприятия могут заметно ограничивать допустимые вариации бюджетов и цен, что мы и наблюдаем. Однако, если применить идею чередования критериев и, используя критерий ρ , перестроить решение, полученное критерием px , можно достигнуть наилучших результатов, что видно из табл. 1.

В табл. 2 и 3 представлены средние результаты по группам тестов FLPr отдельно для каждого порога, стартового размещения и каждой стоимости открытия, кроме 15 и 30 — при этих значениях поведение алгоритмов почти такое же, как при стоимости открытия 5. Для каждого значения r получены два набора данных: в первой строке стартовое

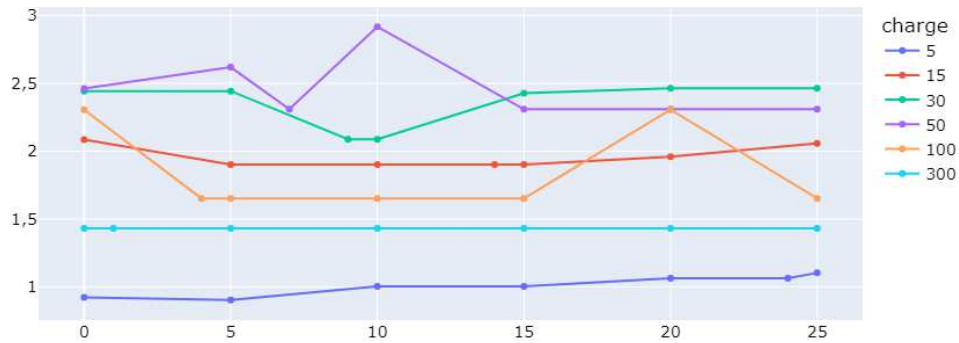


Рис. 1. Абсолютное отклонение от числа открытых предприятий в стартовом размещении на наборе данных FLPr-100-100-09 ($V = 30\%$, критерий px)

размещение предприятий y выбрано случайно, а во второй — жадным способом. Для критерия px характерен почти монотонный рост относительного отклонения с ростом стоимости открытия. Для других критериев картина совершенно другая. Отметим также, что выбор стартового размещения жадным или случайным образом практически не влияет на итоговый результат.

На рис. 1 для критерия px и различных стоимостей открытия предприятий представлены графики зависимости абсолютного отклонения от числа открытых жадной стратегией предприятий в стартовом размещении с уровнем порога 30% . Такая картина встречается крайне редко, лишь на некоторых примерах.

Часто возникает ситуация, когда при r , равном оптимальному или близкому к оптимальному числу предприятий, алгоритм VND спускается в далеко не самый лучший локальный оптимум. Если r сильно больше

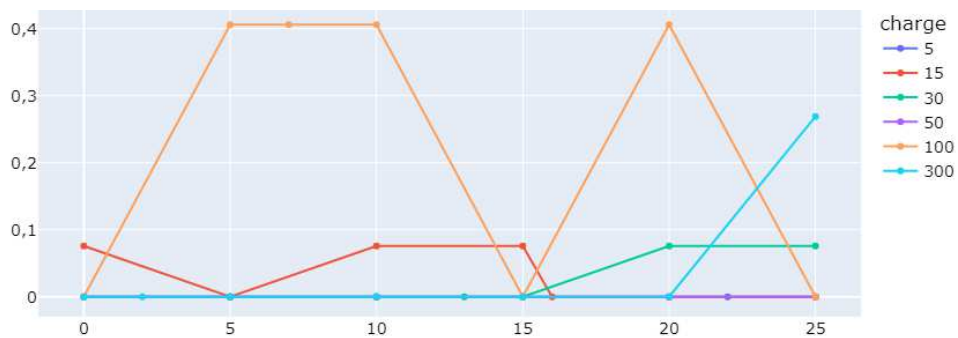


Рис. 2. Абсолютное отклонение от числа открытых предприятий в стартовом размещении на наборе данных FLPr-40-100-05 ($V = 70\%$, критерий ρ)

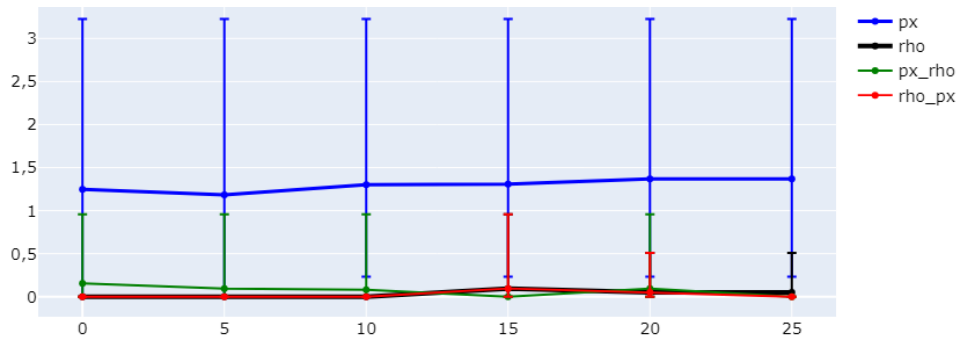


Рис. 3. Относительное отклонение от числа открытых предприятий в стартовом размещении на группе тестов FLPr-40-100 ($V = 70\%$, charge = 30)

или сильно меньше этого числа, то алгоритм спускается в оптимальное решение, как на рис. 2 для charge = 100, где оптимальное число предприятий равно 7. Для charge = 15 оптимальное число предприятий равно 16. При $r = 15$ алгоритм не находит оптимального решения, но если откроем ещё одно предприятие, то придём к оптимуму (рис. 2). Странность результатов заключается в том, что стартовое размещение определяется жадным образом, а значит, при $r \in \{15, 16\}$ первые 15 предприятий одни и те же. Такая картина характерна для всех критериев.

На рис. 3 и 4 для различных критериев представлены средние, максимальные и минимальные значения относительного отклонения по всей группе тестовых примеров FLPr-40-100 в зависимости от числа открытых жадным способом предприятий в стартовом размещении. Отличие в том, что на рис. 4 для наглядности исключены графики для критерия px . Заметно, что критерий px показывает себя хуже остальных, что характерно

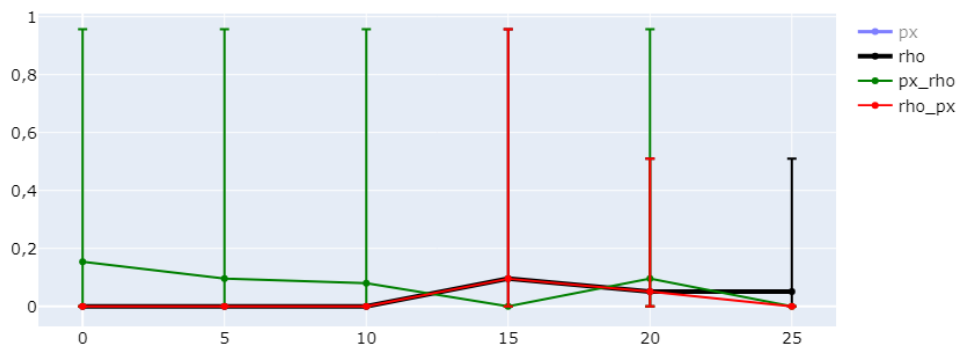


Рис. 4. Относительное отклонение от числа открытых предприятий в стартовом размещении на группе тестов FLPr-40-100 ($V = 70\%$, charge = 30)

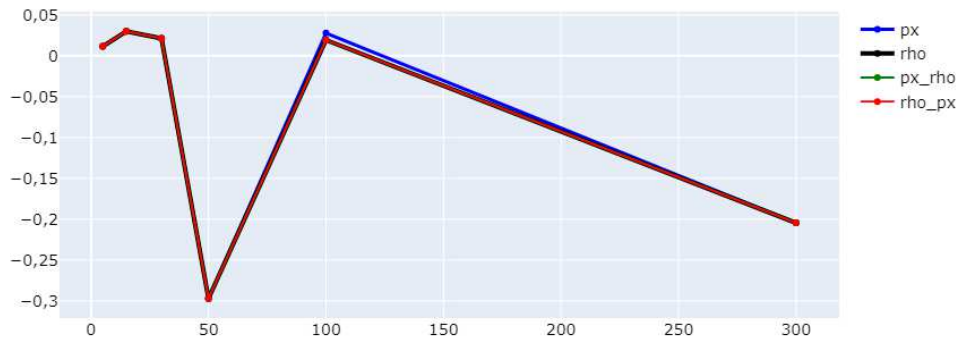


Рис. 5. Абсолютное отклонение от стоимости открытия предприятия на наборе данных gen-100-100-3 с уровнем порога 70%

для всей группы примеров FLPr. Эффективность каждого из оставшихся критериев зависит от числа предприятий в стартовом размещении.

Ещё одна интересная закономерность представлена на рис. 5. Практически на всех примерах групп gen наблюдается одна и та же картина зависимости абсолютного и относительного отклонений от стоимости открытия предприятий, почти идентичная для всех критериев. Для примеров FLPr такая зависимость лишь отчасти прослеживается для всех критериев, кроме px . Нередко встречается картина зависимости, как на рис. 6. Критерий px на этих примерах ведёт себя крайне непредсказуемо, и в сравнении с другими критериями часто наблюдается прямо противоположная картина.

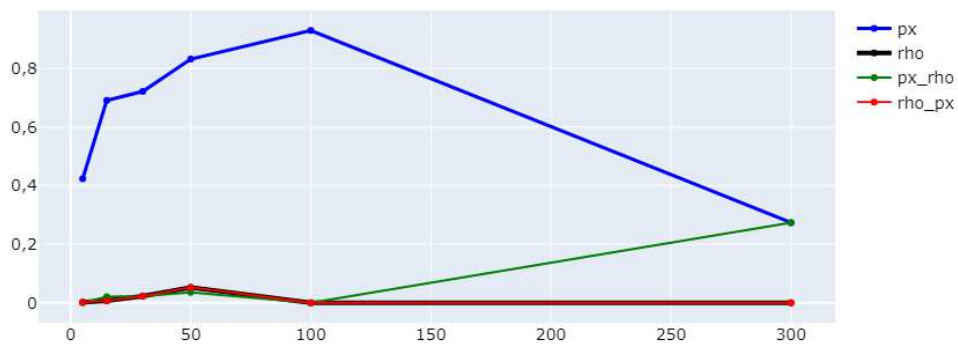


Рис. 6. Абсолютное отклонение от стоимости открытия предприятия на наборе данных FLPr-100-100-09 с уровнем порога 50%

Заключение

В работе рассмотрена проблема пороговой устойчивости при размещении производства и дискриминационном ценообразовании. Предложенный для её решения алгоритм, основанный на спуске с чередующимися окрестностями, даже в своей худшей реализации показывает результаты с относительным отклонением не более 3% в среднем для любой группы тестов, в лучшей реализации — 0,15%. Наибольшее относительное отклонение на отдельном примере достигает 19%. Причиной этому служит относительно небольшое значение оптимального радиуса пороговой устойчивости для данного примера.

Для задач пороговой устойчивости впервые была исследована идея чередования критериев максимизации радиуса пороговой устойчивости и оптимизации целевой функции исходной задачи в итерационных эвристических алгоритмах, позволяющая получать рекордные результаты.

Финансирование работы

Исследование выполнено за счёт Российского научного фонда (проект № 23–21–00424). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Ben-Tal A., Nemirovski A. Robust optimization — Methodology and applications // Math. Program. Ser. B. 2002. V. 92, No. 3. P. 453–480.
2. Carrizosa E., Nickel S. Robust facility location // Math. Methods Oper. Res. 2003. V. 58, No. 2. P. 331–349.
3. Carrizosa E., Ushakov A., Vasilyev I. Threshold robustness in discrete facility location problems: A bi-objective approach // Optim. Lett. 2015. V. 9, No. 7. P. 1297–1314.
4. Hanjoul P., Hansen P., Peeters D., Thisse J.-F. Uncapacitated plant location under alternative spatial price policies // Manage. Sci. 1990. V. 36, No. 1. P. 41–57. DOI: 10.1287/mnsc.36.1.41.
5. 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, No. 6. P. 1295–1315. DOI: 10.1007/s11590-022-01924-3.
6. Vodyan M. E., Panin A. A., Plyasunov A. V. Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem // Proc. 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.

7. **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.
8. **Diakova Z. S., Kochetov Yu. A.** A double VNS heuristic for the facility location and pricing problem // Electron. Notes Discrete Math. 2012. V. 39. P. 29–34. DOI: 10.1016/j.endm.2012.10.005.
9. **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.
10. **Кочетов Ю. А., Младенович Н., Хансен П.** Локальный поиск с чередующимися окрестностями // Дискрет. анализ и исслед. операций. Сер. 2. 2003. Т. 10, № 1. С. 11–43.
11. **Плясунов А. В., Панин А. А.** Задача ценообразования. Часть 2. Вычислительная сложность // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 6. С. 56–71.
12. **Береснев В. Л., Кочетов Ю. А., Пащенко М. Г. [и др.].** Дискретные задачи размещения. Библиотека тестовых задач. Новосибирск: ИМ СО РАН, 2021. URL: math.nsc.ru/AP/benchmarks (дата обращения: 1.11.2024).

Панин Артём Александрович

Пискеева Дарья Алексеевна

Плясунов Александр Владимирович

Статья поступила

8 мая 2024 г.

После доработки —

15 мая 2024 г.

Принята к публикации

22 июня 2024 г.

VARIABLE NEIGHBORHOOD DESCENT FOR FINDING
THE THRESHOLD STABILITY RADIUS IN THE FACILITY
LOCATION AND DISCRIMINATORY PRICING PROBLEMA. A. Panin^a, D. A. Piskeeva^b, and A. V. Plyasunov^c

Sobolev Institute of Mathematics,
4 Acad. Koptug Avenue, 630090 Novosibirsk, Russia
E-mail: ^aaapanin1988@gmail.com,
^bd.piskeeva@g.nsu.ru, ^capljas@math.nsc.ru

Abstract. A new threshold stability problem in the context of facility location and discriminatory pricing is considered. In the statement of facility location and pricing problem, the company decides to open facilities and assign prices to each customer at each facility. The implementation of discriminatory pricing leads to a scenario where each customer is compelled to expend the maximum amount of their available financial resources, thereby ensuring the maximum revenue for the company. In the threshold stability problem, the available financial resources or budget of each consumer is a parameter with a known expected value. The objective is to maximize the deviation of the parameters from the expected value, provided that the company's income remains above a given threshold.

An algorithm based on variable neighborhood descent (VND) is proposed to solve the threshold stability problem. Numerical investigation of the algorithm is carried out on known instances and randomly generated ones. Various ways of constructing the starting facility location and different criteria for comparing the location vectors are analyzed. Tab. 3, illustr. 6, bibliogr. 12.

Keywords: threshold stability, location and pricing, variable neighborhood descent.

References

1. A. Ben-Tal and A. Nemirovski, Robust optimization — Methodology and applications, *Math. Program., Ser. B*, **92** (3), 453–480 (2002).

English transl.: *Journal of Applied and Industrial Mathematics* **18** (4), 789–800 (2024), DOI: 10.1134/S1990478924040136.

2. **E. Carrizosa** and **S. Nickel**, Robust facility location, *Math. Methods Oper. Res.* **58** (2), 331–349 (2003).
3. **E. Carrizosa**, **A. Ushakov**, and **I. Vasilyev**, Threshold robustness in discrete facility location problems: A bi-objective approach, *Optim. Lett.* **9** (7), 1297–1314 (2015).
4. **P. Hanjoul**, **P. Hansen**, **D. Peeters**, and **J.-F. Thisse**, Uncapacitated plant location under alternative spatial price policies, *Manage. Sci.* **36** (1), 41–57 (1990), DOI: 10.1287/mnsc.36.1.41.
5. **A. A. Panin** and **A. V. Plyasunov**, The multilevel facility location and pricing problems: The computational complexity and the stability analysis, *Optim. Lett.* **17** (6), 1295–1315 (2023), DOI: 10.1007/s11590-022-01924-3.
6. **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 *Proc. 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.
7. **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.
8. **Z. S. Diakova** and **Yu. A. Kochetov**, A double VNS heuristic for the facility location and pricing problem, *Electron. Notes Discrete Math.* **39**, 29–34 (2012), DOI: 10.1016/j.endm.2012.10.005.
9. **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)].
10. **Yu. A. Kochetov**, **N. Mladenović**, and **P. Hansen**, Local search with alternating neighborhoods, *Diskretn. Anal. Issled. Oper., Ser. 2*, **10** (1), 11–43 (2003) [Russian].
11. **A. V. Plyasunov** and **A. A. Panin**, The pricing problem. Part II: Computational complexity, *Diskretn. Anal. Issled. Oper.* **19** (6), 56–71 (2012) [Russian] [*J. Appl. Ind. Math.* **7** (3), 420–430 (2013), DOI: 10.1134/S1990478913030150].
12. **V. L. Beresnev**, **Yu. A. Kochetov**, **M. G. Pashchenko**, [et al.]. *Discrete Location Problems. Benchmark Library* (Inst. Mat. SO RAN, Novosibirsk, 2021), URL: math.nsc.ru/AP/benchmarks/english.html (accessed: Nov. 1, 2024).

Artyom A. Panin
 Darya A. Piskeeva
 Aleksandr V. Plyasunov

Received May 8, 2024
 Revised May 15, 2024
 Accepted June 22, 2024

АЛГОРИТМ ПОИСКА РЕШЕНИЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ НА ПРОИЗВОДСТВЕ

*Н. П. Савенкова^{1, a}, А. Ю. Мокин^{1, 2, b},
А. А. Дряженков^{1, 2, c}, Л. А. Артемьева^{1, 2, d}*

¹Московский гос. университет им. М. В. Ломоносова,
Ленинские горы, 1, стр. 52, 119991 Москва, Россия

²Московский Центр фундаментальной и прикладной математики,
Ленинские горы, 1, 119991 Москва, Россия

E-mail: ^ansavenkova@cs.msu.su, ^bmknandrew@mail.ru,
^candrja@yandex.ru, ^dartemieva.luda@gmail.com

Аннотация. Предлагается математическая модель производства, на котором происходит переработка сырья. Производство состоит из установок, обрабатывающих сырьё, резервуаров для его хранения, а также из узлов смешения. В предположении, что каждая из перерабатывающих установок может работать в одном из двух известных режимов, а переключение с одного режима на другой может осуществляться не более одного раза, ставится задача поиска оптимальной мощности выработки продукции на каждой из установок, а также времени переключения установок с одного режима работы на другой, обеспечивающих выполнение заданного плана выработки окончательной продукции. Полученная задача представляет собой задачу дискретной оптимизации. Предлагается метод её решения, включающий переход к выпуклой постановке, а также алгоритм дискретизации полученного управления. Табл. 2, ил. 4, библиогр. 13.

Ключевые слова: календарное планирование, материальный баланс, оптимальное управление, дискретная оптимизация, квадратичное программирование.

Введение

Производственная схема нефтеперерабатывающего завода в большинстве случаев содержит до тысячи различных технологических объектов:

© Н. П. Савенкова, А. Ю. Мокин, А. А. Дряженков, Л. А. Артемьева, 2024

резервуары для хранения сырья, промежуточной продукции и продукции для отгрузки, манифолды, перераспределяющие продукцию между узлами предприятия, различные установки, обрабатывающие продукцию согласно predetermined рецептам. Все технологические объекты объединены развитой инфраструктурой, обеспечивающей доставку сырья и продукции от одного узла к другому. В настоящей работе рассматривается одна из ключевых задач производства — календарное планирование [1–4], в которой необходимо найти мощность каждой установки и время переключения установок с одного режима работы на другой, обеспечивающие выполнение заданного плана выработки окончательной продукции для продажи. Важной особенностью рассматриваемой в работе модели является возможность не более одного переключения режима работы на установках.

В настоящий момент времени к наиболее распространённым подходам для решения задачи календарного планирования можно отнести [5, 6]: методы, в основе которых лежит решение оптимизационных задач, методы статистического моделирования, методы, использующие имитационное моделирование, эвристические методы. Подходы, связанные с имитационным моделированием, включают в себя построение модели рассматриваемого производства, определение её параметров и дальнейшее использование этой модели с целью нахождения переменных, отражающих функционирование производства [7]. В методах статистического моделирования применяют случайное последовательное улучшение имеющегося расписания путём выбора в каждом из последовательно разыгрываемых множеств возможных расписаний лучшего расписания и запоминания рекордного из них. Такие методы позволяют получать приближённые календарные планы, однако их точность сложно оценить, а также сложно гарантировать оптимальность полученного решения.

В случае же использования оптимизационных методов поиск оптимального распределения сырья по резервуарам и установкам, выбор оптимальных режимов работы установок осуществляется за счёт решения задачи математического программирования. В большинстве случаев такие задачи целочисленные [7, 8]. В связи с этим многие существующие на текущий момент подходы к решению таких задач требуют перебора большого числа возможных стратегий, либо же приводят к задачам оптимизации большой размерности [9], что осложняет применение данного подхода для реального внедрения на производстве. В данной работе полученная модель представляет собой задачу оптимального управления [10], которую также можно привести к нелинейной дискретной задаче оптимизации. В настоящей работе предлагается подход к решению задачи календарного планирования, основанный на её сведении к выпуклой задаче оптимизации. Это даёт возможность избежать наличия

локальных минимумов в задаче оптимизации, а также позволяет для её решения применять методы, позволяющие избежать полного перебора возможных стратегий.

1. Математическая модель производства

В основе предлагаемой математической модели управления производством находится ориентированный граф G с множеством вершин $V = \{1, 2, \dots, N\}$, в котором каждой вершине $i \in V$ соответствует тот или иной технологический объект (обрабатывающая сырьё установка, резервуар, узел смешения и пр.). Рёбрам графа G соответствуют элементы инфраструктуры предприятия, обеспечивающие перемещение промежуточной продукции между технологическими объектами (трубопроводы, транспортные линии и пр.). Каждое ребро $(i, j) \in V^2$ графа G характеризуется величиной потока $q_{ij}(t)$, зависящей от времени t . Положительное значение потока означает перемещение продукции в направлении, определяемом ориентацией ребра.

В модели учитываются три типа вершин графа: установки — объекты, на которых происходит обработка промежуточной продукции, резервуары — объекты, в которых хранится промежуточная продукция, и манифолды — объекты, играющие роль распределительных узлов в инфраструктуре предприятия. Обозначим через U , R и M подмножества вершин графа G , которым соответствуют установки предприятия, резервуары и манифолды соответственно.

Потоки, входящие и исходящие из каждой вершины графа, связаны между собой уравнениями. В представленной модели рассматриваются уравнения трёх видов: соотношения материального баланса, соотношения материального баланса с накоплением продукции и уравнения рецептов производства [11]. Каждой вершине графа соответствует некоторый набор уравнений. Их вид определяется типом вершины.

Пусть $q_{ki}(t)$, $k \in P_i \subset V$, — потоки, входящие в вершину $i \in V$, а потоки, исходящие из неё, обозначим через $q_{ik}(t)$, $k \in Q_i \subset V$. Уравнение материального баланса для вершины i означает равенство суммы входящих и исходящих потоков и имеет вид

$$\sum_{k \in P_i} q_{ki}(t) = \sum_{k \in Q_i} q_{ik}(t), \quad i \in U \cup M. \quad (1)$$

Тем самым требуется, чтобы равенства (1) выполнялись на всех установках и манифолдах производства.

Предположим, что на объекте производства, которому соответствует вершина $i \in R$, происходит накопление промежуточной продукции (рис. 1). Обозначим через $x_i(t)$ количество этой продукции в момент времени t . Тогда уравнение материального баланса с накоплением для

вершины i имеет вид

$$\dot{x}_i(t) = \sum_{k \in P_i} q_{ki}(t) - \sum_{k \in Q_i} q_{ik}(t), \quad i \in R, \quad (2)$$

где $\dot{x}_i(t)$ — производная функции $x_i(t)$.

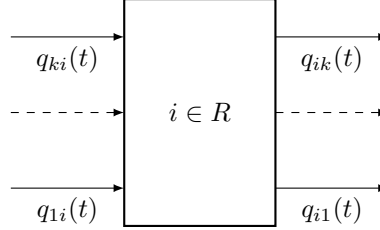


Рис. 1. Резервуар

Под уравнениями рецепта производства в вершине $i \in U$ понимается пропорциональная зависимость между каждым входящим (исходящим) потоком и суммарным входящим (исходящим) потоком этой вершины, т. е. имеют место соотношения

$$q_{ki} = \alpha_{ki} \sum_{s \in P_i} q_{si}, \quad k \in P_i, \quad q_{ik} = \beta_{ik} \sum_{s \in Q_i} q_{is}, \quad k \in Q_i, \quad i \in U, \quad (3)$$

где коэффициенты $\alpha_{ki}, \beta_{ik} \geq 0$ зависят, вообще говоря, от времени t .

Просуммировав по номеру k каждое из равенств (3), получаем

$$\sum_{k \in P_i} \alpha_{ki} = 1, \quad \sum_{k \in Q_i} \beta_{ik} = 1, \quad i \in U. \quad (4)$$

В дальнейшем будем считать эти равенства заведомо выполненными.

Из соотношений баланса (1) следует, что суммарный входящий поток равен суммарному исходящему потоку, поэтому уравнения рецептов (3) принимают вид

$$q_{ki} = \alpha_{ki} W_i, \quad k \in P_i, \quad q_{ik} = \beta_{ik} W_i, \quad k \in Q_i, \quad i \in U, \quad (5)$$

где $W_i = W_i(t)$ — суммарный поток установки в момент времени t . Заметим, что из равенств (4), (5) следует (1) для $i \in U$, поэтому от соотношений материального баланса на установках можно отказаться и оставить равенства (1) только для $i \in M$.

В настоящей модели допускается наличие не более двух режимов работы каждой установки $i \in U$, т. е. существование одного или двух рецептов функционирования. Рецепт характеризуется некоторым набором постоянных величин $\alpha_{ki}, k \in P_i$, и $\beta_{ik}, k \in P_i$. Множество U распадается на два подмножества $U^{(1)}$ и $U^{(2)}$, где $U^{(1)}$ — множество установок, имеющих только один рецепт работы, а $U^{(2)}$ — множество установок, имеющих

два рецепта и допускающих их переключение. Тем самым коэффициенты α_{ki} , $k \in P_i$, и β_{ik} , $k \in P_i$, в равенствах (5) являются известными постоянными величинами при всех $i \in U^{(1)}$. В случае двух режимов соотношения (5) для произвольного $i \in U^{(2)}$ принимают вид

$$q_{ki} = (s_i \alpha_{ki}^{(1)} + (1 - s_i) \alpha_{ki}^{(2)}) W_i, \quad q_{ik} = (s_i \beta_{ik}^{(1)} + (1 - s_i) \beta_{ik}^{(2)}) W_i, \quad (6)$$

где $\alpha_{ki}^{(1)}$, $\alpha_{ki}^{(2)}$, $\beta_{ik}^{(1)}$, $\beta_{ik}^{(2)}$ — известные постоянные величины, весовая функция $s_i = s_i(t)$ отвечает за выбор рецепта и равняется в каждый момент времени нулю или единице.

Управление установкой заключается в выборе её мощности работы $W_i(t)$ из допустимого диапазона $[W_{\min}^{(i)}, W_{\max}^{(i)}]$, а также в выборе режима работы $s_i(t) \in \{0, 1\}$ (при наличии двух рецептов). Цель управления состоит в том, чтобы к заданному моменту времени T выполнить план производства, т. е. накопить в резервуарах необходимое количество продукции. При этом предполагается, что время T невелико и план может быть выполнен с одним переключением установок или вовсе без переключений.

В множестве рёбер графа производства G выделим рёбра, не инцидентные ни одной установке, и обозначим их через $y_1, y_2, \dots, y_m$. Рёбра, инцидентные ровно одной установке, обозначим через $g_1, g_2, \dots, g_l$. Наличие рёбер, инцидентных двум установкам, в настоящей модели исключается.

2. Постановка задачи управления производством

Пусть E^n — евклидово пространство размерности n . Скалярное произведение векторов $a = (a^1, \dots, a^n)$, $b = (b^1, \dots, b^n) \in E^n$ записываем в виде $\langle a, b \rangle = \sum_{i=1}^n a^i b^i$, норма вектора a определяется равенством

$$\|a\| = \left(\sum_{i=1}^n (a^i)^2 \right)^{1/2}.$$

Обозначим через $L_2^r[0, T]$ лебегово пространство вектор-функций $u = u(t) = (u^1(t), \dots, u^r(t))$ с координатами $u^i(t)$, интегрируемыми по Лебегу вместе со своими квадратами на отрезке $[0, T]$. Скалярное произведение в пространстве $L_2^r[0, T]$ имеет вид

$$\langle u, v \rangle = \int_0^T \sum_{i=1}^r u^i(t) v^i(t) dt,$$

а норма вектор-функции $u(t)$ —

$$\|u(t)\| = \left(\int_0^T \sum_{i=1}^r (u^i(t))^2 dt \right)^{1/2}.$$

Занумеруем произвольным образом от 1 до n все резервуары графа производства и положим $x(t) = (x_1(t), \dots, x_n(t)) \in L_2^n[0, T]$, где $x_i(t)$ — запас продукции в i -м резервуаре. Пусть $y(t) \in L_2^m[0, T]$ — вектор потоков, не инцидентных установкам, а $g(t) \in L_2^l[0, T]$ — вектор потоков, инцидентных установкам производства.

Процесс планового производства нефтепродуктов к моменту времени T от начала производства описывается следующей задачей оптимального управления:

$$\dot{x}(t) = A_1 y(t) + C_1 g(t), \quad 0 < t < T, \quad x(0) = x_0, \quad (7)$$

$$x(t) \in X, \quad y(t) \in Y, \quad 0 < t < T, \quad (8)$$

$$A_2 y(t) + C_2 g(t) = 0, \quad 0 < t < T, \quad (9)$$

$$\|Zx(T) - z_T\|^2 \rightarrow \min, \quad (10)$$

где T — финальный момент времени, x_0 — начальное состояние резервуаров, $z_T \in \mathbb{E}^p$ — желаемое финальное состояние, матрицы $A_1 \in \mathbb{E}^{n \times m}$, $C_1 \in \mathbb{E}^{n \times l}$, $A_2 \in \mathbb{E}^{r \times m}$, $C_2 \in \mathbb{E}^{r \times l}$, $Z \in \mathbb{E}^{p \times n}$ заданы и заполняются по графу производства. Под решением задачи (7) будем понимать абсолютно непрерывную функцию $x(t)$, удовлетворяющую уравнению (7) почти всюду [10].

Дифференциальные уравнения (7) представляют собой уравнения баланса для совокупности резервуаров графа, т. е. являются векторной формой записи уравнений типа (2). Следующие после дифференциальных уравнений r однородных линейных алгебраических уравнений (9) описывают уравнения баланса на манифолдах и являются векторной формой записи соотношений (1). Вектор z_T задан и является вектором плановых значений продукции, которые должны быть запасены к моменту времени T . Его размерность p равна количеству резервуаров графа, для которых задан план производства. Матрица Z состоит из нулей и единиц и получается из матрицы $(I|0)$ размера $p \times n$ перестановкой столбцов, где I — единичная матрица порядка p , 0 — нулевая матрица размера $p \times (n - p)$.

Задающие ограничения на резервуары и потоки множества X и Y являются ограниченными параллелепипедами со сторонами, параллельными осям координат:

$$X = \{x \in \mathbb{R}^n \mid x_1 \leq x \leq x_2\}, \quad Y = \{y \in \mathbb{R}^m \mid y_1 \leq y \leq y_2\}.$$

Компоненты вектора $g(t)$ согласно формулам (5), (6) выражаются через функции управления мощностями установок $W_i(t)$, $i \in U$, и функции выбора режимов работы установок $s_i(t)$, $i \in U^{(2)}$. При этом

$$W_{\min}^{(i)} \leq W_i(t) \leq W_{\max}^{(i)}, \quad 0 \leq t \leq T, \quad i \in U,$$

а функции $s_i(t)$, $i \in U^{(2)}$, кусочно постоянные:

$$s_i(t) = \begin{cases} 1 & \text{при } 0 \leq t < t^{(i)}, \\ 0 & \text{при } t^{(i)} \leq t \leq T, \end{cases} \quad (11)$$

где $t^{(i)}$ — момент переключения режима работы установки $i \in U^{(2)}$.

3. Замена переменных. Овыпукленная модель

Задача (7)–(10) представляет собой задачу оптимального управления с фазовыми ограничениями, в которой роль управлений играют функции $W_i = W_i(t)$, $i \in U$, и $s_i = s_i(t)$, $i \in U^{(2)}$. В роли фазовых переменных выступают функции $x_i = x_i(t)$, $i = 1, 2, \dots, n$. В силу квадратичной зависимости (6) потоков, инцидентных установкам, от управлений динамическая система (7) нелинейна. Для её сведения к выпуклой системе будем предполагать, что $s_i(t) \in [0, 1]$, $i \in U^{(2)}$, а для её линеаризации сделаем следующую замену: для всех установок, на которых имеется один рецепт, положим

$$w_i(t) = W_i(t), \quad i \in U^{(1)}. \quad (12)$$

Для всех установок с двумя рецептами положим

$$w_i^1(t) = s_i(t)W_i(t), \quad w_i^2(t) = (1 - s_i(t))W_i(t), \quad i \in U^{(2)}. \quad (13)$$

В этом случае управление установкой будет заключаться в выборе управляющих параметров $w_i(t)$, $i \in U^{(1)}$, и $w_i^1(t)$, $w_i^2(t)$, $i \in U^{(2)}$, из допустимого диапазона, который представляет собой отрезок $[W_{\min}^{(i)}, W_{\max}^{(i)}]$ для всех управляющих параметров, инцидентных установкам с одним рецептом. Для установок с двумя рецептами имеют место следующие ограничения на управляющие параметры:

$$W_{\min}^{(i)} \leq w_i^1(t) + w_i^2(t) \leq W_{\max}^{(i)}.$$

С учётом новых обозначений равенства (6) принимают вид

$$q_{ki} = \alpha_{ki}^{(1)} w_i^1(t) + \alpha_{ki}^{(2)} w_i^2(t), \quad q_{ik} = \beta_{ki}^{(1)} w_i^1(t) + \beta_{ki}^{(2)} w_i^2(t).$$

Составим из функций $w_i(t)$, $i \in U^{(1)}$, и $w_i^1(t)$, $w_i^2(t)$, $i \in U^{(2)}$, определённых согласно (12) и (13), вектор-функцию $u(t)$, которую примем за управление. Итоговая задача оптимального управления формально

будет иметь тот же вид, что и в (7)–(10), однако благодаря приведённой выше замене данную задачу можно будет интерпретировать как выпуклую задачу оптимизации. Решая полученную задачу минимизации с ограничениями, можно найти управление как непрерывную функцию, зависящую от времени. Это связано с технической необходимостью перехода от исходной задачи с дискретным управлением к задаче в непрерывной форме. Однако в силу неединственности решения задачи оптимизации восстановление по полученному управлению его дискретного аналога без дополнительной информации может быть затруднительным. В этом случае для решения задачи календарного планирования при наличии не более одного переключения на установках предлагается воспользоваться подходом, связанным с регуляризацией исходной задачи [10], а именно: заменить целевой функционал (10) на

$$J_{\text{reg}}(u) = \|Zx(T) - z_T\|^2 + \alpha \langle c, u \rangle,$$

где

$$\langle c, u \rangle = \sum_{i \in U^{(2)}} \int_0^T (tw_i^1(t) + (T-t)w_i^2(t)) dt,$$

$\alpha > 0$ — некоторая фиксированная малая величина.

При $\alpha \rightarrow 0$ решение регуляризованной задачи сходится к решению исходной задачи, а такой вид минимизируемой функции позволяет учесть особенность дискретного вида управления. Далее полученную задачу минимизации с ограничениями предлагается решать методом внутренней точки [12], который относится к наиболее успешным на текущий момент методам решения задач оптимизации большой размерности при наличии ограничений [13].

4. Переход к дискретному управлению

От полученного непрерывного управления необходимо перейти к дискретному, в котором в любой момент времени $t \in [0, T]$ каждая установка работает в одном и только одном режиме.

Рассмотрим произвольную установку $i \in U^{(2)}$, допускающую переключение режимов. Пусть к моменту времени $t \in [0, T]$ она вырабатывает следующее количество продукции в первом и втором режимах соответственно:

$$M_i^1(t) = \int_0^t w_i^1(p) dp, \quad M_i^2(t) = \int_0^t w_i^2(p) dp.$$

В частности, $M_i^1(T)$ и $M_i^2(T)$ — общий объём продукции, выработанный установкой $i \in U$ в первом и втором режимах за весь интервал времени $[0, T]$. Величина $M_i^1(t)/M_i^1(T)$ представляет собой долю продукции, выработанную в первом режиме к моменту времени t , а разность $1 - M_i^2(t)/M_i^2(T)$ равна доле продукции второго режима, которую предстоит выработать за интервал времени $[t, T]$.

Для перехода к дискретному управлению вычислим время переключения режимов $t = t_i^*$ на установке $i \in U$ из соотношения

$$\frac{M_i^1(t)}{M_i^1(T)} = 1 - \frac{M_i^2(t)}{M_i^2(T)},$$

т. е. из равенства

$$\frac{1}{M_i^1(T)} \int_0^t w_i^1(p) dp + \frac{1}{M_i^2(T)} \int_0^t w_i^2(p) dp = 1.$$

В силу монотонности левой части данного уравнения по переменной t время переключения определено однозначно.

В дальнейшем будем считать, что при $0 \leq t < t_i^*$ установка $i \in U^{(2)}$ работает в первом режиме, при $t_i^* \leq t \leq T$ — во втором. Тем самым функция $s_i(t)$, обеспечивающая переключение режимов, определяется однозначно равенством (11), в котором $t^{(i)} = t_i^*$.

Определив функцию переключения режимов на всех установках, допускающих переключение, необходимо вычислить новые значения потоков продукции и запасов резервуаров, удовлетворяющих всем ограничениям задачи календарного планирования. Для этого следует заново минимизировать квадратичный целевой функционал J_{reg} с ограничениями (7)–(9). Принципиальное отличие от ранее решённой задачи (7)–(10) заключается в том, что функции $s_i(t)$ уже выбраны и потоки q_{ki} , q_{ik} , инцидентные установкам с переключением режимов, согласно формулам (6) зависят только от мощностей $W_i = W_i(t)$. Иначе говоря, требуется решить задачу (7)–(10), в которой минимизация целевого функционала происходит только за счёт выбора мощностей установок из допустимого диапазона значений. Эта задача выпуклая и может быть решена стандартными методами решения задач квадратичного программирования (см. [1]).

5. Пример задачи календарного планирования

На рис. 2 представлен граф производства, включающий четыре установки Y_i , $i = 1, \dots, 4$, семь узлов смешения продукции (I)–(VII) и семь

резервуаров накопления и хранения продукции R_i , $i = 1, \dots, 7$. Подписанными стрелками на графе отмечены входящие и выходящие потоки продукции.

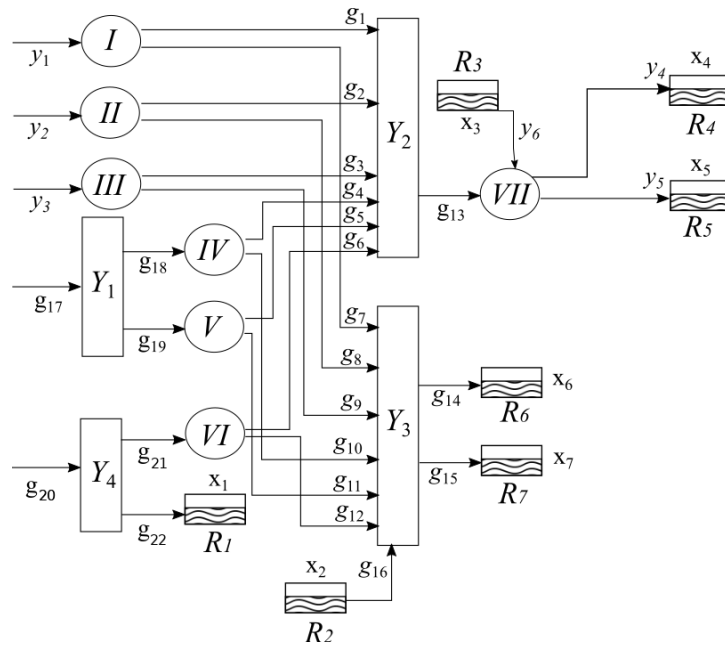


Рис. 2. Фрагмент графа предприятия

Отличительной особенностью этого графа является то, что среди четырёх установок первые три имеют изменяющиеся со временем потоки, а на четвёртой установке все потоки стационарны. Для всех потоков и резервуаров заданы диапазоны изменения их величин. Известны период времени $[0, T]$, на котором ведётся планирование, и плановый объём некоторых видов продукции за этот период, а также начальное значение запасов во всех резервуарах. На каждой установке заданы рецепты режимов работы.

Требуется определить мощности установок и их моменты переключений с одного режима работы на другой с целью выполнения плана производства разных видов продукции на заданном отрезке времени (календарное планирование).

Установки Y_1 , Y_2 и Y_3 могут работать в одном из двух режимов, суммарный поток, проходящий через установку, нефиксирован, а режим установки Y_4 неизменен, и суммарный поток, проходящий через установку, фиксирован и равен $W_4 = 0,96$.

Рассмотрим рецепты, заданные на установках. Начнём с первой установки. Суммарный поток на этой установке обозначим через $W_1(t) = w_1^1(t) + w_1^2(t)$. Известно, что $2,25 \leq W_1(t) \leq 2,35$. Тогда в первом режиме на этой установке

$$g_{17}(t) = 1 \cdot w_1^1(t), \quad g_{18}(t) = 1 \cdot w_1^1(t), \quad g_{19}(t) = 0 \cdot w_1^1(t),$$

а во втором

$$g_{17}(t) = 1 \cdot w_1^2(t), \quad g_{18}(t) = 0,4815 \cdot w_1^2(t), \quad g_{19}(t) = 0,5185 \cdot w_1^2(t).$$

За суммарную производительность второй установки примем $W_2(t) = w_2^1(t) + w_2^2(t)$. Тогда в первом режиме на второй установке

$$\begin{aligned} g_{13}(t) &= 1 \cdot w_2^1(t), \quad g_1(t) = 0,6592 \cdot w_2^1(t), \quad g_2(t) = 0,07 \cdot w_2^1(t), \\ g_3(t) &= 0,1008 \cdot w_2^1(t), \quad g_4(t) = 0,17 \cdot w_2^1(t), \\ g_5(t) &= 0 \cdot w_2^1(t), \quad g_6(t) = 0 \cdot w_2^1(t), \end{aligned}$$

во втором режиме на второй установке

$$\begin{aligned} g_{13}(t) &= 1 \cdot w_2^2(t), \quad g_1(t) = 0,7333 \cdot w_2^2(t), \quad g_2(t) = 0,0383 \cdot w_2^2(t), \\ g_3(t) &= 0,0583 \cdot w_2^2(t), \quad g_4(t) = 0,1204 \cdot w_2^2(t), \\ g_5(t) &= 0,0366 \cdot w_2^2(t), \quad g_6(t) = 0,0131 \cdot w_2^2(t). \end{aligned}$$

Аналогично в первом режиме на третьей установке с суммарной производительностью $W_3(t) = w_3^1(t) + w_3^2(t)$ имеем

$$\begin{aligned} g_{14}(t) &= 1 \cdot w_3^1(t), \quad g_{15}(t) = 0 \cdot w_3^1(t), \quad g_7(t) = 0,5268 \cdot w_3^1(t), \\ g_8(t) &= 0,0534 \cdot w_3^1(t), \quad g_9(t) = 0,0453 \cdot w_3^1(t), \quad g_{10}(t) = 0,3341 \cdot w_3^1(t), \\ g_{11}(t) &= 0 \cdot w_3^1(t), \quad g_{12}(t) = 0,0404 \cdot w_3^1(t), \quad g_{16}(t) = 0 \cdot w_3^1(t), \end{aligned}$$

а во втором режиме на третьей установке

$$\begin{aligned} g_{14}(t) &= 0 \cdot w_3^2(t), \quad g_{15}(t) = 1 \cdot w_3^2(t), \quad g_7(t) = 0 \cdot w_3^2(t), \\ g_8(t) &= 0,2219 \cdot w_3^2(t), \quad g_9(t) = 0,2383 \cdot w_3^2(t), \quad g_{10}(t) = 0 \cdot w_3^2(t), \\ g_{11}(t) &= 0,5078 \cdot w_3^2(t), \quad g_{12}(t) = 0 \cdot w_3^2(t), \quad g_{16}(t) = 0,032 \cdot w_3^2(t). \end{aligned}$$

Четвёртая установка имеет постоянную производительность $W_4(t) = w_4(t) = 0,96$ и постоянно работает в следующем режиме:

$$\begin{aligned} g_{22} &= 0,877 \cdot w_4 = 0,877 \cdot 0,96 = 0,84192, \\ g_{21} &= 0,123 \cdot w_4 = 0,11808, \quad g_{20} = 1 \cdot w_4 = 0,96. \end{aligned}$$

Также предполагаются заданными ограничения на входящие потоки в первые три узла смешения:

$$6,6 \leq y_1 \leq 6,9, \quad 0,7 \leq y_2 \leq 0,73, \quad 0,9 \leq y_3 \leq 0,95.$$

В начальный момент времени все установки работают в первом режиме (каждая в своём). Дифференциальные уравнения из (1) имеют для этого примера вид

$$\begin{aligned}\dot{x}_1 &= 0,84192, & \dot{x}_2 &= -g_{16}, & \dot{x}_3 &= -y_6, \\ \dot{x}_4 &= y_4, & \dot{x}_5 &= y_5, & \dot{x}_6 &= g_{14}, & \dot{x}_7 &= g_{15}.\end{aligned}$$

Алгебраические уравнения, возникающие из-за наличия узлов смешения, имеют вид

$$\begin{aligned}y_1 - g_1 - g_7 &= 0, & y_2 - g_2 - g_8 &= 0, \\ y_3 - g_3 - g_9 &= 0, & g_{18} - g_{10} - g_4 &= 0, \\ g_{19} - g_{11} - g_5 &= 0, & 0,11808 - g_{12} - g_6 &= 0, \\ 0,9998y_5 - (g_{13} - y_4) &= 0, & g_{13} + y_6 - y_4 - y_5 &= 0.\end{aligned}$$

По плану производства количество продукции типов 1, 4, 5, 6, 7 за весь период планирования равно 7,57728, 34,94, 42,55, 11,7, 8,5 соответственно.

Таблица 1

**Результаты расчётов количества
выработанной продукции**

Тип	1	4	5	6	7
Количество	7,5573	34,9330	42,5430	11,6974	8,4941

К рассматриваемому примеру применён предложенный выше метод решения задачи календарного планирования. В табл. 1 представлены результаты расчётов плановых величин. Из таблицы видно, что максимальная погрешность их вычисления не превосходит 0,02.

Несложно показать, что в описанном примере аналитическое решение единственно и точное время переключения установок 1, 2, 3 совпадает и равно 4. В табл. 2 представлены полученное в результате расчётов время переключения на сетках с разным числом узлов τ . Из таблицы следует, что отклонение расчётного времени переключения от точных значений уменьшается с ростом τ .

Таблица 2

**Время переключения каждой из установок
при разных значениях числа узлов τ по времени**

τ	Установка 1	Установка 2	Установка 3
45	3,9	4,1	3,9
90	3,95	4,05	3,95
180	3,975	4,025	3,975

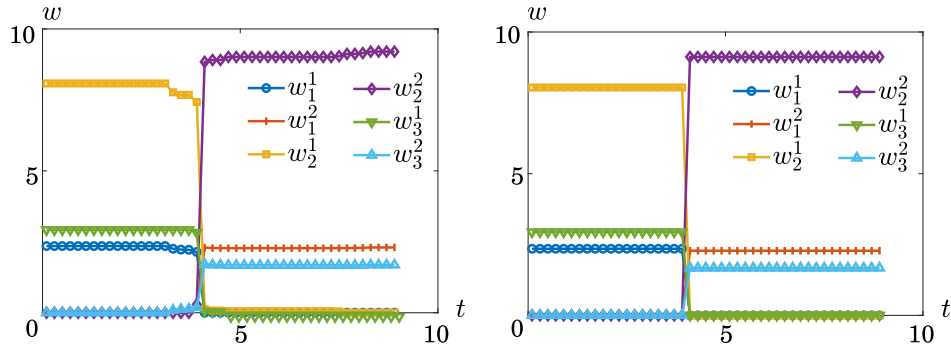


Рис. 3. Производительность установок до дискретизации управления (слева) и после дискретизации управления (справа)

Графики производительности установок в зависимости от времени изображены на рис. 3.

Нетрудно видеть, что графики производительности установок слева (до дискретизации) не могут быть использованы для управления производством. Действительно, в интервале времени $t \in (3, 4)$ вторая установка одновременно работает в двух режимах, что не может быть реализовано на практике. После дискретизации (графики справа) управление имеет ярко выраженный вид «ступенек», что соответствует реальному режиму работы установок.

На рис. 4 представлены графики запасов в резервуарах: слева — для резервуаров 2 и 3, справа — для остальных. Отметим, что значения запасов в резервуарах до дискретизации отличаются от их значений после дискретизации незначительно.

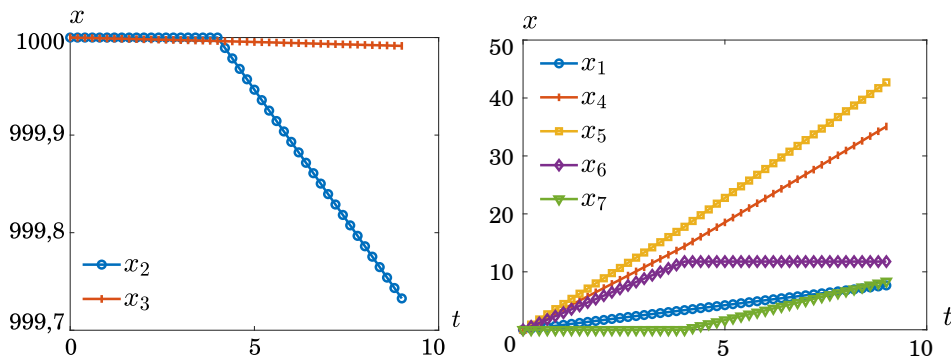


Рис. 4. Графики запасов в резервуарах после дискретизации

Заключение

В работе предложен оригинальный подход к решению задачи календарного планирования, в рамках которого предлагается исходную задачу дискретной оптимизации привести к выпуклой задаче непрерывной минимизации, а затем вернуться к решению дискретной задачи с помощью специально разработанного алгоритма. Отметим, что такой подход позволяет использовать разработанный на текущий момент аппарат для решения задач оптимизации [10, 13].

Предложенный в работе пример является участком нефтеперерабатывающего завода. Решение, представленное на рис. 3, 4, получено в результате расчётов на однопроцессорном компьютере в однопоточном режиме. Весь завод состоит более чем из тысячи объектов. Возможно применение алгоритма к расчёту календарного плана всего производства без потери точности вычислений, что потребует увеличения расчётного времени и, как следствие, переход к параллельным вычислениям на многопроцессорных комплексах.

Метод внутренней точки, использованный для получения календарного плана, является одношаговым итерационным алгоритмом условной минимизации. Как во всяком итерационном алгоритме, распараллеливанию подлежат вычисления внутри итерации, связанные с умножением матриц, умножением матрицы на вектор и т. п. Расчётное время напрямую зависит от числа итераций, необходимых для достижения требуемой точности. Число итераций, в свою очередь, зависит от выбора итерационных параметров и начального приближения.

Заметим, что время расчёта также зависит от выбора шага сетки по времени на отрезке $[0, T]$. Вычисления на грубых сетках требуют значительно меньше времени, чем на сетках с мелким шагом, поэтому при построении календарного плана имеет смысл проводить расчёты на сгущающихся сетках: сначала выполнить расчёт на грубой сетке и взять его результаты в качестве начального приближения для дальнейших вычислений.

Финансирование работы

Исследование выполнено за счёт бюджетов организаций, указанных авторами на первой странице статьи. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Смирнов А. Н.** Объёмное календарное планирование — основа оперативного управления // Вестн. Санкт-Петербург. ун-та. Сер. 10. Прикл. математика. Информатика. Процессы управления. 2005. № 1–2. С. 161–170.
2. **Смирнов А. Н.** Математическая модель объёмного календарного планирования технологически зависимых операций // Автоматика и телемеханика. 1985. № 10. С. 108–115.
3. **Дудников Е. Е., Цодиков Ю. М.** Типовые задачи оперативного управления непрерывным производством. М.: Энергия, 1979. 272 с.
4. **Кувыкин В. И.** Оптимальное планирование и анализ моделей непрерывного производства // Автоматизация в промышленности. 2015. № 8. С. 161–170.
5. **Антамошкина Е. А., Шарыпова К. В.** Эвристический алгоритм составления расписаний дискретных производств // Информатика и системы управления. 2010. Т. 23, № 1. С. 67–73.
6. **Левин В. И.** Некоторые мысли о теории расписаний // Вестн. Тамбов. гос. техн. ун-та. 2005. Т. 11. № 2. С. 341–347.
7. **Шайдуллин Р. А., Хохлов А. С., Проказина М. В.** Имитационные модели в комплексе календарного планирования НПЗ // Автоматизация в промышленности. 2012. № 10. С. 15–21.
8. **Kaneva O. N., Zykina A. V., Volodchenko M. M.** The problem of mixed integer programming for optimal schedule of petroleum products manufacturing // Proc. Workshop Applied Mathematics and Fundamental Computer Science (Omsk, Russia, Apr. 24–29, 2021). Aachen: RWTH Aachen Univ., 2021. Paper ID 2. 7 p. (CEUR Workshop Proc.; Vol. 2928). URL: ceur-ws.org/vol-2928/paper2.pdf (accessed: Oct. 1, 2024).
9. **Прилуцкий М. Х.** Многокритериальное распределение однородного ресурса с иерархических системах // Автоматика и телемеханика. 1996. № 2. С. 139–146.
10. **Васильев Ф. П.** Методы оптимизации. В 2-х кн. М.: МЦНМО, 2011. 624 с.; 434 с.
11. **Савенкова Н. П., Артемьева Л. А., Лапонин В. С., Мокин А. Ю., Дряженков А. А.** Математическое моделирование сведения материальных балансов // Деловой журн. Neftegaz.RU. 2018. № 4. С. 91.
12. **Евтушенко Ю. Г.** Методы решения экстремальных задач и их применение в системах оптимизации. М.: Наука, 1982. 412 с.
13. **Nocedal J., Wright S. J.** Numerical optimization. New York: Springer, 2006. 664 p.

Савенкова Надежда Петровна
Мокин Андрей Юрьевич
Дряженков Андрей Александрович
Артемьева Людмила Анатольевна

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

AN ALGORITHM TO SEARCH A SOLUTION
TO THE PRODUCTION SCHEDULING PROBLEM

N. P. Savenkova^{1, a}, *A. Yu. Mokin*^{1, 2, b},
A. A. Dryazhenkov^{1, 2, c}, and *L. A. Artemyeva*^{1, 2, d}

¹Lomonosov Moscow State University,
1 Bld. 52 Leninskie Gory, 119991 Moscow, Russia

²Moscow Centre of Fundamental and Applied Mathematics,
1 Leninskie Gory, 119991 Moscow, Russia

E-mail: ^ansavenkova@cs.msu.su, ^bmknandrew@mail.ru,
^candrja@yandex.ru, ^dartemieva.luda@gmail.com

Abstract. A mathematical model of raw materials processing is proposed. The production consists of units processing raw materials, storage tanks, and mixing units. The processing units are assumed to operate in one of two known modes. Switching from one mode to another can be carried out no more than once. The problem of finding the optimal capacity of production at each of units, as well as the time of switching units from one operating mode to another is formulated in a form of discrete optimization problem. The solution of this problem should ensure an achievement of the specified production plan. A method for its solution is proposed, including a transition to a convex statement, as well as an algorithm for discretizing the obtained control. Tab. 2, illustr. 4, bibliogr. 13.

Keywords: task scheduling, material balance, optimal control, discrete optimization, quadratic programming.

References

1. **A. N. Smirnov**, Three-dimensional scheduling — The basis for operational management, *Vestn. Sankt-Peterburg. Univ., Ser. 10*, No. 1–2, 161–170 (2005) [Russian].
2. **A. N. Smirnov**, Mathematical model of three-dimensional scheduling of technologically linked activities, *Autom. Telemekh.*, No. 10, 108–115 (1985) [Russian] [*Autom. Remote Control* **46**, 1297–1303 (1985)].

3. **E. E. Dudnikov** and **Yu. M. Tsodikov**, *Typical Tasks in Operational Management of Continuous Production* (Ehnergiya, Moscow, 1979) [Russian].
4. **V. I. Kuvykin**, Optimal scheduling and analysis of continuous production models, *Avtom. Prom.*, No. 8, 161–170 (2015) [Russian].
5. **E. A. Antamoshkina** and **K. V. Sharypova**, Heuristic algorithm for scheduling discrete production, *Inform. Sist. Upr.* **23** (1), 67–73 (2010) [Russian].
6. **V. I. Levin**, Some thoughts on scheduling theory, *Vestn. Tambov. Gos. Tekh. Univ.* **11** (2), 341–347 (2005) [Russian].
7. **R. A. Shaidullin**, **A. S. Khokhlov**, and **M. V. Prokazina**, Simulation models in the complex of oil refinery scheduling, *Avtom. Prom.*, No. 10, 15–21 (2012) [Russian].
8. **O. N. Kaneva**, **A. V. Zykina**, and **M. M. Volodchenko**, The problem of mixed integer programming for optimal schedule of petroleum products manufacturing, in *Proc. Workshop Applied Mathematics and Fundamental Computer Science, Omsk, Russia, Apr. 24–29, 2021* (RWTH Aachen Univ., Aachen, 2021), ID 2 (CEUR Workshop Proc., V. 2928), URL: ceur-ws.org/vol-2928/paper2.pdf (accessed: Oct. 1, 2024).
9. **M. Kh. Prilutskii**, Multicriteria distribution of a homogeneous resource in hierarchical systems, *Avtom. Telemekh.*, No. 2, 139–146 (1996) [Russian] [*Autom. Remote Control* **57** (2), Pt. 2, 266–271 (1996)].
10. **F. P. Vasilyev**, *Optimization Methods*, in 2 books (MTsNMO, Moscow, 2011) [Russian].
11. **N. P. Savenkova**, **L. A. Artemyeva**, **V. S. Laponin**, **A. Yu. Mokin**, and **A. A. Dryazhenkov**, Mathematical modeling of material balances, *Bus. Zh. Neftegaz.RU*, No. 4, 91 (2018) [Russian].
12. **Yu. G. Evtushenko**, *Methods for Solution of Extremal Problems and Its Application in Optimization Systems* (Nauka, Moscow, 1982) [Russian].
13. **J. Nocedal** and **S. J. Wright**, *Numerical optimization* (Springer, New York, 2006).

Nadezhda P. Savenkova
 Andrey Yu. Mokin
 Andrey A. Dryazhenkov
 Lyudmila A. Artemyeva

Received March 22, 2024
 Revised April 27, 2024
 Accepted June 22, 2024

О СЛОЖНОСТИ ПАРАЛЛЕЛЬНЫХ ПРЕФИКСНЫХ СХЕМ С ОГРАНИЧЕННЫМ ВЕТВЛЕНИЕМ

И. С. Сергеев

Научно-исследовательский институт «Квант»,
4-й Лихачёвский пер., 15, 125438 Москва, Россия
E-mail: isserg@gmail.com

Аннотация. Доказано, что сложность универсальной префиксной схемы глубины n на 2^n входах с ограничением 2 на ветвление выходов элементов не меньше $0,75(n-1)2^n$. Также приводятся несколько простых конструкций и верхних оценок сложности префиксных схем с ветвлением 2 и глубиной $n+k$. Ил. 4, библиогр. 14.

Ключевые слова: параллельная префиксная схема, схема с ограниченным ветвлением, сложность схемы, глубина схемы.

Введение

Пусть $(S, \circ)$ — полугруппа. Множество функций

$$x_1 \circ x_2 \circ \dots \circ x_i, \quad 1 \leq i \leq m, \quad (1)$$

называется *системой префиксных сумм* переменных $x_1, \dots, x_m$, принимающих значения в S . Схемы из функциональных элементов над базисом $\{x \circ y, x\}$, реализующие систему (1), называют *префиксными схемами*. Напомним, что *сложностью* и *глубиной* схемы называются соответственно общее число элементов схемы и максимальное число элементов в ориентированном пути, соединяющем вход и выход схемы. Подробнее о понятиях схемы, сложности и глубины см., например, в [1, 2]. (Примеры префиксных схем показаны на рис. 1.) Под *ветвлением* вершины в графе (или элемента в схеме) далее понимаем число рёбер, исходящих из неё.

Префиксные схемы применяются во многих теоретических и прикладных задачах синтеза, например в конструкциях сумматоров двоичных чисел, а также в задачах сортировки, решения линейных рекуррентных последовательностей. В ряде задач возникает потребность в построении *параллельных* префиксных схем, т. е. имеющих глубину $O(\log m)$ относительно числа входов m .

Универсальные префиксные схемы правильно вычисляют суммы независимо от выбора полугруппы S ¹⁾. Легко проверить, что избыточная²⁾ универсальная схема использует только операции вида $s_1 \circ s_2$, где $s_1 = x_i \circ x_{i+1} \circ \dots \circ x_j$ и $s_2 = x_{j+1} \circ x_{j+2} \circ \dots \circ x_k$.

Пусть $L(P)$ означает сложность схемы P , а $L_f(k, m)$ — сложность минимальной универсальной префиксной схемы m переменных и глубины $\lceil \log_2 m \rceil + k$ с ограничением f на ветвление входов схемы и внутренних элементов и ограничением $f - 1$ на ветвление выходов схемы³⁾. Положим $L_f(m) = L_f(0, m)$ и $L() = L_\infty()$. Далее, как правило, ограничиваемся рассмотрением наиболее интересного и важного случая $m = 2^n$.

Разнообразные конструкции параллельных префиксных схем предлагались с 1960-х гг., но только около 1980 г. Ладнер и Фишер [3] показали, что можно строить схемы минимальной глубины и линейной сложности. В частности, ими было доказано, что $L(2^n) < 4 \cdot 2^n$. Вскоре Фич [4] улучшила верхнюю оценку и получила нетривиальную нижнюю: $L(2^n) \geq (10/3 - o(1)) \cdot 2^n$. Автор [5, 6] установил точное значение сложности, имеющее вид $L(2^n) = (3,5 - o(1)) \cdot 2^n$. Лучшие верхние оценки для схем почти минимальной глубины также получены в [5, 6]. Они имеют вид $L(k, 2^n) \leq (2 + 2^{-k})2^n - O(2^{(n-k)/2})$ при $1 \leq k \ll n$. При $k \geq (\log_\varphi 2 - 1)n - O(1) \approx 0,44n - O(1)$, где $\varphi = \frac{1+\sqrt{5}}{2}$, выполнено $L(k, 2^n) = 2^{n+1} - n - k - 2$ (см. [7]). Более подробное изложение этой линии результатов см. в [6].

С точки зрения электроники существенное влияние на характеристики схемы, помимо глубины и сложности, оказывает также степень ветвления элементов: чем степень меньше, тем лучше. Поэтому значительный объём прикладных и теоретических исследований посвящён параллельным префиксным схемам с ограниченным ветвлением. Отметим, что только ограничение 2 на ветвление принуждает к использованию тождественных элементов. В других случаях без них можно обойтись.

Изящную конструкцию схем минимальной глубины и ветвления 2 предложили в начале 1970-х гг. Коге и Стоун [8]. Их пример доказывает оценку

$$L_2(2^n) \leq (n - 0,5)2^n. \quad (2)$$

Принципиальный результат получила Фич [4], показав, что линейная сложность для схем минимальной глубины с ветвлением 2 недостижима:

$$L_2(2^n) \geq (n + 1)2^{n-1} - O(n^2). \quad (3)$$

¹⁾ Достаточно потребовать, чтобы результат был правильным, скажем, для группы симметрий окружности (см. [6]).

²⁾ Т. е. не содержащая элементов, не связанных с выходами.

³⁾ Более жёсткое условие на ветвление выходов накладывается, чтобы обеспечить возможность использования префиксных схем в качестве подсхем.

С другой стороны, при любом $f \geq 3$ выполнено $L_f(2^n) = O(2^n)$. Фич указала способ построения таких схем⁴⁾.

Ослабляя ограничение на глубину, можно добиться линейной сложности и для схем с ветвлением 2. Подобную схему построили Брент и Кунг [9]. Их конструкция влечёт оценку $L_2(n-1, 2^n) \leq 2,5 \cdot 2^n - n - 3$. Используя рекурсивный метод [3], но выбирая схему Коге — Стоуна в качестве базовой, Хан и Карлсон [10] построили семейство схем, приводящее при $1 \leq k \leq n-1$ к оценке сложности⁵⁾

$$L_2(k, 2^n) \leq (n-k-3)2^{n-k} + 5 \cdot 2^{n-1} - k. \quad (4)$$

Автор [6] предложил модификацию базовой схемы, уточнив оценку сложности при $1 \leq k \leq n-2$ до

$$L_2(k, 2^n) \leq (n-k-3,25)2^{n-k} + 5 \cdot 2^{n-1} - k. \quad (5)$$

Схемы из двух семейств отличаются только числом тождественных элементов. Нижняя оценка из [4] имеет вид

$$L_2(k, 2^n) \geq (n-2)2^{n-k-1} + 2^n - O(n^2 2^{-k})$$

и содержательна только при $k \ll \log_2 n$, поскольку при больших k приоритет имеет оценка $L(k, 2^n) \geq 2^{n+1} - n - k - 2$, вытекающая из стандартного соотношения

$$L + D \geq 2m - 2, \quad (6)$$

справедливого для произвольной префиксной схемы m переменных глубины D ⁶⁾ с числом двухвходовых элементов L [4, 11]. Схемы, на которых достигается равенство в (6), называют *оптимальными*.

Схемы линейной сложности с ветвлением $f \geq 3$, построенные Фич [4], не вполне практичны, поскольку мультипликативные константы в оценках велики: например, $L_3(2^n) \leq 54,25 \cdot 2^n$. В современной работе [12] авторы путём модификации семейства схем Ладнера — Фишера получили более аккуратные оценки, в частности $L_3(2^n) \leq (7,75 + o(1)) \cdot 2^n$.

Имеется обширная литература по параллельным префиксным схемам с различными ограничениями на ветвление элементов. Например, в работах [13, 14] построены оптимальные параллельные префиксные схемы с ветвлением 2 и максимально возможным числом входов (там же приводится подробная библиография по этому направлению). Минимально возможная глубина оптимальной схемы с ветвлением 2 на m входах

⁴⁾ Она также получила верхние оценки сложности схем с ветвлением 2, но они, во-первых, превышают сложность схем Коге — Стоуна, а во-вторых, демонстрируются на схемах, некоторые входы которых имеют ветвление 3.

⁵⁾ В качестве частного случая при $k = n-1$ получается схема Брента — Кунга.

⁶⁾ Более точно, оценка доказывается для D , равного глубине вычисления старшей префиксной суммы — суммы всех переменных — в схеме.

оказывается равной $\lfloor \log_2 m \rfloor + \lfloor \log_2(2m/3) \rfloor$. Однако общая сложность схем с учётом тождественных элементов авторами не оценивалась.

Далее докажем нижнюю оценку $L_2(2^n) > \frac{3}{4}(n-1)2^n$, сокращая разрыв между известными верхней (2) и нижней (3) оценками сложности параллельных префиксных схем с ветвлением 2. Новая оценка опирается на свойство минимальных схем вычислять только «интервальные» суммы вида $x_l \circ x_{l+1} \circ \dots \circ x_r$, поэтому справедлива не только для универсальных схем, но и для схем над конкретными некоммутативными полугруппами, включая идемпотентные. В частности, она применима в случае полугруппы, используемой при построении (префиксных) сумматоров⁷⁾. Отметим, что предшествующая оценка Фич выполняется для схем над любой конкретной полугруппой, за исключением вырожденных случаев⁸⁾.

В заключение сделаем несколько простых замечаний относительно верхних оценок сложности. Сначала покажем, что оценки (4), (5) могут быть уточнены на величину $O(2^k)$. Далее, докажем, что асимптотически минимально возможная для параллельной схемы оценка $L_2(k, m) = (2 + o(1))m$ достигается при $k = O(\log \log m)$, а при $k = (1 + o(1)) \log_2 m$ демонстрируется оптимальными префиксными схемами.

1. Нижняя оценка

Лемма 1. В корневом бинарном дереве с m листьями минимально возможная сумма длин всех путей от корня к листьям равна $(l+1)m - 2^l$, где $l = \lfloor \log_2 m \rfloor$.

ДОКАЗАТЕЛЬСТВО. Это решение известной оптимизационной задачи, связанной с неравенством Крафта — Макмиллана и алфавитным кодированием. Задача состоит в минимизации суммы (глубин листьев) $\sum_{i=1}^m d_i$ при условии (существования дерева с заданным набором глубин)

$$\sum_{i=1}^m 2^{-d_i} \leq 1. \quad (7)$$

Несложно видеть, что решение доставляет набор чисел d_i , которые принимают одно или два соседних целочисленных значения. Действительно, если, скажем, $d_1 \leq d_2 + 2$, то пару чисел d_1, d_2 можно заменить парой $d_1 + 1, d_2 - 1$. При этом целевая сумма не изменится, а неравенство (7) останется в силе.

В оптимальном дереве $2^l - m$ вершин расположены на глубине $l - 1$ и $2m - 2^l$ — на глубине l . Лемма 1 доказана.

⁷⁾ Имеется в виду множество $\{0, 1\}^2$ с операцией $(f, g) \circ (f', g') = (f \vee gf', gg')$.

⁸⁾ Скажем, когда сумма $a \circ b$ не зависит от одного из слагаемых.

Далее для записи интервальных префиксных сумм будем использовать сокращённое обозначение $x_{[l;r]} = x_l \circ x_{l+1} \circ \dots \circ x_r$.

Напомним, что в избыточной универсальной префиксной схеме любой элемент вычисляет сумму вида $x_{[l;r]}$. Число r назовём *индексом* элемента.

Теорема 1. Справедливо $L_2(2^n) > \frac{3}{4}(n-1)2^n$.

ДОКАЗАТЕЛЬСТВО. При $n \geq 2$ рассмотрим избыточную префиксную схему Π глубины n на 2^n входах. Схему можно условно разбить на две половины: в левой — элементы с индексами из $[1, 2^{n-1}]$, в правой — с индексами из $[2^{n-1} + 1, 2^n]$. Обозначим через $\lambda(\Pi)$ и $\rho(\Pi)$ число элементов в левой и правой половинах схемы соответственно.

Следующее наблюдение обычно для анализа параллельных префиксных схем минимальной глубины безотносительно ограничений на ветвление элементов.

Утверждение 1. $\rho(\Pi) \geq L_2(2^{n-1}) + 2^{n-1}$.

ДОКАЗАТЕЛЬСТВО. Заметим, во-первых, что в правой половине схемы на глубине n расположены 2^{n-1} элементов, вычисляющих финальные суммы $x_{[1;i]}$, $2^{n-1} < i \leq 2^n$, как $s_i^0 \circ s_i^1$, где подсумма s_i^0 вычисляется в левой половине схемы, а s_i^1 — в правой на глубине не выше $n-1$.

Оборвём рёбра, соединяющие элементы из левой половины с элементами из правой. Тогда элементы, прежде вычислявшие суммы s_i^1 , будут вычислять суммы $x_{[2^{n-1}+1;i]}$, т. е. все возможные префиксные суммы переменных $x_{2^{n-1}+1}, x_{2^{n-1}+2}, \dots, x_{2^n}$ с глубиной $n-1$. Степени ветвления элементов в правой половине схемы Π при этом не изменяются. Утверждение 1 доказано.

Теперь оценим снизу $\lambda(\Pi)$. Будем анализировать пути, соединяющие входы x_i , $1 \leq i \leq 2^{n-1}$, с выходами q_j , вычисляющими суммы $x_{[1;j]}$. Для любой пары (i, j) при $i \leq j$ в универсальной схеме Π имеется ровно один путь от x_i до q_j ⁹⁾. Если путь из x_i в q_j , где $i \leq j$, не содержит элемента глубины d , то будем говорить, что путь *пропускает уровень d* .

I. Сделаем несколько простых замечаний. Пусть $E_{[r_1, r_2]}^d$ обозначает множество расположенных на глубине d элементов схемы Π с индексами в интервале $[r_1, r_2]$.

⁹⁾ В общем случае путей может быть несколько. Доказательство проходит и в некоторых таких ситуациях, например, в случае идемпотентной некоммутативной операции. Тогда следует выбрать по одному пути для каждого выхода так, чтобы множество выбранных путей образовывало дерево (с корневой вершиной x_1), а остальные пути не принимать в расчёт.

1) Отметим, что любой путь из вершины x_i к выходам либо проходит через элемент из $E_{[i, i+2^d-1]}^d$, либо пропускает уровень d .

2) Через любой элемент (в силу ограничения на ветвление) проходит не более 2^n различных путей, соединяющих входы и выходы схемы: элемент на глубине d связан не более чем с 2^d входами и не более чем с 2^{n-d} выходами.

3) Согласно лемме 1 пути от входа x_i , где $i \leq 2^{n-1}$, к выходам (таких путей $2^n - i + 1$) пропускают суммарно не более $i - 1$ уровней (напомним, что всего в схеме n уровней).

II. Обозначим через $\lambda_d = |E_{[1, 2^{n-1}]}^d|$ число элементов в левой половине схемы П на глубине d . Согласно замечаниям из предыдущего пункта через эти элементы проходит не более $\lambda_d 2^n$ путей между входами и выходами схемы. Пути из вершин $x_1, \dots, x_{2^{n-1}-2^d+1}$ могут проходить только через эти элементы, либо пропускать уровень d . Общее число путей из указанных вершин к выходам равно

$$\pi_d = 2^n + (2^n - 1) + \dots + (2^{n-1} + 2^d) = 3(2^{2n-3} + 2^{n-2}) - (2^n - 1)2^{d-1} - 2^{2d-1}.$$

Таким образом, не менее $\sigma_d = \max\{0, \pi_d - \lambda_d 2^n\}$ из этих путей пропускают уровень d .

Оценивая при помощи леммы 1 общее число пропусков уровней путями от входов $x_1, \dots, x_{2^{n-1}-1}$ к выходам, получаем

$$\sum_{d=1}^{n-1} \sigma_d \leq \frac{1}{2}(2^{n-1} - 1)(2^{n-1} - 2) = 2^{2n-3} - 3 \cdot 2^{n-2} + 1.$$

Ввиду оценки $\lambda_d \geq 2^{-n}(\pi_d - \sigma_d)$ получаем

$$\begin{aligned} \lambda(\Pi) &\geq \sum_{d=1}^{n-1} \lambda_d \geq (3n - 4)2^{n-3} + \frac{3n}{4} - 2^{-n} - \\ &\quad - (1 - 2^{-n}) \sum_{d=1}^{n-1} 2^{d-1} - 2^{-n} \sum_{d=1}^{n-1} 2^{2d-1} = \\ &= 3n2^{n-3} - \frac{7}{6} \cdot 2^n + O(n). \quad (8) \end{aligned}$$

При помощи утверждения 1 отсюда уже извлекается оценка сложности в виде $L_2(2^n) \geq \frac{3}{4} \cdot n2^n - O(2^n)$. В следующем пункте уточним линейное слагаемое (порядка 2^n) в (8), чтобы получить чуть более точную оценку, указанную в формулировке теоремы.

III. При $n \geq 4$ оценим более аккуратно число элементов на уровнях $n - 3$, $n - 2$, $n - 1$. Запишем

$$\lambda_{n-2} = \lambda_{n-2,0} + \lambda_{n-2,1}, \quad \lambda_{n-3} = \lambda_{n-3,0} + \dots + \lambda_{n-3,3},$$

где $\lambda_{n-k,i} = |E_{[i2^{n-k}+1,(i+1)2^{n-k}]}^{n-k}|$. Напомним, что элемент на глубине $n-k$ связан путями не более чем с 2^k выходами схемы.

Все пути из x_1 к выходам проходят через элементы из $E_{[1,2^d]}^d$, этих путей 2^n , они не пропускают уровни, поэтому $\lambda_{n-1} = 2^{n-1}$, $\lambda_{n-2,0} = 2^{n-2}$, $\lambda_{n-3,0} = 2^{n-3}$.

Пути из $x_{2^{n-2}+1}$ проходят через элементы из $E_{[2^{n-2}+1,2^{n-2}+2^d]}^d$ либо пропускают соответствующие уровни, всего путей $3 \cdot 2^{n-2}$, а число пропусков не превосходит 2^{n-2} , поэтому

$$\max\{3 \cdot 2^{n-2} - 4\lambda_{n-2,1}, 0\} + \max\{3 \cdot 2^{n-2} - 8\lambda_{n-3,2}, 0\} \leq 2^{n-2},$$

откуда $\lambda_{n-2,1} + \lambda_{n-3,2} \geq 7 \cdot 2^{n-5}$.

Наконец, пути из вершины $x_{i2^{n-3}+1}$ проходят через $E_{[i2^{n-3}+1,(i+1)2^{n-3}]}^{n-3}$ или пропускают уровень $n-3$, таких путей $(8-i)2^{n-3}$, а пропускают они суммарно не более $i2^{n-3}$ уровней, поэтому получаем $\lambda_{n-3,1} \geq 3 \cdot 2^{n-5}$ и $\lambda_{n-3,3} \geq 2^{n-5}$.

Окончательно находим: $\lambda_{n-1} + \lambda_{n-2} + \lambda_{n-3} \geq 39 \cdot 2^{n-5}$. Теперь оценка (8) усиливается до следующей:

$$\begin{aligned} \lambda(\Pi) &\geq \sum_{d=1}^{n-1} \lambda_d \geq 39 \cdot 2^{n-5} + 2^{-n} \sum_{d=1}^{n-4} \pi_d - 2^{-n} \sum_{d=1}^{n-1} \sigma_d \geq \\ &\geq 39 \cdot 2^{n-5} + (3n-13)2^{n-3} + \frac{3n-9}{4} - 2^{-n} - \\ &\quad - (1-2^{-n}) \sum_{d=1}^{n-4} 2^{d-1} - 2^{-n} \sum_{d=1}^{n-4} 2^{2d-1} = \\ &= 3n2^{n-3} - \frac{181}{192} \cdot 2^{n-1} + \frac{3n}{4} - \frac{19}{16} - \frac{2^{2-n}}{3} > 3n2^{n-3} - 2^{n-1}, \end{aligned}$$

так как внутренние суммы в третьей строке равны $2^{n-4} - 1$ и $\frac{2}{3}(4^{n-4} - 1)$.

IV. Применяя утверждение 1, приходим при $n \geq 4$ к рекуррентному соотношению

$$L_2(2^n) > L_2(2^{n-1}) + \frac{3}{4}n2^{n-1}.$$

В качестве базы индукции возьмём достаточно несложно получаемую оценку¹⁰⁾ $L_2(8) \geq 13$. Теперь соотношение разрешается так, как заявлено в формулировке теоремы. Результат верен и в случае $0 \leq n \leq 3$. Теорема 1 доказана.

¹⁰⁾ Например, она следует из легко проверяемого соотношения $L_2(4) = 6$ и утверждения 1, применяемого с $n = 3$. На самом деле, рассуждение п. III вместе с утверждением 1 приводят к оценке $L_2(8) \geq 17$.

2. Некоторые верхние оценки

2.1. Напомним способ, которым строятся схемы почти минимальной глубины с ветвлением 2, дающие оценки сложности (4) и (5). Указанные конструкции основаны на базовых схемах Коге — Стоуна [8] минимальной глубины и их модификации из работы [6]. Схемы этих семейств обозначаем через K_n и S_n соответственно, где 2^n — число входов. На рис. 1 показаны схемы с 8 входами.

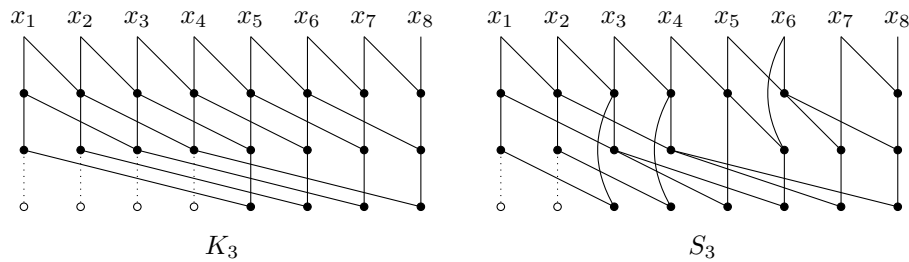


Рис. 1. Схема Коге — Стоуна (слева) и схема из [6] (справа)

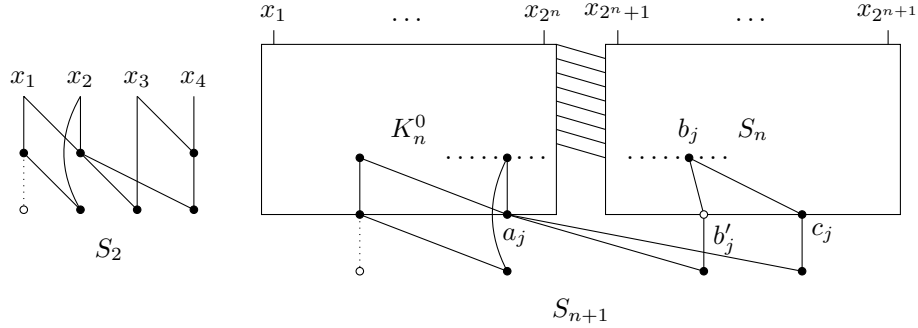
При изображении префиксных схем следуем традиции располагать входы сверху слева направо, элементы с одинаковым индексом — друг под другом, элементы одного уровня — на одной горизонтали. Элементы обозначаются чёрными кружками.

В рекурсивных построениях применяются варианты базовых схем, достроенные до схем с нулевым ветвлением выходов. На рис. 1 дополнительные элементы изображаются белыми кружками, а соединительные линии — пунктиром. Таким образом достроенные варианты схем будем обозначать через K_n^0 и S_n^0 .

Схемы Коге — Стоуна K_n имеют простую регулярную структуру. Поясним строение схемы S_n на рис. 2. Схема S_{n+1} составлена из подсхем K_n^0 и S_n , которые стыкуются на первых $n - 1$ уровнях по правилу схем Коге — Стоуна. В результате стыковки элементы b_j на уровне $n - 1$ в левой половине подсхемы S_n вычисляют суммы $x_{[2^{n-1}+j+1; 2^n+j]}$, $j = 1, \dots, 2^{n-1}$. Элемент b_j соединён с одним (c_j) или двумя¹¹⁾ (b'_j и c_j) выходами подсхемы S_n . Выход a_j подсхемы K_n^0 , в котором вычисляется сумма $x_{[1; 2^{n-1}+j]}$, и элементы b_j (либо b'_j) и c_j участвуют в вычислении пары финальных префиксных сумм. Сумма $x_{[1; 2^{n-1}+j]}$ вычисляется повторно на следующем уровне.

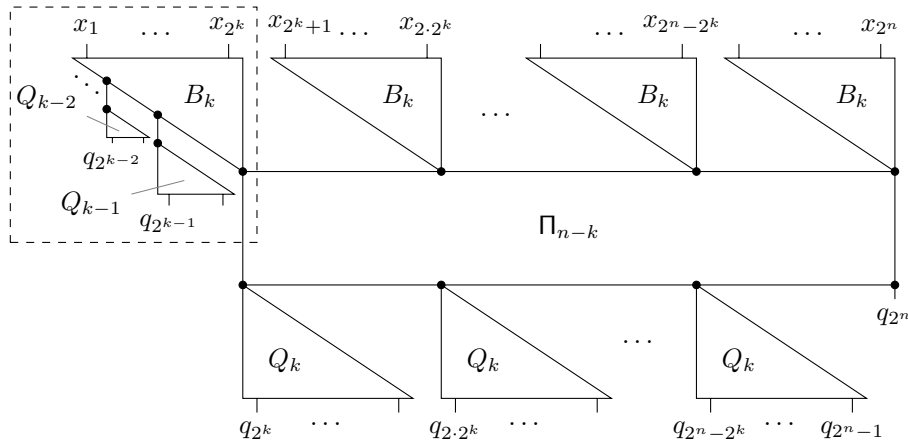
Сложность схем K_n и S_n совпадает, $L(K_n) = L(S_n) = (n - 0,5)2^n$, при этом дополненные схемы S_n^0 имеют меньшую сложность, чем схемы K_n^0 ,

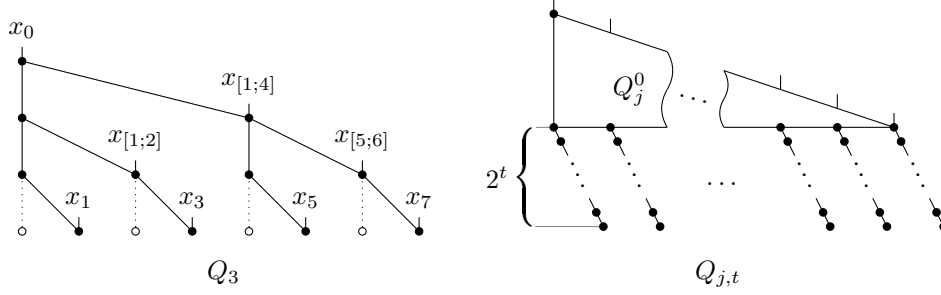
¹¹⁾ С одним при $j \leq 2^{n-2}$ и с двумя при $j > 2^{n-2}$.

Рис. 2. Рекурсивное определение схем S_n

а именно, $L(K_n^0) = n2^n$ и $L(S_n^0) = (n - 0,25)2^n$ (разница в числе тождественных элементов).

В методе из [10] (который по сути является переложением метода из [3] на схемы с ограничением ветвления) строятся схемы $\Pi_{k,n}$ глубины $n+k$ на 2^n входах, которые имеют вид, изображённый на рис. 3. Входы разбиты на группы по 2^k штук. Суммы в группах вычисляются полными бинарными деревьями B_k . Их выходы присоединены к префиксной схеме Π_{n-k} минимальной глубины. На выходах этой подсхемы реализуются все префиксные суммы с числом слагаемых, кратным 2^k . Полный комплект сумм исходных переменных вычисляется присоединёнными «корневыми» вершинами к выходам внутренней подсхемы «перевёрнутыми» бинарными деревьями Q_k , а также деревьями $Q_1, Q_2, \dots, Q_{k-1}$,

Рис. 3. Схема $\Pi_{k,n}$

Рис. 4. Конструкции (под)схем Q_j и $Q_{j,t}$

присоединёнными к внутренним элементам главной цепочки¹²⁾ первого дерева B_k . Другими входами подсхем Q_j являются переменные и промежуточные суммы, вычисляемые на первых $k-1$ уровнях. На рис. 4 слева изображён пример схемы Q_3 (с 8 выходами). На выходах q_j схемы $\Pi_{k,n}$ вычисляются суммы $x_{[1;j]}$.

На схемах $\Pi_{k,n}$ достигаются оценки сложности (4) или (5), в зависимости от того, какая базовая схема используется. Отметим, что они удовлетворяют ещё одному существенному ограничению: старшая префиксная сумма вычисляется на глубине n .

Заметим, что обведённый пунктиром (левый верхний) фрагмент схемы $\Pi_{k,n}$ имеет запас по глубине и поэтому реализован неоптимально. Чтобы получить небольшой выигрыш в сложности, вместо схем Q_j для вычисления финальных сумм в рассматриваемом фрагменте можно использовать схемы типа $Q_{j,t}$. Схема $Q_{j,t}$ функционально эквивалентна схеме Q_{j+t} . Она позволяет получить 2^{j+t} выходных сумм, но имеет глубину $j + 2^t - 1$ (см. рис. 4 (справа)). Внутренняя подсхема Q_j^0 (т. е. схема Q_j , дополненная до схемы с нулевым ветвлением выходов) вычисляет суммы с числом слагаемых, кратным 2^t . Остальные суммы вычисляются последовательно прибавлением одной переменной к уже вычисленным. Схема $Q_{j,1}$ совпадает с Q_{j+1} .

Без учёта корневой вершины схемы Q_{j+t} и $Q_{j,t}$ содержат по $2^{j+t} - 1$ бинарных элементов, но $2^{j+t-1} - 1$ и $2^j - 1$ тождественных элементов соответственно.

Таким образом, замена подсхем Q_j в выделенной части схемы $\Pi_{k,n}$ подсхемами $Q_{j-t,t}$ с максимально возможным t позволяет при $k \geq 3$ уточнить оценки (4), (5) на величину $O(2^k)$ за счёт сокращения некоторых тождественных элементов.

¹²⁾ Главная цепочка соединяет вход x_1 с выходом, в котором вычисляется старшая префиксная сумма. В данном случае элементы цепочки вычисляют суммы $x_{[1;2]}, x_{[1;4]}, \dots, x_{[1;2^{k-1}]}$.

При малых k экономия незначительна, однако, например, сложность схемы Брента — Кунга ($k = n - 1$) только при замене $Q_3, Q_4, \dots, Q_{k-1}$ на $Q_{1,2}, Q_{2,2}, \dots, Q_{k-3,2}$ уменьшается на $2^{n-3} - 1$ до $2^{\frac{3}{8}} \cdot 2^n - n - 2$ и далее ещё примерно на $0,01 \cdot 2^n$ при использовании подсхем $Q_{j,t}$ с $t \geq 3$.

Если дублировать элементы в главной цепочке (ограничение на ветвление это позволяет), то вообще все внешние подсхемы $Q_3, Q_4, \dots, Q_k$ можно заменить подсхемами $Q_{j,t}$ с $t \geq 2$, что приводит к схеме сложности менее $2,25 \cdot 2^n$, но при этом число бинарных элементов увеличивается на $n - 2$.

Отказ от (несколько искусственного) ограничения на глубину вычисления старшей префиксной суммы позволил бы перейти к ещё более экономным конструкциям. Далее сформулируем несколько общих оценок.

2.2. Как следует из (6), число L бинарных элементов в префиксной схеме с m входами, имеющей глубину D (или глубину D вычисления старшей суммы $x_{[1;m]}$), никогда не меньше чем $2m - D - 2$. Оценка достигается, если D не слишком мало, в частности в случае ограничения 2 на ветвление элементов, при

$$D \geq \lfloor \log_2 m \rfloor + \lfloor \log_2(2m/3) \rfloor = 2 \log_2 m - O(1).$$

Соответствующие конструкции схем приведены в [13, 14]. Фактически из процедуры построения следует доказательство минимальности глубины для класса оптимальных схем. Схема [9] оптимальна в слабом смысле, т. е. при дополнительном ограничении на глубину вычисления старшей суммы, и также имеет минимально возможную для оптимальной схемы глубину.

Приведём несколько простых результатов для общей сложности. Множество элементов любой префиксной схемы можно разбить на оригиналы и дубликаты. Суммы, вычисляемые *оригиналами*, попарно различны. *Дубликат* вычисляет сумму, которая либо тождественна одной из переменных, либо встречается среди сумм, вычисляемых оригиналами. Легко проверить, что число дубликатов в любой схеме с ветвлением 2 по порядку не меньше m/D . Справедлива

Лемма 2. Число дубликатов в префиксной схеме глубины D с m входами и ветвлением 2 не меньше $\frac{m-D-1}{2(D-1)}$.

Доказательство. Обозначим через U множество всех элементов избыточной схемы, в которых вычисляются суммы $x_{[1;i]}$. Через U_2 обозначим множество всех бинарных элементов в U . Заметим, что бинарные элементы вычисляют все $m - 1$ префиксных сумм $x_{[1;i]}$, кроме x_1 .

Множество U_2 естественным образом распадается на непересекающиеся ориентированные цепочки с выколотыми стартовыми вершинами: такие вершины принадлежат U , но не обязательно U_2 . Опишем подходящее

неприводимое разбиение — такое, в котором никакие две цепочки нельзя объединить в одну.

Будем строить цепочки в порядке возрастания глубины стартовых вершин до тех пор, пока множество U_2 не будет исчерпано. Формируя очередную цепочку, продолжаем её до тех пор, пока к ней нельзя будет добавить новую вершину из U_2 . Главная цепочка (она единственная) начинается от входа x_1 . Стартовой вершиной любой очередной цепочки может быть либо тождественный элемент, либо бинарный элемент ранее построенной цепочки. В первом случае поставим цепочке в соответствие стартовый тождественный элемент — можем считать его дубликатом. Во втором случае ветвление стартового элемента в схеме равно 2, поэтому он не может быть выходом, и, как следствие, для него найдётся дубликат. Среди всех дубликатов (вычисляющих одну и ту же сумму) выберем и поставим в соответствие построенной цепочке тот, который 1) не был ранее сопоставлен цепочке с бинарным стартовым элементом и 2) является либо бинарным элементом, либо выходом. Условиям 1 и 2 всегда можно удовлетворить, поскольку среди элементов, вычисляющих фиксированную сумму $x_{[1;j]}$, бинарных элементов с ветвлением 2 (т. е. тех, которые могут быть стартовыми вершинами) меньше, чем тех, которые удовлетворяют условию 2.

По построению каждый элемент-дубликат соответствует не более чем двум цепочкам разбиения. В главной цепочке вычисляется не более D выходных сумм, ей не ставятся в соответствие дубликаты. Каждая очередная цепочка добавляет не более чем $D - 1$ новых выходных сумм. Лемма 2 доказана.

Дубликаты — это цена, которую приходится платить за ограничение ветвления. Чаще всего дубликатами являются тождественные элементы¹³⁾.

Теорема 2. Соотношение $L_2(k, m) = (2 + o(1))m$ достигается

- 1) при $k = O(\log \log m)$;
- 2) на оптимальной префиксной схеме при $k = (1 + o(1)) \log_2 m$.

ДОКАЗАТЕЛЬСТВО. 1) Пусть $m = t2^n$. Разобьём входные переменные на группы по t штук и вычислим групповые суммы $y_i = x_{[(i-1)t+1; it]}$, $1 \leq i \leq 2^n$, с глубиной $h = \lceil \log_2 t \rceil$ и общей сложностью $(t - 1)2^n$.

На входах y_i построим префиксную схему глубины $n + \log_2 n$ и линейной сложности $O(2^n)$. Такая схема существует, например, согласно (5). На выходах схемы вычислены все суммы $x_{[1; it]}$, $1 \leq i \leq 2^n$.

¹³⁾ При условии, что оригиналами могут быть только бинарные элементы.

Оставшиеся суммы вычислим для каждой группы переменных при $i = 0, \dots, 2^n - 1$ последовательно как

$$x_{[1;it+1]} = x_{[1;it]} \circ x_{it+1}, \quad \dots, \quad x_{[1;it+t-1]} = x_{[1;it+t-2]} \circ x_{it+t-1},$$

используя при необходимости тождественный элемент в начале каждой цепочки¹⁴⁾. Эти вычисления выполняются с глубиной t и сложностью $t2^n$.

Общая сложность схемы равна $2m + O(2^n)$, а глубина — $n + \log_2 n + t + h$. Нужный результат получается, например, при $t \sim \log_2 n$. Конструкция легко адаптируется на случай произвольного m .

2) Этот пункт доказывается аналогично. Напомним, что оптимальность префиксной схемы заключается в выполнении следующих условий [4, 7]:

- все её бинарные элементы либо принадлежат дереву, вычисляющему старшую сумму $x_{[1;m]}$, либо вычисляют выходные суммы $x_{[1;i]}$ — по одному разу каждую;
- старшая сумма вычисляется на глубине, равной глубине схемы.

Положим $m = t(2^n + 1) + n$. На первом шаге вычисляются групповые суммы $y_i = x_{[(i-1)t+1;it]}$, $1 \leq i \leq 2^n$, с глубиной h и общей сложностью $(t-1)2^n$.

К выходам y_i присоединим схему Брента — Кунга¹⁵⁾: она имеет глубину $2n - 1$, сложность около $2,5 \cdot 2^n$, а старшую сумму $y_{[1;2^n]}$ вычисляет с глубиной n .

Как и в конструкции из п. 1, оставшиеся суммы вычислим по группам последовательно. При этом добавим в схему 2^n цепочек: каждая состоит из $t - 1$ бинарных элементов и начинается, при необходимости, с тождественного элемента. Старшие суммы $x_{[1;t2^n+1]}, \dots, x_{[1;t2^n+t+n]}$ вычисляем в цепочке длины $n + t$, присоединённой к старшему выходу подсхемы Брента — Кунга.

Таким образом, общая сложность схемы равна $2m + O(2^n)$, а глубина равна $h + 2n + t < 2\log_2 m + t$ и совпадает с глубиной вычисления старшей суммы. По построению множество бинарных элементов удовлетворяет условию оптимальности. Нужный результат получается при выборе параметра t медленно растущим, скажем, $t \in \omega(1) \cap o(n)$. Теорема 2 доказана.

Заметим, что при выборе $t = \Theta(n)$ в любой из конструкций теоремы 2 одновременно достигается минимальное по порядку число дубликатов $\Theta(m/\log m)$ для параллельной префиксной схемы (согласно лемме 2),

¹⁴⁾ Если цепочка присоединяется к элементу, имеющему ветвление 1.

¹⁵⁾ Можно использовать и схемы из [13, 14].

но ценой дополнительного увеличения глубины. В оптимальной схеме дубликаты составляют в точности множество тождественных элементов.

Финансирование работы

Исследование выполнено за счёт бюджета Научно-исследовательского института «Квант». Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Лупанов О. Б.** Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984. 138 с.
2. **Jukna S.** Boolean function complexity. Heidelberg: Springer, 2012. 618 p.
3. **Ladner R. E., Fischer M. J.** Parallel prefix computation // J. ACM. 1980. V. 27, No. 4. P. 831–838.
4. **Fich F. E.** New bounds for parallel prefix circuits // Proc. 15th ACM Annu. Symp. Theory of Computing (Boston, USA, April 25–27, 1983). New York: ACM, 1983. P. 100–109.
5. **Сергеев И. С.** О минимальных параллельных префиксных схемах // Вестн. Моск. унив. Сер. 1. Математика. Механика. 2011. № 5. С. 48–51.
6. **Sergeev I. S.** On the complexity of parallel prefix circuits // Electron. Colloq. Comput. Complex.: Tech. Rep. TR13-041. Rehovot: Weizmann Inst. Sci., 2013. 47 p.
7. **Zhu H., Cheng C.-K., Graham R.** On the construction of zero-deficiency parallel prefix circuits with minimum depth // ACM Trans. Des. Autom. Electron. Syst. 2006. V. 11, No. 2. P. 387–409.
8. **Kogge P. M., Stone H. S.** A parallel algorithm for the efficient solution of a general class of recurrence equations // IEEE Trans. Comput. 1973. V. 22, No. 8. P. 786–793.
9. **Brent R. P., Kung H. T.** A regular layout for parallel adders // IEEE Trans. Comput. 1982. V. 31, No. 3. P. 260–264.
10. **Han T., Carlson D. A.** Fast area-efficient VLSI adders // Proc. IEEE 8th Symp. Computer Arithmetic (Como, Italy, May 19–21, 1987). Los Alamitos: IEEE, 1987. P. 49–56.
11. **Snir M.** Depth-size trade-offs for parallel prefix computation // J. Algorithms. 1986. V. 4. P. 185–201.
12. **Bund J., Lenzen C., Medina M.** Optimal metastability-containing sorting via parallel prefix computation // IEEE Trans. Comput. 2020. V. 69, No. 2. P. 198–211.
13. **Lin Y.-C., Hung L.-L.** Straightforward construction of depth-size optimal, parallel prefix circuits with fan-out 2 // ACM Trans. Des. Autom. Electron. Syst. 2009. V. 14, No. 1. Paper ID 15. 13 p.

14. **Sheeran M., Parberry I.** A new approach to the design of optimal parallel prefix circuits. Tech. Rep. No. 2006:1. Göteborg: Chalmers Univ. Tech., 2006. 25 p.

Сергеев Игорь Сергеевич

Статья поступила

22 декабря 2023 г.

После доработки —

14 апреля 2024 г.

Принята к публикации

22 июня 2024 г.

ON THE COMPLEXITY OF FANOUT-BOUNDED PARALLEL PREFIX CIRCUITS

I. S. Sergeev

Scientific and Research Institute “Kvant”,
15 Chetvyortyi Likhachyovskii Lane, 125438 Moscow, Russia
E-mail: isserg@gmail.com

Abstract. We prove that the complexity of a universal depth- n parallel prefix circuit on 2^n inputs with fanout bounded by 2 is at least $0.75(n-1)2^n$. We also propose a number of simple constructions and upper complexity bounds on fanout-2 prefix circuits of depth $n+k$. Illustr. 4, bibliogr. 14.

Keywords: parallel prefix circuit, fanout-bounded circuit, circuit complexity, circuit depth.

References

1. **O. B. Lupanov**, *Asymptotic Bounds for the Complexity of Control Systems* (Izd. Mosk. Gos. Univ., Moscow, 1984) [Russian].
2. **S. Jukna**, *Boolean Function Complexity* (Springer, Heidelberg, 2012).
3. **R. E. Ladner** and **M. J. Fischer**, Parallel prefix computation, *J. ACM* **27** (4), 831–838 (1980).
4. **F. E. Fich**, New bounds for parallel prefix circuits, in *Proc. 15th ACM Annu. Symp. Theory of Computing, Boston, USA, April 25–27, 1983* (ACM, New York, 1983), pp. 100–109.
5. **I. S. Sergeev**, Minimal parallel prefix circuits, *Vestn. Mosk. Univ., Ser. 1. Mat. Mekh.*, No. 5, 48–51 (2011) [Russian] [*Mosc. Univ. Math. Bull.* **66** (5), 215–218 (2011)].
6. **I. S. Sergeev**, On the complexity of parallel prefix circuits, *Electron. Colloq. Comput. Complex* (Tech. Rep. TR13-041) (Weizmann Inst. Sci., Rehovot, 2013).
7. **H. Zhu**, **C.-K. Cheng**, and **R. Graham**, On the construction of zero-deficiency parallel prefix circuits with minimum depth, *ACM Trans. Des. Autom. Electron. Syst.* **11** (2), 387–409 (2006).

8. **P. M. Kogge** and **H. S. Stone**, A parallel algorithm for the efficient solution of a general class of recurrence equations, *IEEE Trans. Comput.* **22** (8), 786–793 (1973).
9. **R. P. Brent** and **H. T. Kung**, A regular layout for parallel adders, *IEEE Trans. Comput.* **31** (3), 260–264 (1982).
10. **T. Han** and **D. A. Carlson**, Fast area-efficient VLSI adders, in *Proc. IEEE 8th Symp. Computer Arithmetic, Como, Italy, May 19–21, 1987* (IEEE, Los Alamitos, 1987), pp. 49–56.
11. **M. Snir**, Depth-size trade-offs for parallel prefix computation, *J. Algorithms* **4**, 185–201 (1986).
12. **J. Bund**, **C. Lenzen**, and **M. Medina**, Optimal metastability-containing sorting via parallel prefix computation, *IEEE Trans. Comput.* **69** (2), 198–211 (2020).
13. **Y.-C. Lin** and **L.-L. Hung**, Straightforward construction of depth-size optimal, parallel prefix circuits with fan-out 2, *ACM Trans. Des. Autom. Electron. Syst.* **14** (1), ID 15 (2009).
14. **M. Sheeran** and **I. Parberry**, A new approach to the design of optimal parallel prefix circuits, *Tech. Rep. No. 2006:1* (Chalmers Univ. Tech., Göteborg, 2006).

Igor S. Sergeev

Received December 22, 2023

Revised April 14, 2024

Accepted June 22, 2024

МЕТОДЫ НЕГЛАДКОГО АНАЛИЗА В ЗАДАЧЕ
МИНИМИЗАЦИИ СУММЫ МОДУЛЕЙ
ОТ АФФИННЫХ ФУНКЦИЙ

Г. Ш. Тамасян^{1, 2, a}, Г. С. Шульга^{3, 2, b}

¹ Военно-космическая академия им. А. Ф. Можайского,
ул. Ждановская, 13, 197082 Санкт-Петербург, Россия

² Институт проблем машиноведения,
Большой пр., 61, В. О., 199178 Санкт-Петербург, Россия

³ Санкт-Петербургский гос. университет,
Университетская наб., 7–9, 199034 Санкт-Петербург, Россия

E-mail: ^agrigoriytamasjan@mail.ru, ^bgdextrous@gmail.com

Аннотация. Демонстрируется применение аппарата конструктивного негладкого анализа на задаче минимизации выпуклой кусочно аффинной функции, заданной в виде суммы модулей от аффинных. Для общего случая использовалось гиподифференциальное исчисление, в скалярном — субдифференциальное. Из анализа критерия оптимальности получено, что точку, доставляющую глобальный минимум, можно найти, решая соответствующую задачу линейного программирования. В скалярном же случае решением задачи является взвешенная медиана узлов ломаной. Библиогр. 30.

Ключевые слова: кусочно аффинная функция, ломаная, метод наименьших модулей, субдифференциал, гиподифференциал, взвешенная медиана.

Введение

Пусть заданы векторы $a_k \in \mathbb{R}^n$ и числа $b_k \in \mathbb{R}$, $k \in 1 : s$. Рассмотрим задачу поиска безусловного минимума

$$\sum_{k=1}^s |x^\top a_k - b_k| \rightarrow \min_{x \in \mathbb{R}^n}. \quad (1)$$

К проблеме минимизации суммы модулей от аффинных функций сводятся, например, такие прикладные задачи, как машинное обучение, линейный регрессионный анализ и обработка измерительной информации

с аномально большими ошибками (сбоями) [1–4], к которой метод наименьших модулей менее чувствителен, чем метод наименьших квадратов [1, 5–7]. К решению задачи минимизации выпуклой кусочно аффинной функции может быть сведена проблема разрешимости интервальной системы линейных алгебраических уравнений (см. [8, 9]). Заметим, что и задачи с линейными ограничениями (типа равенств и/или неравенств) можно свести к задаче (1). Для этого надо воспользоваться аппаратом точных штрафных функций (см. [10–12]).

В работе проводится исследование экстремальной задачи (1) с задействованием аппарата конструктивного негладкого анализа [13–17]. Так как целевая функция выпуклая, для общего случая используется гиподифференциальное исчисление, а для скалярного — субдифференциальное.

В разд. 1 приводятся вспомогательные сведения из негладкого анализа. В разд. 2–4 дана формальная постановка задачи и её решение в общем случае. В разд. 5 анализируется скалярный случай.

1. Вспомогательные сведения

1.1. Элементы негладкого анализа. Приведём некоторые сведения из конструктивного негладкого анализа [15–17]. Начнём с основного понятия, связывающего классический (гладкий) анализ с недифференцируемой оптимизацией — производной по направлению.

Пусть функция f определена на открытом множестве $U \subset \mathbb{R}^n$. Положим $S = \{g \in \mathbb{R}^n \mid \|g\| = 1\}$.

Определение 1. Говорят, что функция f дифференцируема в точке $x_0 \in U$ по направлению $g \in S$, если существует конечный предел

$$f'(x_0, g) = \lim_{\alpha \rightarrow +0} \frac{f(x_0 + \alpha g) - f(x_0)}{\alpha}.$$

Величина $f'(x_0, g)$ называется производной функции f в точке x_0 по направлению g .

Определение 2. Говорят, что функция f , заданная и конечная на U , субдифференцируема в точке $x_0 \in U$, если её приращение допускает представление

$$f(x_0 + \Delta) - f(x_0) = \max_{v \in \partial f(x_0)} v^\top \Delta + o(\|\Delta\|), \quad (2)$$

где $\partial f(x_0) \subset \mathbb{R}^n$ — выпуклый компакт, $\frac{o(\|\Delta\|)}{\|\Delta\|} \rightarrow 0$ при $\Delta \rightarrow 0$. Множество $\partial f(x_0)$ называется субдифференциалом функции f в точке x_0 .

Из (2) следует, что у субдифференцируемой в точке x_0 функции $f(x)$ существуют производные по всем направлениям и справедлива формула

$$f'(x_0, g) = \max_{v \in \partial f(x_0)} v^\top g, \quad g \in S.$$

Нетрудно убедиться, что для дифференцируемой в точке x_0 функции $f(x)$ субдифференциал $\partial f(x_0)$ состоит из одного элемента $\{f'(x_0)\}$.

Определение 3. Говорят, что функция f , заданная и конечная на U , *гиподифференцируема* в точке $x \in U$, если существует выпуклый компакт $\mathbf{d}f(x) \subset \mathbb{R} \times \mathbb{R}^n$ такой, что справедливы разложение в точке x вида

$$f(x + \Delta) = f(x) + \max_{(\alpha, v) \in \mathbf{d}f(x)} [\alpha + v^\top \Delta] + o_x(\Delta),$$

и равенство

$$\max_{(\alpha, v) \in \mathbf{d}f(x)} \alpha = 0.$$

Здесь $\alpha \in \mathbb{R}$, $v \in \mathbb{R}^n$, $\frac{o_x(\gamma\Delta)}{\gamma} \rightarrow 0$ при $\gamma \rightarrow 0$ для всех $\Delta \in \mathbb{R}^n$.

Множество $\mathbf{d}f(x)$ называется *гиподифференциалом* функции f в точке x . Элементы множества $\mathbf{d}f(x)$ называются *гипогradientами*.

Определение 4. Функция f называется *непрерывно гиподифференцируемой* в точке $x \in U$, если она гиподифференцируема в каждой точке некоторой окрестности $U_x \subset U$ точки x и существует гиподифференциальное отображение $U_x \ni z \mapsto \mathbf{d}f(z)$, непрерывное по Хаусдорфу в точке x .

Классы субдифференцируемых и гиподифференцируемых функций совпадают. Более того, справедливо соотношение

$$\partial f(x_0) = \{v \in \mathbb{R}^n \mid \begin{pmatrix} 0 \\ v \end{pmatrix} \in \mathbf{d}f(x_0)\}.$$

Однако, в отличие от субдифференциала, непрерывные гиподифференциалы наиболее интересны с практической точки зрения в конструировании численных методов оптимизации. В частности, последние обладают непрерывным направлением спуска (подъёма), аналогично градиентным методам в гладком случае.

Приведём основные формулы исчисления непрерывно гиподифференцируемых функций.

1) Пусть f_1 и f_2 непрерывно гиподифференцируемы в точке x . Тогда сумма этих функций непрерывно гиподифференцируема в x , при этом

$$\mathbf{d}(f_1 + f_2)(x) = \mathbf{d}f_1(x) + \mathbf{d}f_2(x). \quad (3)$$

2) Пусть f непрерывно гиподифференцируема в точке x . Тогда при $\lambda \geq 0$ функция λf также непрерывно гиподифференцируема в x , причём

$$\mathbf{d}(\lambda f)(x) = \lambda \mathbf{d}f(x). \quad (4)$$

3) Пусть функции $f_i(x)$, $i \in 1:s$, непрерывно гиподифференцируемы в точке x . Тогда $f(x) = \max_{i \in 1:s} f_i(x)$ непрерывно гиподифференцируема в x , при этом гиподифференциал $df(x)$ описывается следующим образом:

$$df(x) = \text{conv} \left\{ \begin{pmatrix} f_i(x) - f(x) \\ \mathbf{0}_n \end{pmatrix} + df_i(x) \mid 1 \leq i \leq s \right\}. \quad (5)$$

Здесь применяются операции, определённые над гиподифференциалами, а именно *сложение* по Минковскому и *умножение* на константу:

$$\begin{aligned} W + V &= \{w + v \mid w \in W, v \in V\}, \\ \lambda V &= \{\lambda v \mid v \in V\}, \end{aligned}$$

где W, V — выпуклые компакты в $\mathbb{R}^n$, λ — вещественное число.

Таким образом, класс непрерывно гиподифференцируемых функций довольно богат. Он содержит гладкие функции и замкнут относительно операций сложения, умножения на положительную константу и взятия поточечного максимума от непрерывно гиподифференцируемых функций. В частности, ему принадлежат выпуклые функции, как гладкие, так и негладкие.

Пример 1. Для произвольной непрерывно дифференцируемой на $\mathbb{R}^n$ функции f в качестве непрерывного гиподифференциала можно взять

$$df(x) = \left\{ \begin{pmatrix} 0 \\ f'(x) \end{pmatrix} \right\}.$$

Пример 2. Пусть заданы $a \in \mathbb{R}^n$, $b \in \mathbb{R}$. Найдём гиподифференциал функции $f(x) = |x^\top a - b|$. Заметим, что функцию $f(x)$ можно переписать следующим образом:

$$f(x) = \max\{x^\top a - b, -(x^\top a - b)\}.$$

Тогда, используя формулу (5) и предыдущий пример, получим

$$df(x) = \text{conv} \left\{ \begin{pmatrix} x^\top a - b - |x^\top a - b| \\ a \end{pmatrix}, \begin{pmatrix} -(x^\top a - b) - |x^\top a - b| \\ -a \end{pmatrix} \right\}.$$

Теорема 1. Для того чтобы точка $x_* \in \mathbb{R}^n$ была точкой минимума функции $f(x)$ на $\mathbb{R}^n$, необходимо, а в случае выпуклости f и достаточно, чтобы для любого $g \in S$ выполнялось неравенство

$$f'(x_*, g) \geq 0. \quad (6)$$

Точка x_* , для которой выполняется неравенство (6), называется *стационарной* точкой функции $f(x)$ на $\mathbb{R}^n$.

Теорема 2. *Условие стационарности равносильно включению*

$$\mathbf{0}_{n+1} \in \mathbf{d}f(x_*).$$

Замечание 1. Субдифференциальное исчисление и условие минимума формально имеет такой же вид, что и для гиподифференцируемых функций (см. (3)–(5)). Для этого в формулах необходимо заменить знак $\mathbf{d}$ на ∂ и учесть, что для функции $f(x) = \max_{i \in 1:s} f_i(x)$ субдифференциал $\partial f(x_0)$ вычисляется следующим образом:

$$\partial f(x_0) = \text{conv} \bigcup_{i \in I(x_0)} \partial f_i(x_0),$$

где $I(x_0) = \{i \in 1:s \mid f_i(x_0) = f(x_0)\}$.

1.2. Кусочно аффинная функция. В [18, п. 4.2], в частности, доказано утверждение, что непрерывная кусочно аффинная функция представима в виде разности двух выпуклых кусочно аффинных функций. В [19–22] показано, что любая кусочно аффинная функция допускает аналитическое представление в виде суммы максимума и минимума от конечных семейств аффинных функций. В [23] с использованием аппарата кодифференциального исчисления реализованы алгоритмы представления кусочно аффинной функции в указанном выше виде.

Покажем, что целевая функция задачи (1) является выпуклой кусочно аффинной функцией, а следовательно, гиподифференцируема. Для начала напомним несколько свойств, связанных с функцией максимума:

$$|g(x)| = \max\{-g(x), g(x)\}, \quad (7)$$

$$h(x) + \max_{k \in 1:m} g_k(x) = \max_{k \in 1:m} \{h(x) + g_k(x)\}, \quad (8)$$

$$|g(x)| + |h(x)| = \max\{-g(x) - h(x), g(x) - h(x), -g(x) + h(x), g(x) + h(x)\}. \quad (9)$$

Применяя необходимое число раз равенство (9), сумму модулей можно представить в виде максимума от 2^s аффинных функций. В общем случае большинство функций под максимумом неактивные и их можно отбросить (см. [23, 24]). Например, в скалярном случае их число равно $s + 1$. Выпуклость целевой функции следует из того, что поточечный максимум от выпуклых функций образует выпуклую функцию.

2. Задача

Используя аппарат гиподифференциального исчисления, исследовать задачу поиска безусловного минимума

$$f(x) := \sum_{j=1}^s |f_j(x)| \rightarrow \min_{x \in \mathbb{R}^n}, \quad (10)$$

где $f_j(x) = x^\top a_j - b_j$, $j \in 1 : s$.

Как было ранее замечено, целевая функция $f(x)$ — непрерывная выпуклая кусочно аффинная функция на всём пространстве $\mathbb{R}^n$. Более того, функция $f(x)$ ограничена снизу нулём, а минимум в задаче (10) достигается, в общем случае, не в единственной точке.

3. Гиподифференциал целевой функции

Гиподифференциал целевой функции складывается из гиподифференциалов $d|f_j(x)|$, $j \in 1 : s$. Используя результаты примера 2, имеем

$$d|f_j(x)| = \text{conv} \left\{ \begin{pmatrix} f_j(x) - |f_j(x)| \\ a_j \end{pmatrix}, \begin{pmatrix} -f_j(x) - |f_j(x)| \\ -a_j \end{pmatrix} \right\}.$$

Тогда искомый гиподифференциал функции $f(x)$ принимает вид

$$df(x) = \sum_{j=1}^s d|f_j(x)| = \sum_{j=1}^s \text{conv} \left\{ \begin{pmatrix} f_j(x) \\ a_j \end{pmatrix}, \begin{pmatrix} -f_j(x) \\ -a_j \end{pmatrix} \right\} - \begin{pmatrix} f(x) \\ \mathbf{0}_n \end{pmatrix}.$$

Запишем общий вид гипогрadients $G \in df(x)$. Заметим, что любой элемент $G \in df(x)$ можно представить в виде

$$G = \sum_{j=1}^s G_j - \begin{pmatrix} f(x) \\ \mathbf{0}_n \end{pmatrix},$$

где G_j — соответствующие элементы из $d|f_j(x)|$. Более того, имеем

$$G_j(\lambda_j) = \lambda_j \begin{pmatrix} f_j(x) \\ a_j \end{pmatrix} + (1 - \lambda_j) \begin{pmatrix} -f_j(x) \\ -a_j \end{pmatrix} = (2\lambda_j - 1) \begin{pmatrix} f_j(x) \\ a_j \end{pmatrix},$$

где $\lambda_j \in [0, 1]$, $j \in 1 : s$.

Положим $\mu_j = 2\lambda_j - 1$, $j \in 1 : s$. Тогда получим общий вид гипогрadients $G \in df(x)$:

$$G(\mu_1, \dots, \mu_s) = \begin{pmatrix} \sum_{j=1}^s \mu_j f_j(x) - f(x) \\ \sum_{j=1}^s \mu_j a_j \end{pmatrix}, \quad (11)$$

где $\mu_j \in [-1, 1]$, $j \in 1 : s$. Таким образом, гиподифференциал целевой функции представим в виде

$$\mathrm{d}f(x) = \operatorname{conv}\{G(\mu_1, \dots, \mu_s) \mid \mu_j \in [-1, 1], j \in 1 : s\}.$$

4. Критерий оптимальности

Обозначим $A = [a_1, a_2, \dots, a_s]$, $b = (b_1, b_2, \dots, b_s)$, $\mu = (\mu_1, \mu_2, \dots, \mu_s)^\top$. Тогда (11) примет вид

$$G(\mu) = \begin{pmatrix} (x^\top A - b)\mu - f(x) \\ A\mu \end{pmatrix}. \quad (12)$$

Утверждение 1. Для того чтобы точка $x_* \in \mathbb{R}^n$ доставляла глобальный минимум в задаче (10), необходимо и достаточно, чтобы нашёлся вектор $\mu^* \in \mathbb{R}^s$ такой, что

$$\begin{cases} -b\mu^* = f(x_*), \\ A\mu^* = \mathbf{0}_n, \\ \mu_j^* \in [-1, 1], \quad j \in 1 : s. \end{cases} \quad (13)$$

Доказательство. В силу выпуклости целевой функции $f(x)$ достаточно показать, что условие (13) равносильно включению (см. теорему 2)

$$\mathbf{0}_{n+1} \in \mathrm{d}f(x_*). \quad (14)$$

В самом деле, в силу общего вида гипогрadients (12) включение (14) означает, что найдётся такой вектор $\mu^* \in \mathbb{R}^s$ с компонентами $\mu_j^* \in [-1, 1]$, $j \in 1 : s$, для которого $G(\mu^*) = \mathbf{0}_{n+1}$, т. е.

$$\begin{cases} (x_*^\top A - b)\mu^* - f(x_*) = 0, \\ A\mu^* = \mathbf{0}_n. \end{cases}$$

Отсюда получаем систему (13), если упростить первое уравнение, воспользовавшись вторым условием. Утверждение 1 доказано.

Рассмотрим задачу линейного программирования

$$\begin{aligned} -b\mu &\rightarrow \max, \\ A\mu &= \mathbf{0}_n, \\ \mu_j &\in [-1, 1], \quad j \in 1 : s. \end{aligned} \quad (15)$$

Она имеет решение, так как множество планов (точек, удовлетворяющих ограничениям задачи) образует непустой выпуклый компакт. Обозначим произвольное решение задачи (15) через $\hat{\mu}$.

Следствие 1. Множество решений задачи (10) совпадает с множеством решений уравнения

$$\sum_{j=1}^s |x^\top a_j - b_j| = -b\hat{\mu}. \quad (16)$$

Утверждение 2. Пусть $\hat{\mu}$ — некоторое решение задачи (15). Всё множество решений задачи (10) описывается линейной системой

$$\begin{cases} x^\top a_j - b_j \leq 0 & \text{при } j \in I_-(\hat{\mu}), \\ x^\top a_j - b_j = 0 & \text{при } j \in I_0(\hat{\mu}), \\ x^\top a_j - b_j \geq 0 & \text{при } j \in I_+(\hat{\mu}). \end{cases} \quad (17)$$

Здесь $I_-(\hat{\mu})$, $I_0(\hat{\mu})$, $I_+(\hat{\mu})$ — индексные множества, определяемые следующим образом:

$$\begin{aligned} I_-(\hat{\mu}) &= \{j \in 1:s \mid \hat{\mu}_j = -1\}, & I_+(\hat{\mu}) &= \{j \in 1:s \mid \hat{\mu}_j = 1\}, \\ I_0(\hat{\mu}) &= \{j \in 1:s \mid \hat{\mu}_j \in (-1, 1)\}. \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Так как $\hat{\mu}$ — оптимальный план задачи (15), то

$$A\hat{\mu} = \mathbf{0}_n.$$

Умножив обе части этой системы на $x^\top$ слева и прибавив к (16), получим

$$\sum_{j=1}^s |x^\top a_j - b_j| = (x^\top A - b)\hat{\mu}.$$

Отсюда следует, что

$$\sum_{j=1}^s |x^\top a_j - b_j| = \sum_{j=1}^s (x^\top a_j - b_j)\hat{\mu}_j,$$

поэтому

$$\begin{cases} x^\top a_j - b_j \leq 0 & \text{при } \hat{\mu}_j = -1, \\ x^\top a_j - b_j = 0 & \text{при } |\hat{\mu}_j| < 1, \\ x^\top a_j - b_j \geq 0 & \text{при } \hat{\mu}_j = 1. \end{cases}$$

Утверждение 2 доказано.

Замечание 2. Чтобы найти (частное) решение x_* задачи (10), можно поступить так: последовательно решить две задачи линейного программирования, а именно: сперва задачу (15), а после — с нулевой целевой функцией при ограничениях (17).

Замечание 3. В силу теории двойственности [25–27] решения задач (15) и (10) тесно взаимосвязаны. Указанную связь используют современные методы решения задач линейного программирования, например,

двойственный симплекс-метод и метод внутренней точки. Эти методы определяют одновременно решения прямой и двойственной задач, т. е. достаточно решить подобным методом только *одну* задачу (15), и на выходе будет пара $(\hat{\mu}, x_*)$ — решение исходной задачи (10), где x_* — двойственные переменные.

5. Скалярный случай

Рассмотрим задачу минимизации непрерывной выпуклой ломаной (в форме Бернштейна)

$$f(x) = \sum_{k=1}^s a_k |x - t_k| \rightarrow \min_{x \in \mathbb{R}}, \quad (18)$$

где $a_k > 0$, $k \in 1 : s$, $t_1, t_2, \dots, t_s$ — попарно различные узлы ломаной.

Применяя аппарат субдифференциального исчисления, покажем, что решением задачи (18) является взвешенная медиана. Заметим, что поиск взвешенной медианы занимает линейное время [28]. Другой подход к решению задачи (18) рассмотрен в [29].

Определение 5 [30, с. 255]. Пусть имеется конечный набор различных вещественных чисел $b_1, b_2, \dots, b_s$, для которых заданы положительные веса $w_1, w_2, \dots, w_s$. *Взвешенной (нижней) медианой* называется элемент b_ℓ такой, что справедливы неравенства

$$\sum_{k: b_k < b_\ell} w_k < \frac{A}{2}, \quad \sum_{k: b_k > b_\ell} w_k \leq \frac{A}{2}$$

для $A = w_1 + w_2 + \dots + w_s$.

Упорядочим пары (t_k, a_k) , $k = 1 : s$, по возрастанию первой компоненты. Получим последовательность

$$(b_1, w_1), (b_2, w_2), \dots, (b_s, w_s), \quad b_1 < b_2 < \dots < b_s.$$

Целевая функция $f(x)$ состоит из $s + 1$ звеньев, а их угловые коэффициенты можно вычислить по формулам

$$p_\ell = \sum_{j=1}^{\ell} w_j - \sum_{i=\ell+1}^s w_i, \quad \ell \in 0 : s. \quad (19)$$

Отметим, что точка минимума (если она не единственная, то крайняя слева на вещественной оси) характеризуется изменением угла наклона звена ломаной с отрицательного значения на неотрицательное, т. е. $p_j < 0$, а $p_{j+1} \geq 0$. Если же угловой коэффициент звена ломаной изменяется с отрицательного на положительное значение, то глобальный минимум единственный.

Обозначим через $\ell^* \in 1 : s$ индекс взвешенной медианы для набора точек $b_1, b_2, \dots, b_s$ с весами $w_1, w_2, \dots, w_s$, т. е. b_{ℓ^*} — взвешенная медиана и для набора узлов $t_1, t_2, \dots, t_s$ с весами $a_1, a_2, \dots, a_s$.

Утверждение 3. *Взвешенная медиана набора узлов $t_1, t_2, \dots, t_s$ с весами $a_1, a_2, \dots, a_s$ доставляет глобальный минимум в задаче (18).*

ДОКАЗАТЕЛЬСТВО. Для доказательства утверждения достаточно показать, что на индексе $\ell^* \in 1 : s$ (взвешенной медианы) выполняются неравенства:

$$p_{\ell^*-1} = \sum_{j=0}^{\ell^*-1} w_j - \sum_{i=\ell^*}^{s+1} w_i < 0, \quad (20)$$

$$p_{\ell^*} = \sum_{j=0}^{\ell^*} w_j - \sum_{i=\ell^*+1}^{s+1} w_i \geq 0. \quad (21)$$

Вычислим субдифференциал целевой функции:

$$\partial f(x) = \sum_{j=1}^s \partial f_j(x),$$

где

$$\partial f_j(x) = \begin{cases} \{w_j\}, & \text{если } x > b_j, \\ \text{conv}\{-w_j, w_j\}, & \text{если } x = b_j, \\ \{-w_j\}, & \text{если } x < b_j. \end{cases}$$

Введём индексные множества

$$N_+(x) = \{j \in 1 : s \mid b_j < x\}, \quad N_-(x) = \{j \in 1 : s \mid b_j > x\}, \\ N_0(x) = \{j \in 1 : s \mid b_j = x\}.$$

Очевидно, что при наших предположениях $|N_-(x)| + |N_+(x)| + |N_0(x)| = s$, причём множество $N_0(x)$ либо пустое, либо одноэлементное. Положим $S = \sum_{j \in N_+(x)} w_j - \sum_{j \in N_-(x)} w_j$. Тогда

$$\partial f(x) = \begin{cases} \{S\}, & \text{если } N_0(x) = \emptyset, \\ \{S\} + \text{conv}\{-w_k, w_k\}, & \text{если } N_0(x) = \{k\}. \end{cases} \quad (22)$$

Разберём подробно оба случая.

СЛУЧАЙ 1: $N_0(x) = \emptyset$. Значит, точка x не совпадает ни с одним из узлов ломаной, а тогда критерий оптимальности $0 \in \partial f(x)$ будет выполнен, если $S = 0$, т. е.

$$\sum_{j \in N_+(x)} w_j = \sum_{j \in N_-(x)} w_j. \quad (23)$$

Определим индекс $\ell_* \in N_+$ такой, что

$$b_{\ell_*} = \max_{j \in N_+(x)} b_j.$$

Ясно, что тогда $b_{\ell_*+1} = \min_{j \in N_-(x)} b_j$. В итоге в силу равенства (23) имеем выполнение неравенств (20) и (21) только для точек $x \in (b_{\ell_*}, b_{\ell_*+1})$. Это означает, что каждая точка x из интервала $(b_{\ell_*}, b_{\ell_*+1})$ является взвешенной медианой, доставляющей глобальный минимум целевой функции.

СЛУЧАЙ 2: $N_0(x) = \{k\}$. Тогда критерий оптимальности будет выполнен, если

$$0 \in \text{conv}\{S - w_k, S + w_k\}.$$

В силу положительности w_j , $j \in 1 : s$, данное включение равносильно одной из следующих систем:

$$\begin{cases} \sum_{j \in N_+(x)} w_j - \sum_{j \in N_-(x)} w_j - w_k < 0, \\ \sum_{j \in N_+(x)} w_j - \sum_{j \in N_-(x)} w_j + w_k \geq 0 \end{cases} \quad (24)$$

или

$$\begin{cases} \sum_{j \in N_+(x)} w_j - \sum_{j \in N_-(x)} w_j - w_k \leq 0, \\ \sum_{j \in N_+(x)} w_j - \sum_{j \in N_-(x)} w_j + w_k > 0. \end{cases} \quad (25)$$

Выражения в левых частях этих систем суть не что иное, как угловые коэффициенты ломаной (см. (19)), которые в узле b_k изменили знаки. Точнее, ввиду (24) взвешенная нижняя медиана b_{ℓ_*} совпадает с b_k , а в силу (25) имеем

$$b_{\ell_*} = \begin{cases} b_k, & \text{если } S < w_k, \\ b_{k-1}, & \text{если } S = w_k. \end{cases}$$

Таким образом, глобальный минимум целевой функции достигается в единственной точке b_k , если $|S| \neq w_k$, иначе на отрезке $[b_{\ell_*}, b_{\ell_*+1}]$ при $S + w_k = 0$ или на отрезке $[b_{\ell_*-1}, b_{\ell_*}]$ при $S - w_k = 0$. Утверждение 3 доказано.

Следствие 2. Пусть t_* — взвешенная медиана для набора точек $t_1, t_2, \dots, t_s \in \mathbb{R}$ с весами $a_1, a_2, \dots, a_s \in \mathbb{R}$. Тогда угловой коэффициент звена ломаной, располагающегося справа от взвешенной медианы, вычисляется по формуле

$$p_* = \sum_{k: t_k \leq t_*} w_k - \sum_{k: t_k > t_*} w_k.$$

Если $p_* > 0$, то t_* — единственный глобальный минимум в задаче (18). Если $p_* = 0$, то глобальный минимум достигается на точках из отрезка $[t_*, \hat{t}]$, где $\hat{t} = \min_{t_j > t_*} t_j$.

Финансирование работы

Результаты разд. 3–5 получены в Институте проблем машиноведения РАН за счёт Российского научного фонда (проект № 23–41–00060). В остальном исследование выполнено за счёт бюджетов организаций, указанных авторами на первой странице статьи. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Мудров В. И., Кушко В. Л. Методы обработки измерений: квазиправдоподобные оценки. М.: Радио и связь, 1983. 304 с.
2. Зуховицкий С. И. О приближении несовместной системы линейных уравнений по принципу минимизации суммы модулей всех уклонов // Докл. АН СССР. 1962. Т. 143, № 5. С. 1030–1033.
3. Акимов П. А., Матасов А. И. Итерационный алгоритм для l_1 -аппроксимации в динамических задачах оценивания // Автоматика и телемеханика. 2015. № 5. С. 7–26.
4. Лакеев А. В., Носков С. И. Метод наименьших модулей для линейной регрессии: число нулевых ошибок аппроксимации // Современные технологии. Системный анализ. Моделирование. 2012. № 2. С. 48–50.
5. Горелик В. А., Трёмбачева О. С. Решение задачи линейной регрессии с использованием методов матричной коррекции в метрике l_1 // Журн. вычисл. математики и мат. физики. 2016. Т. 56, № 2. С. 202–207.
6. Суринов В. А., Тырсин А. Н. Применение обобщенного метода наименьших модулей в задачах обработки и анализа изображений // Вестн. Астрахан. гос. техн. ун-та. Сер. Управление, вычисл. техника и информатика. 2020. № 2. С. 45–55. DOI: 10.24143/2072-9502-2020-2-45-55.
7. Тамасян Г. Ш. О диких точках // Семинар по оптимизации, машинному обучению и искусственному интеллекту. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2023. 5 с. URL: oml.cmlaboratory.com/rep23.shtml (дата обращения: 27.10.2024).
8. Шарый С. П. Сильная согласованность в задаче восстановления зависимостей при интервальной неопределенности данных // Вычисл. технологии. 2017. Т. 22, № 2. С. 150–172.
9. Шарый С. П., Шарая И. А. Распознавание разрешимости интервальных уравнений и его приложения к анализу данных // Вычисл. технологии. 2013. Т. 18, № 3. С. 80–109.

10. **Ерёмин И. И.** Метод «штрафов» в выпуклом программировании // Докл. АН СССР. 1967. Т. 173, № 4. С. 748–751.
11. **Долгополик М. В.** Основы теории точных штрафных функций // Семинар по конструктивному негладкому анализу и недифференцируемой оптимизации. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2015. 17 с. URL: cnsa.cmlaboratory.com/rep15.shtml (дата обращения: 27.10.2024).
12. **Полякова Л. Н.** Метод точных штрафов: другой подход // Семинар по конструктивному негладкому анализу и недифференцируемой оптимизации. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2014. 16 с. URL: cnsa.cmlaboratory.com/rep14.shtml (дата обращения: 27.10.2024).
13. **Демьянов В. Ф., Малозёмов В. Н.** Введение в минимакс. М.: Наука, 1972. 368 с.
14. **Демьянов В. Ф., Васильев Л. В.** Недифференцируемая оптимизация. М.: Наука, 1981. 384 с.
15. **Демьянов В. Ф., Рубинов А. М.** Основы негладкого анализа и квазидифференциальное исчисление. М.: Наука, 1990. 432 с.
16. **Полякова Л. Н.** Непрерывные методы безусловной минимизации гиподифференцируемых функций // Вестн. С.-Петербург. ун-та. Сер. 10. 2007. Вып. 3. С. 71–81.
17. **Долгополик М. В.** О непрерывной кодифференцируемости // Семинар по конструктивному негладкому анализу и недифференцируемой оптимизации. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2019. 19 с. URL: cnsa.cmlaboratory.com/rep19.shtml (дата обращения: 27.10.2024).
18. **Залгаллер В. А.** О представлении функций нескольких переменных разностью выпуклых функций // Зап. науч. сем. ПОМИ. 1997. Т. 246. С. 36–65.
19. **Melzer D.** On the expressibility of piecewise linear continuous functions as the difference of two piecewise linear convex functions // Quasidifferential calculus. Amsterdam: North-Holland, 1986. P. 118–134. (Math. Program. Stud.; V. 29).
20. **Kripfganz A., Schulze R.** Piecewise affine functions as a difference of two convex functions // Optimization. 1987. V. 18, No. 1. P. 23–29. DOI: 10.1080/02331938708843210.
21. **Gorokhovich V. V., Zorko O. I., Birkhoff G.** Piecewise affine functions and polyhedral sets // Optimization. 1994. V. 31, No. 3. P. 209–221. DOI: 10.1080/02331939408844018.
22. **Gorokhovich V. V.** Geometrical and analytical characteristic properties of piecewise affine mappings. Ithaca, NY: Cornell Univ., 2011. 12 p. (Cornell Univ. Libr. e-Print Archive; arXiv:1111.1389). DOI: 10.48550/arXiv.1111.1389.
23. **Ангелов Т. А.** Представление кусочно аффинных функций в виде разности полиэдральных // Вестн. С.-Петербург. ун-та. Сер. 10. 2016. Вып. 1. С. 4–18.
24. **Тамасян Г. Ш.** О компактном представлении кодифференциала непрерывной ломаной, заданной в форме Бернштейна // Семинар по оптимизации, машинному обучению и искусственному интеллекту. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2023. 10 с. URL: oml.cmlaboratory.com/rep23.shtml (дата обращения: 27.10.2024).

25. Гавурин М. К., Малозёмов В. Н. Экстремальные задачи с линейными ограничениями. Л.: Изд-во ЛГУ, 1984. 176 с.
26. Васильев Ф. П., Иваницкий А. Ю. Линейное программирование. М.: МЦНМО, 2020. 416 с.
27. Тамасян Г. Ш., Шульга Г. С. К вопросу о минимизации выпуклой кусочно аффинной функции // Семинар по оптимизации, машинному обучению и искусственному интеллекту. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2022. 5 с. URL: oml.cmlaboratory.com/refs22.shtml (дата обращения: 27.10.2024).
28. Малистов А. С. О поиске медианы массива за линейное время // Мат. просвещение. Сер. 3. Вып. 21. М.: МЦНМО, 2017. С. 265–270.
29. Тамасян Г. Ш., Шульга Г. С. Быстрый алгоритм минимизации выпуклой ломаной // Семинар по конструктивному негладкому анализу и недифференцируемой оптимизации. Избр. докл. С.-Пб.: С.-Петербург. гос. ун-т, 2021. 6 с. URL: cnsa.cmlaboratory.com/refs21.shtml (дата обращения: 27.10.2024).
30. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ. М.: Вильямс, 2013. 1328 с.

Тамасян Григорий Шаликович
Шульга Георгий Сергеевич

Статья поступила
26 марта 2024 г.
После доработки —
20 апреля 2024 г.
Принята к публикации
22 июня 2024 г.

METHODS OF NONSMOOTH ANALYSIS
IN APPLICATION TO THE PROBLEM OF MINIMIZING
THE SUM OF AFFINE FUNCTIONS' MODULESG. Sh. Tamasyan^{1,2, a} and G. S. Shulga^{3,2, b}¹ Mozhaiskiy Space Military Academy,
13 Zhdanovskaya Street, 197082 St. Petersburg, Russia² Institute of Problems of Mechanical Engineering
61 Bolshoi Avenue, V. O., 199178 St. Petersburg, Russia³ St. Petersburg State University,
7–9 Universitetskaya Embankment, 199034 St. Petersburg, RussiaE-mail: ^agrigoriytamasjan@mail.ru, ^bgdextrous@gmail.com

Abstract. An application of constructive nonsmooth analysis methods to the problem of minimizing a convex piecewise affine function defined as the sum of absolute values of affine functions is demonstrated. Hypodifferential calculus was used in the general (multidimensional) case, while subdifferential calculus was employed in the scalar case. Analyzing the optimality criterion, one can reveal that the point delivering the global minimum can be found by solving the corresponding linear programming problem. In the scalar case, the solution can also be found in closed form as the weighted median of the nodes of a broken line. Bibliogr. 30.

Keywords: piecewise affine function, broken line, least absolute values, subdifferential, hypodifferential, weighted median.

References

1. V. I. Mudrov and V. L. Kushko, *Methods of Processing Measurements: Quasiplausible Estimates* (Radio Svyaz', Moscow, 1983) [Russian].
2. S. I. Zukhovitskii, On the approximation of an incompatible system of linear equations using the principle of minimizing the sum of the moduli of all deviations, *Dokl. Akad. Nauk SSSR* **143** (5), 1030–1033 (1962).

3. **P. A. Akimov** and **A. I. Matasov**, An iterative algorithm for l_1 -norm approximation in dynamic estimation problems, *Avtom. Telemekh.*, No. 5, 7–26 (2015) [Russian] [*Autom. Remote Control* **76** (5), 733–748 (2015)].
4. **A. V. Lakeev** and **S. I. Noskov**, The method of least moduli for linear regression: The number of zero approximation errors, *Sovrem. Tekhnol., Sist. Anal., Model.*, No. 2, 48–50 (2012) [Russian].
5. **V. A. Gorelik** and **O. S. Trembacheva**, Solution of the linear regression problem using matrix correction methods in the l_1 -metric, *Zh. Vychisl. Mat. Mat. Fiz.* **56** (2), 202–207 (2016) [Russian] [*Comput. Math. Math. Phys.* **56** (2), 200–205 (2016)].
6. **V. A. Surin** and **A. N. Tyrsin**, Application of the generalized method of least moduli in problems of image processing and analysis, *Vestn. Astrakhan. Gos. Tekhn. Univ., Ser. Upr. Vychisl. Tekhn. Inform.*, No. 2, 45–55 (2020), DOI: 10.24143/2072-9502-2020-2-45-55 [Russian].
7. **G. Sh. Tamasyan**, On wild points, in *Sel. Talks Semin. Optimization, Machine Learning and Artificial Intelligence* (St. Petersburg. Gos. Univ, St. Petersburg, 2023) [Russian], URL: oml.cmlaboratory.com/reps23.shtml (accessed: Oct. 27, 2024).
8. **S. P. Sharyi**, Strong consistency in the problem of dependence recovery under interval data uncertainty, *Vychisl. Tekhnol.* **22** (2), 150–172 (2017) [Russian].
9. **S. P. Sharyi** and **I. A. Sharaya**, Recognition of solvability of interval equations and its applications to data analysis, *Vychisl. Tekhnol.* **18** (3), 80–109 (2013) [Russian].
10. **I. I. Eremin**, The “penalty” method in convex programming, *Dokl. Akad. Nauk SSSR* **173** (4), 748–751 (1967) [Russian].
11. **M. V. Dolgopolik**, Fundamentals of the theory of exact penalty functions, in *Sel. Talks Semin. Constructive Nonsmooth Analysis and Nondifferentiable Optimization* (St. Petersburg. Gos. Univ, St. Petersburg, 2015) [Russian], URL: cnsa.cmlaboratory.com/reps15.shtml (accessed: Oct. 27, 2024).
12. **L. N. Polyakova**, The Method of exact penalties: Another approach, in *Sel. Talks Semin. Constructive Nonsmooth Analysis and Nondifferentiable Optimization* (St. Petersburg. Gos. Univ, St. Petersburg, 2014) [Russian], URL: cnsa.cmlaboratory.com/reps14.shtml (accessed: Oct. 27, 2024).
13. **V. F. Demyanov** and **V. N. Malozyomov**, *Introduction to Minimax* (Nauka, Moscow, 1972) [Russian].
14. **V. F. Demyanov** and **L. V. Vasilyev**, *Nondifferentiable Optimization* (Nauka, Moscow, 1981) [Russian].
15. **V. F. Demyanov** and **A. M. Rubinov**, *Fundamentals of Nonsmooth Analysis and Quasidifferential Calculus* (Nauka, Moscow, 1990) [Russian].
16. **L. N. Polyakova**, Continuous methods for unconditional minimization of hypodifferentiable functions, *Vestn. St. Petersburg. Univ., Ser. 10*, No. 3, 71–81 (2007) [Russian].

17. **M. V. Dolgopolik**, On continuous codifferentiability, in *Sel. Talks Semin. Constructive Nonsmooth Analysis and Nondifferentiable Optimization* (St. Petersburg. Gos. Univ, St. Petersburg, 2019) [Russian], URL: cnsa.cmlaboratory.com/rep19.shtml (accessed: Oct. 27, 2024).
18. **V. A. Zalgaller**, Representation of functions of several variables by differences of convex functions, *Zap. Nauchn. Semin. POMI* **246**, 36–65 (1997) [Russian] [*J. Math. Sci., New York* **100** (3), 2209–2227 (2000)].
19. **D. Melzer**, On the expressibility of piecewise linear continuous functions as the difference of two piecewise linear convex functions, in *Quasidifferential Calculus* (North-Holland, Amsterdam, 1986), pp. 118–134 (Math. Program. Stud., Vol. 29).
20. **A. Kripfganz** and **R. Schulze**, Piecewise affine functions as a difference of two convex functions, *Optimization* **18** (1), 23–29 (1987), DOI: 10.1080/02331938708843210.
21. **V. V. Gorokhovik**, **O. I. Zorko**, and **G. Birkhoff**, Piecewise affine functions and polyhedral sets, *Optimization* **31** (3), 209–221 (1994), DOI: 10.1080/02331939408844018.
22. **V. V. Gorokhovik**, Geometrical and analytical characteristic properties of piecewise affine mappings (Cornell Univ., Ithaca, NY, 2011) (Cornell Univ. Libr. e-Print Archive, arXiv:1111.1389), DOI: 10.48550/arXiv.1111.1389.
23. **T. A. Angelov**, Representation of piecewise affine functions as a difference of polyhedral ones, *Vestn. St. Petersburg. Univ., Ser. 10*, No. 1, 4–18 (2016) [Russian].
24. **G. Sh. Tamasyan**, On a compact representation of the codifferential of a continuous broken line given in the Bernstein form, in *Sel. Talks Semin. Optimization, Machine Learning and Artificial Intelligence* (St. Petersburg. Gos. Univ, St. Petersburg, 2023) [Russian], URL: oml.cmlaboratory.com/rep23.shtml (accessed: Oct. 27, 2024).
25. **M. K. Gavurin** and **V. N. Malozyomov**, *Extremum Problems with Linear Constraints* (Izd. LGU, Leningrad, 1984) [Russian].
26. **F. P. Vasilyev** and **A. Yu. Ivanitskii**, *Linear Programming* (MTsNMO, Moscow, 2020) [Russian].
27. **G. Sh. Tamasyan** and **G. S. Shulga**, To the question of minimization of a convex piecewise affine function, in *Sel. Talks Semin. Optimization, Machine Learning and Artificial Intelligence* (St. Petersburg. Gos. Univ, St. Petersburg, 2022) [Russian], URL: oml.cmlaboratory.com/rep22.shtml (accessed: Oct. 27, 2024).
28. **A. S. Malistov**, On finding the median of an array in linear time, in *Mat. Prosveshch., Ser. 3*, No. 21 (MTsNMO, Moscow, 2017), pp. 265–270 [Russian].
29. **G. Sh. Tamasyan** and **G. S. Shulga**, Fast algorithm for minimizing a convex broken line, in *Sel. Talks Semin. Constructive Nonsmooth Analysis and Nondifferentiable Optimization* (St. Petersburg. Gos. Univ, St. Petersburg, 2021) [Russian], URL: cnsa.cmlaboratory.com/rep21.shtml (accessed: Oct. 27, 2024).

-
- 30. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein,** *Introduction to Algorithms* (MIT Press, Cambridge, MA, 2009; Vilyams, Moscow, 2013 [Russian]).

Grigory Sh. Tamasyan
Georgy S. Shulga

Received March 26, 2024
Revised April 20, 2024
Accepted June 22, 2024

МЕТОД ДЕКОМПОЗИЦИИ ДЛЯ УПРАВЛЕНИЯ ЗАПАСАМИ В ДВУХЭШЕЛОННОЙ СИСТЕМЕ СКЛАДОВ

А. Д. Юськов^{1,a}, И. Н. Кулаченко^{2,b},
А. А. Мельников^{2,c}, Ю. А. Кочетов^{2,d}

¹ Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

² Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: ^a a.yuskov@ngs.ru, ^b ink@math.nsc.ru,
^c melnikov@math.nsc.ru, ^d jkochet@math.nsc.ru

Аннотация. Склады первого эшелона в двухэшелонной системе предназначены для выполнения заказов клиентов. Во втором эшелоне находится центральный склад, пополняющий запасы на складах первого эшелона. Заказы клиентов можно выполнять частично, но общая доля выполненных заказов должна быть не меньше заданного порога. Требуется минимизировать общую стоимость хранения товаров на всех складах. Работа системы моделируется с помощью детерминированной имитационной модели, которая вычисляет долю удовлетворения заказов и стоимость хранения в течение планового периода в зависимости от параметров управления запасами на каждом складе по каждому типу товара. Разработан метод декомпозиции, основанный на решении подзадач для каждого типа товара. Предложены подходы для точного решения задачи. Приводятся результаты вычислительных экспериментов на примерах со 100 складами и 1000 типами товаров. На примерах с известным точным решением в двух случаях удалось найти оптимум, в остальных случаях отклонение от оптимума составило не более 1,9%. Табл. 5, ил. 1, библиогр. 23.

Ключевые слова: оптимизация «чёрного ящика», задача о рюкзаке, локальный поиск.

Введение

Оптимизация цепей поставок играет важную роль для сокращения издержек крупных компаний [1, 2]. Как правило, поступление заказов клиентов трудно предсказуемо, поэтому хранение запасов на складах

необходимо для своевременного удовлетворения запросов клиентов. Однако это ведёт к большим затратам на хранение. Управление политиками пополнения запасов направлено на поиск компромисса между долей удовлетворённых заказов и затратами на хранение. Основным способом управления запасами является использование на каждом складе политик хранения, которые регулируют запросы на пополнение склада. При разных политиках склады могут заказывать новые товары периодически или основываясь на текущих остатках. Размер заказа также может быть фиксированным или основываться на спросе клиентов. Обзор различных подходов, связанных с управлением запасами, можно найти в [3].

Многоэшелонные модели складов активно исследовались с середины прошлого столетия как в теоретическом [4, 5], так и в алгоритмическом плане [6]. В работах [7, 8] авторы предлагают математическую модель и методы решения с помощью эвристик, точных подходов и стохастического программирования. К сожалению, такой путь приводит либо к сильно упрощённым моделям, либо к высокой вычислительной сложности методов [9]. В данной работе рассматривается другой подход, который имеет ряд преимуществ, главным из которых является возможность сколь угодно точно моделировать бизнес-процессы. Он основан на имитационном моделировании и использовании таких моделей в численных методах. В этом случае алгоритм оптимизации использует имитационную модель в качестве «чёрного ящика» и оптимизирует её входные параметры. Недавние исследования продемонстрировали высокий потенциал этого подхода в детерминированном [10] и стохастическом [11] случаях. Однако, поскольку свойства целевой функции и ограничений теперь неизвестны, а алгоритм не может использовать специфическую информацию о задаче, эта схема порождает новые трудности: нет аналитических выражений для всех или части ограничений и целевой функции, нет производных, градиентов, свойств выпуклости и др. В некоторых подходах используются только входные данные и вывод симулятора [12, 13], другие сочетают моделирование с математическим программированием [14], стараясь привлечь как можно больше дополнительной информации об оптимизируемом бизнес-процессе. В последнее время растёт внимание ко второму подходу [15]. Насколько нам известно, постановка задачи, приведённая в данной работе, нова, хотя ранее встречались похожие варианты, например, в [16, 17].

В данной работе предлагаются идеи декомпозиции чёрного ящика на много маленьких, оптимизируемых независимо друг от друга, а потом собираемых воедино моделью о многовариантном рюкзаке. Исследуется задача управления запасами в двухэшелонной системе складов.

Склады первого эшелона предназначены для выполнения заказов клиентов. Во втором эшелоне находится центральный склад. Он не обслуживает заказы клиентов и предназначен только для пополнения запасов на складах первого эшелона. Заказы клиентов можно выполнять частично, но общая доля выполненных заказов должна быть не меньше заданного порога. Требуется минимизировать общую стоимость хранения товаров на всех складах при условии, что доля выполненных заказов не ниже указанного порога. Для вычисления стоимости хранения и доли выполненных заказов используется детерминированная имитационная модель. Для решения задачи предлагается эвристический алгоритм, основанный на идеях декомпозиции. Алгоритм решает подзадачи для одного типа товаров на всех складах при разных значениях порога, используя имитационную модель как чёрный ящик. Эти решения затем используются в модели о многовариантном рюкзаке для поиска оптимальной комбинации долей удовлетворения заказов клиентов по каждому типу товаров. Показано, что такой подход гарантирует получение точного решения всей задачи, если точно решать подзадачи для каждого типа товаров и правильно выбрать доли удовлетворения заказов. Приводится нижняя оценка минимума суммарных затрат, которая позволяет строить точные алгоритмы. Проведены вычислительные эксперименты на сгенерированных примерах со 100 складами и 1000 типами товаров. Эксперименты показали высокую эффективность предложенного подхода и его превосходство перед пакетом *Nevergrad* [18] и специализированным пакетом МЕО для решения описанной задачи, предоставленным компанией вместе с имитационной моделью. На примерах с известным точным решением в двух случаях удалось найти оптимум, в остальных случаях отклонение от оптимума составило не более 1,9%. Данная статья является расширенной и исправленной версией работы [19], представленной на международной конференции *MOTOR-2023* в Екатеринбурге. В новой версии используется улучшенная модель для одного типа товаров, чтобы гарантировать оптимальность финального решения при правильно подобранных долях выполнения заказов клиентов.

Статья организована следующим образом. В разд. 1 приводится постановка задачи. В разд. 2 описывается предлагаемый метод декомпозиции, в частности, в п. 2.2 представлен алгоритм координатора, а в п. 2.3 — алгоритм локального поиска для подзадач одного товара. В разд. 3 представлены некоторые свойства алгоритма и идеи для построения точного решения. В разд. 4 обсуждаются результаты вычислительных экспериментов на реальных данных, а в заключении формулируются основные выводы.

1. Двухэшелонная задача управления запасами

Политика управления запасами определяет, когда и в каком количестве отправлять заказы на пополнение. Ниже рассматривается двухэшелонная система хранения со складами, которая состоит из нескольких местных складов в первом эшелоне и одного центрального склада во втором эшелоне.

На местных складах применяется политика (s, S) . Для каждого отдельного товара эта политика содержит два целочисленных положительных параметра s и S , представляющих минимальный и максимальный уровни запасов. Когда на склад приходит заказ на товар, он немедленно удовлетворяется товарами со склада, если таковые имеются в наличии; в противном случае он резервируется. Виртуальный уровень запасов — это количество товара в наличии, а также количество товаров, отправленных на склад, но ещё не полученных, за вычетом зарезервированных. Если виртуальный уровень запасов на складе опускается ниже порогового значения s , то на центральный склад отправляется запрос на пополнение запасов до уровня S .

На центральном складе применяется политика $(\bar{S} - 1, \bar{S})$. Данная политика содержит только один параметр $\bar{S}$, представляющий уровень пополнения. Этот параметр определяет количество товаров, хранящихся на складе. Если центральный склад получает запрос на пополнение, то он выполняет его и запрашивает такое же количество товаров у внешнего поставщика. Предполагается, что у этого поставщика имеется неограниченное количество товаров.

Если существует m местных складов, то должны быть определены $2m + 1$ параметров политик для управления уровнями запасов на всех складах для одного товара. Так как на складах одновременно хранится большое количество различных видов товаров, число переменных может быть довольно большим.

Показатели эффективности системы хранения обычно связаны с удовлетворённостью клиентов: например, с процентом заказов или спроса, выполненных в установленные сроки. Помимо удовлетворённости клиентов, для компании также важны затраты на хранение запасов. Стоимость хранения равна интегралу от реального уровня запасов по времени хранения, умноженному на стоимость хранения одной единицы товара.

Оптимизация системы хранения заключается в определении таких значений параметров управления запасами s и S для всех местных складов, а также параметров $\bar{S}$ центрального склада для каждого товара, чтобы процент заказов, выполненных сразу, был не меньше заданного значения, а затраты на хранение были минимальны. Табл. 1 содержит обозначения, используемые в статье.

Таблица 1

Список обозначений

Множества:	
I	множество индексов товаров
F	множество индексов местных складов
Функции:	
φ^L	суммарный спрос на местных складах
φ_i^L	суммарный спрос на товар $i \in I$ на местных складах
φ^C	суммарный спрос на центральном складе
φ_i^C	суммарный спрос на товар $i \in I$ на центральном складе
ψ_i^L	суммарный удовлетворённый спрос на товар $i \in I$ на мест. складах
ψ_i^C	суммарный удовлетворённый спрос на товар $i \in I$ на центр. складе
C	стоимость хранения, вычисляемая имитационной моделью
C_i	стоимость хранения товаров типа $i \in I$
ω^L	доля удовлетворённого спроса на местных складах
ω^C	доля удовлетворённого спроса на центральном складе
l	минимальное возможное значение переменной
u	максимальное возможное значение переменной
Константы:	
α	минимальная допустимая доля удовлетворённого суммарного спроса на местных складах
β	минимальная допустимая доля удовлетворённого суммарного спроса на центральном складе
Переменные задачи:	
x_{if}^1	мин. уровень запасов для товара $i \in I$ на местном складе $f \in F$
x_{if}^2	макс. уровень запасов для товара $i \in I$ на местном складе $f \in F$
x_i^3	уровень пополнения для товара $i \in I$ на центральном складе
X	вектор переменных параметров политик, состоящий из всех $x_{if}^1, x_{if}^2, x_i^3$
X_i	вектор параметров политик, состоящий из переменных $x_{if}^1, x_{if}^2, x_i^3$ для товара i

Функции $\varphi^C, \varphi_i^C, \psi_i^L, \psi_i^C, C_i, \omega^L, \omega^C$ зависят от параметров политик управления запасами.

В данных обозначениях задача управления запасами может быть записана следующим образом:

$$\min_X C(X), \quad (1)$$

$$\omega^L(X) \geq \alpha, \quad (2)$$

$$\omega^C(X) \geq \beta, \quad (3)$$

$$x_{if}^2 \geq x_{if}^1, \quad i \in I, f \in F, \quad (4)$$

$$l(X) \leq X \leq u(X), \quad X \in \mathbb{Z}. \quad (5)$$

Целевая функция (1) минимизирует общую стоимость хранения запасов. Показатели удовлетворённости обеспечиваются в силу ограничений (2)–(3). Неравенства (4) гарантируют, что минимальный уровень запасов не больше максимального. Область допустимых значений переменных определяется при помощи неравенств (5).

Будем рассматривать случай, в котором известно, что общая стоимость хранения вычисляется как сумма стоимостей хранения всех товаров. Тогда можно выписать следующие соотношения:

$$C(X) = \sum_i C_i(X),$$

$$\varphi^L = \sum_i \varphi_i^L,$$

$$\varphi^C(X) = \sum_i \varphi_i^C(X),$$

$$\psi^L(X) = \sum_i \psi_i^L(X),$$

$$\psi^C(X) = \sum_i \psi_i^C(X),$$

$$\omega^L(X) = \psi^L(X)/\varphi^L,$$

$$\omega^C(X) = \psi^C(X)/\varphi^C(X).$$

Для получения значений φ_i^L , $\varphi_i^C(X)$, $C_i(X)$, $\psi_i^L(X)$ и $\psi_i^C(X)$ при заданных параметрах X используется детерминированная имитационная модель.

2. Предлагаемый метод решения

Проведём декомпозицию представленной задачи, которая позволит разбить её на независимые подзадачи, соответствующие отдельным товарам. Выгода от использования такой декомпозиции заключается в том, что моделирование работы склада для разных товаров может выполняться независимо, а это открывает возможности выполнения параллельных вычислений. На основе предложенной декомпозиции построим многоагентную систему, в которой подзадачи решаются отдельными агентами, управляемыми центральным координатором.

2.1. Декомпозиция задачи. Можно заметить, что целевая функция (1) и ограничения (2), (3) представляют сумму независимых величин

по товарам. Также известно, что функции $C_i(X)$, $\varphi_i^C(X)$, $\psi_i^L(X)$ и $\psi_i^C(X)$ могут быть вычислены независимо для каждого товара. Тем самым выглядит разумной схема, в которой алгоритм пытается подобрать оптимальный вклад каждого товара в ограничения (2), (3), а затем независимо находит наилучшие в плане стоимости параметры политик хранения для каждого товара так, чтобы удовлетворялись заданные пороги удовлетворения спроса.

Ограничение (2) можем переписать следующим образом:

$$\sum_{i \in I} \psi_i^L(X_i) \geq \alpha \sum_{i \in I} \varphi_i^L, \quad (6)$$

а (3) запишем в таком виде:

$$\sum_{i \in I} (\psi_i^C(X_i) - \beta \varphi_i^C(X_i)) \geq 0. \quad (7)$$

Введём следующие обозначения: переменные γ_i , $i \in I$, определяют нижнюю границу удовлетворённого спроса клиентов на i -й товар (значение под суммой в левой части ограничения (6)); переменные δ_i , $i \in I$, определяют нижнюю границу значения под суммой в ограничении (7) для i -го товара: $\psi_i^C(X_i) - \beta \varphi_i^C(X_i) \geq \delta_i$. С помощью этих обозначений задачу можно переписать следующим образом:

$$\min_{(\gamma_i), (\delta_i)} \sum_{i \in I} C_i(\gamma_i, \delta_i), \quad (8)$$

$$\sum_{i \in I} \gamma_i \geq \alpha \sum_{i \in I} \varphi_i^L, \quad (9)$$

$$\sum_{i \in I} \delta_i \geq 0, \quad (10)$$

где стоимость $C_i(\gamma_i, \delta_i)$ хранения товара i на всех складах находится из решения задачи $\text{OneItem}_i(\gamma_i, \delta_i)$

$$\min_{X_i} C_i(X_i), \quad (11)$$

$$\psi_i^L(X_i) \geq \gamma_i, \quad (12)$$

$$\psi_i^C(X_i) - \beta \varphi_i^C(X_i) \geq \delta_i, \quad (13)$$

$$x_{if}^2 \geq x_{if}^1, \quad f \in F, \quad (14)$$

$$l(X_i) \leq X_i \leq u(X_i), \quad X_i \in \mathbb{Z}. \quad (15)$$

Задача координатора (8)–(10) направлена на нахождение порогового значения удовлетворённого спроса для каждого товара таким образом, чтобы суммарная доля удовлетворённого спроса составляла не менее α для местных складов и не менее β для центрального склада. Заметим,

что ограничения (9), (10) гарантируют это требование при условии выполнения ограничений (12), (13). На нижнем уровне рассматриваются $|I|$ задач оптимизации, направленных на минимизацию затрат на хранение для каждого товара при условии, что должен быть удовлетворён определённый объём спроса. Значения целевой функции (11) и ограничений (12), (13) вычисляются при помощи имитационной модели. В разд. 3 будет показано, что оптимальные решения задач (1)–(5) и (8)–(15) совпадают.

Заметим, что при данных политиках хранения местные склады всегда запрашивают у центрального не больше предметов, чем суммарный спрос от клиентов, поэтому $\varphi_i^C(X_i) \leq \varphi_i^L$. Так как $\psi_i^C(X_i) \leq \varphi_i^C(X)$, можно установить границы для значений δ :

$$-\beta\varphi_i^L \leq \psi_i^C(X_i) - \beta\varphi_i^C(X_i) \leq (1 - \beta)\varphi_i^L.$$

Для допустимых запросов γ_i и δ_i введём множества $\Gamma_i = \{0, \dots, \varphi_i^L\}$, $\Delta_i = \{-\beta\varphi_i^L, \dots, (1 - \beta)\varphi_i^L\}$, $U_i = \Gamma_i \times \Delta_i$. Минимальные и максимальные возможные значения запросов используются в алгоритме, представленном далее, а обозначения Γ_i и Δ_i используются при теоретических обоснованиях в разд. 3.

2.2. Многоагентная система. В этом пункте опишем эвристический подход к решению задачи (8)–(15). Подход основан на том факте, что задачи одного товара (11)–(15) могут обрабатываться независимо для каждого отдельного $i \in I$. Это позволяет разработать многоагентную схему (MAS), где несколько *рабочих агентов* параллельно решают задачи для одного товара с использованием алгоритма локального поиска, в то время как *координирующий агент* направляет поиск, запрашивая решения для некоторых конкретных γ_i и δ_i , и агрегирует решения задач для одного товара в решение для исходной задачи.

Решения для отдельных товаров. Во время вычисления сохраняем наборы наилучших найденных решений $P_i = \{X_i^1, \dots, X_i^{k_i}\}$ для каждой задачи **OneItem**, $i \in I$, и разных пар (γ, δ) . Пусть имеется набор номеров запомненных решений $K_i = \{1, \dots, k_i\}$, $i \in I$. После решения подзадачи для каждого решения X_i^k , $i \in I$, $k \in K_i$, известны связанные с ним величины $d_{ik}^C = \varphi_i^C(X_i^k)$, $s_{ik}^L = \psi_i^L(X_i^k)$, $s_{ik}^C = \psi_i^C(X_i^k)$, и $c_{ik} = C_i(X_i^k)$. Кроме того, сохраняется пара $(\gamma_{ik}, \delta_{ik})$ соответствующих значений параметров задачи **OneItem**.

Скажем, что решение X_i^j *доминирует* решение X_i^k , если $\gamma_{ij} \geq \gamma_{ik}$, $\delta_{ij} \geq \delta_{ik}$ и $c_{ij} \leq c_{ik}$. Решение X_i^j назовём *недоминируемым*, если в P_i нет X_i'' , доминирующего X_i^j . Всякий раз при сохранении нового решения

проверяем, доминирует ли оно некоторое из существующих и является ли доминируемым. Все доминируемые решения удаляются.

На начальном шаге для вычисления запрашиваются решения задачи для одного товара, соответствующие парам (γ_i, δ_i) таким, что $\gamma_i = \delta_i + \beta \varphi_i^L = \lceil \frac{l}{p} \varphi_i^L \rceil$. Здесь $p \in \mathbb{Z}^+$ является параметром алгоритма, а l принимает целочисленные значения от 1 до p . Следует отметить, что решение, полученное для некоторой пары (γ, δ) , допустимо для задачи с одним элементом с параметрами (γ', δ') такими, что $\gamma \geq \gamma'$ и $\delta \geq \delta'$. Таким образом, разумно начать вычисление решений с задачи для одного товара с пары (γ, δ) , соответствующей значению $l = p$. Полученное решение используется позже для инициализации локального поиска при решении задачи с $l = p - 1$. Затем процесс повторяется дальше, обрабатывая значения l в порядке убывания. В наших экспериментах использовано $p = 20$. Если для элемента требуется менее 10^5 оцениваний целевой функции, чтобы полностью перечислить все возможные конфигурации значений политик, то применяем полный перебор к этим элементам и сохраняем все решения, не являющиеся доминируемыми.

Координирующий агент. Решение исходной задачи (1)–(5) получаем, беря одно решение для каждой из задач одного товара из набора решений и объединяя эти решения вместе. Для заданного набора решений $\mathbf{P} = (P_1, \dots, P_{|I|})$ наилучшая комбинация может быть получена путём решения вспомогательной задачи булева программирования. Введём булевы переменные (z_{ik}) , $i \in I$, $k \in P_i$, равные единице, если решение X_i^k входит в итоговое решение, и нулю в противном случае. Используя эти обозначения, сформулируем вспомогательную задачу $\text{coordinator}(\mathbf{P})$, решив которую, можно получить ответ к исходной задаче:

$$\min_{(z_{ik})} \sum_{i \in I} \sum_{k \in K_i} c_{ik} z_{ik}, \quad (16)$$

$$\sum_{k \in K_i} z_{ik} = 1, \quad i \in I, \quad (17)$$

$$\sum_{i \in I} \sum_{k \in K_i} s_{ik}^L z_{ik} \geq \alpha \sum_{i \in I} \varphi_i^L, \quad (18)$$

$$\sum_{i \in I} \sum_{k \in K_i} s_{ik}^C z_{ik} \geq \beta \sum_{i \in I} \sum_{k \in K_i} d_{ik}^C z_{ik}, \quad (19)$$

$$z_{ik} \in \{0, 1\}, \quad i \in I, k \in K_i. \quad (20)$$

Целевая функция (16) состоит в минимизации стоимости хранения запасов в полном решении. В силу равенств (17) для каждой из задач одного товара в итоговое решение будет взято единственное решение.

Неравенства (18), (19) гарантируют допустимый процент удовлетворённых заказов. При изложенной ранее схеме запроса частичных решений данная задача всегда имеет допустимое решение. Действительно, решение с коэффициентами выполнения на местных и центральном складах, равными единице, всегда помещается в историю на шаге инициализации, когда решается задача одного товара, соответствующая $l = p$. По определению доминирования такое решение может быть удалено из истории только в том случае, если в него будет помещено другое решение, имеющее тот же процент выполнения заказов.

Алгоритм координирующего агента итеративный: на каждом шаге решается задача верхнего уровня для текущего набора частичных решений задач одного товара, а затем координатор посылает запросы на вычисление решений подзадач с установленными процентами выполнения заказов в окрестности тех, которые соответствуют решениям, выбранным в итоговое решение.

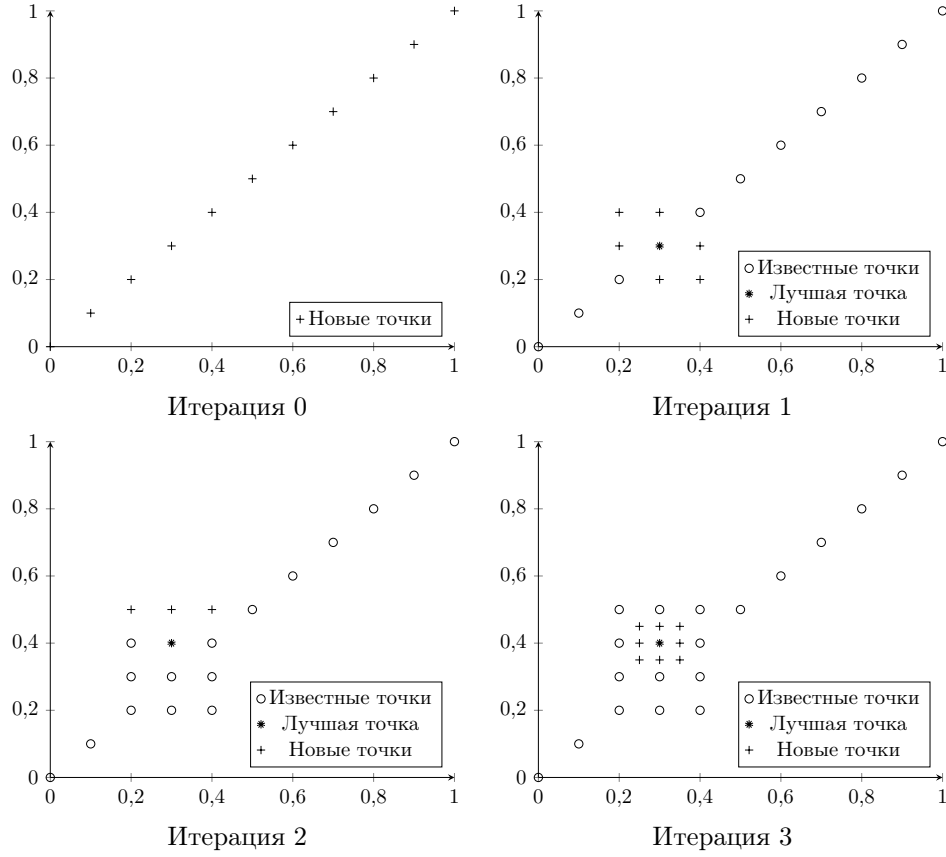
Пусть (z_{ik}) , $i \in I$, $k \in K_i$, — оптимальное решение задачи координатора на текущем шаге, и пусть $i \in I$, $k \in K_i$ таковы, что $z_{ik} = 1$. Тогда X_i^k является частью итогового решения, построенного на текущем шаге. Рассмотрим пару $(\gamma_{ik}, \delta_{ik})$, относящуюся к этому решению. Перед началом следующего шага координатор запросит найти решения задач **OneItem** _{i} со схожими уровнями.

Через ε обозначим величину шага, на который в задачах одного товара изменяются уровни γ и δ . Изначально $\varepsilon = \varepsilon_0$, и эта величина остаётся постоянной, пока полное решение изменяется. Когда полное решение

Алгоритм 1. Многоагентная система (MAS)

```

1: function MAS
2:    $\mathbf{Q}^0 = (Q_i^0), i \in I;$  ▷ Начальное множество пар  $(\gamma, \delta)$ 
3:    $P_i = \emptyset, i \in I;$  ▷ Начальный набор решений для каждого товара
4:   for all  $(\gamma_i, \delta_i) \in Q_i^0$  do oneItemLS $(\gamma_i, \delta_i, P_i);$ 
5:    $\gamma^0, \delta^0 \leftarrow \text{solveMP}(P);$ 
6:    $t \leftarrow 1, \varepsilon \leftarrow \varepsilon_0;$ 
7:   while не достигнут критерий остановки do
8:     for  $i \in I$  do
9:        $Q_i^t \leftarrow Q_i^{t-1} \cup A(\gamma_i^t, \delta_i^t);$ 
10:      for all  $(\gamma_i, \delta_i) \in Q_i^t \setminus Q_i^{t-1}$  do oneItemLS $(\gamma_i, \delta_i, P_i);$ 
11:       $\gamma^t, \delta^t \leftarrow \text{solveMP}(P);$ 
12:      if  $\gamma^t = \gamma^{t-1}$  и  $\delta^t = \delta^{t-1}$  then  $\varepsilon \leftarrow \varepsilon/2;$ 
13:       $t \leftarrow t + 1;$ 
14:   return лучшее найденное решение.
```

Рис. 1. Пример изменения множества Q_i

оказывается таким же, как и на предыдущем шаге, изменяем величину шага $\varepsilon \leftarrow \varepsilon/2$ — аналогично алгоритмам координатного поиска [20]. Таким образом, если ε — величина шага на итерации t , γ_i^t, δ_i^t — лучшие найденные на текущий момент запросы процентов выполнения заказов, то на следующей итерации запрашиваются решения задач для уровней из множества $A(\gamma_i^t, \delta_i^t) = \{\gamma_i^t \pm \lceil \varepsilon \varphi_i^L \rceil\} \times \{\delta_i^t \pm \lceil \varepsilon \varphi_i^L \rceil\}$. Пусть Q_i^t — множество запросов локального поиска для товара i на шаге t . Тогда $Q_i^0 = \{(\lceil \frac{l}{p} \varphi_i^L \rceil, \lceil \frac{l}{p} \varphi_i^L \rceil - \beta \varphi_i^L) \mid l = 0, \dots, p\}$, $Q_i^{t+1} = Q_i^t \cup A(\gamma_i^t, \delta_i^t)$.

Если $\lceil \varepsilon \varphi_i^L \rceil \leq 1$ для всех $i \in I$ и объединённое решение с предыдущего шага не меняется, то алгоритм завершается. Изложенная многоагентная система представлена в виде алгоритма 1. На рис. ?? приведён пример эволюции множества Q_i для некоторого товара в ходе работы этого алгоритма.

2.3. Задача для одного товара. Агенты могут независимо решать задачи, поставленные координатором. Чтобы найти решения для этих задач, агенты используют рандомизированный локальный поиск с дополнительными компонентами, повышающими его производительность. При заданном типе товаров $i \in I$ и паре ограничений (γ_i, δ_i) задача $\text{OneItem}_i(\gamma_i, \delta_i)$, определённая как (11)–(15), направлена на то, чтобы найти параметры X_i политик для товара i на всех складах.

В случае, когда множество P_i найденных на данный момент решений без доминирования не пусто, рандомизированный локальный поиск начинается с наиболее подходящего решения из этого множества. Это решение $X_i^{k'} \in P_i$, где

$$k' = \arg \min_{k \in K_i} \{C_i(X_i^k) \mid \gamma_{ik} \geq \gamma \text{ и } \delta_{ik} \geq \delta\}.$$

На этапе инициализации MAS набор решений может быть пустым. В соответствии с порядком запросов на этапе инициализации, описанным в п. 2.2, первым запрашивается то решение, в котором удовлетворены все заказы. Такое решение можно построить с помощью простой эвристики. Эта эвристика использует предположение о независимости каждого местного склада при некотором большом значении $\bar{S}$ на центральном складе и необходимости полностью удовлетворить весь спрос на местных складах. Используя предположение о монотонности стоимости хранения запасов на отрезке $[s, S]$, находим наилучшие решения с помощью алгоритма дихотомии. После этого корректируем политику $\bar{S}$ центрального склада, чтобы минимизировать затраты на хранение на нём.

Окрестность $N(X_i)$, используемая в рандомизированном локальном поиске, состоит из полудопустимых векторов (т. е. удовлетворяющих ограничениям (14)), которые определяют параметры политики, различающиеся не более чем на двух складах (местных или центральном), и разница составляет не более двух единиц. Рандомизированная окрестность решения X_i размера $r \in \mathbb{Z}^*$ будет обозначаться через $N_r(X_i)$. Она состоит из r случайно выбранных элементов множества $N(X_i)$.

Также используем список запретов, в котором сохраняем десять последних посещённых решений, чтобы избежать попадания в цикл, когда набор $N(X_i)$ мал и рандомизация не приводит к выходу из локальных оптимумов. Во время рассмотрения окрестности $N_r(X_i)$ алгоритм переходит к лучшему соседу, которого нет в списке запретов. При этом используем штрафную функцию

$$\text{InfPenalty}(X_i^k) = M(\max\{\gamma_i - \psi^L(X_i^k), 0\} + \max\{\delta_i - \psi^C(X_i^k), 0\}),$$

где M — большое число, чтобы предотвратить переход в недопустимое решение.

По мере вычислений обновляем множества P_i , сохраняя новые найденные недоминируемые решения. Когда лучшее из найденных решений не меняется в течение T итераций, поиск возвращает его как лучшее решение. Общая схема рандомизированного локального поиска представлена алгоритмом 2.

Алгоритм 2. Алгоритм локального поиска для задачи одного товара

```

1: function oneItemLS( $\gamma_i, \delta_i, P_i$ )
2:    $X_i = \arg \min_{X_i^k \in P_i} \{C(X_i^k) \mid \gamma_{ik} \geq \gamma_i \text{ и } \delta_{ik} \geq \delta_i\};$ 
3:   while не выполнен критерий остановки do
4:      $X_i \leftarrow \arg \min_{X_i^k \in N_r(X_i)} \{C(X_i^k) + \text{InfPenalty}(X_i^k, v)\};$ 
5:     обновить множество  $P_i$ , добавив решение  $X_i$ ;
6:      $X_i^* \leftarrow$  лучшее из  $X_i, X_i^*$ ;
7:     if выполнен критерий перезапуска then
8:        $X_i \leftarrow X_i^*$ ;
9:       очистить список запретов;
10:  return  $X_i^*$ .

```

3. Точное решение задачи

Лемма 1. Пусть X^* — оптимальные параметры политик хранения, $p_i^* = (\gamma_i^*, \delta_i^*)$ — соответствующие значения для задачи агента (11)–(15). Тогда если на некоторой итерации t для любого товара $p_i^* \in Q_i^t$, т. е. оптимальные значения γ_i^* и δ_i^* попали в множество запросов к локальному поиску, и алгоритм нижнего уровня вернул некоторые оптимальные решения для соответствующих задач (11)–(15) в этих точках, то координатор вернёт оптимальное решение исходной задачи (1)–(5).

Доказательство. Пусть алгоритм вернул решения $X_i^{k'}$ с соответствующими ему уровнями удовлетворённости $p_i^{k'} = (s_{ik'}^L, s_{ik'}^C - \beta d_{ik'}^C)$ и стоимостями $c_{ik'}$. Так как решение агента допустимо, то $p_i^{k'} \succeq p_i^*$. Значит, решение

$$z_{ik'} = 1, \quad z_{ik} = 0, \quad k \neq k', \quad (21)$$

допустимо для задачи координатора. Кроме того, поскольку решение X_i^* допустимо для задачи агента и он вернул оптимальное решение для своей задачи, то $c_{ik'} \leq C_i(X_i^*)$. Тогда для стоимости решения (21) имеет место неравенство $\sum_{i \in I} c_{ik'} \leq C(X^*)$. Таким образом, $\sum_{i \in I} c_{ik'} = C(X^*)$ и решение $X^{k'}$, полученное из (21), оптимально. Значит, решение, возвращённое координатором, также оптимально. Лемма 1 доказана.

Лемма 1 говорит о том, что если мы умеем точно решать задачу агента, то для получения оптимального решения достаточно определить правильные значения γ_i и δ_i . Это позволяет полностью перейти к задаче координатора. Обозначим через $\text{coordinator}(\mathbf{Q}, \mathbf{H})$ процедуру, которая точно решает задачу координатора с множествами запросов $\mathbf{Q} = (Q_1, \dots, Q_n)$ и функциями $\mathbf{H} = (H_1, \dots, H_n)$, выдающими стоимость удовлетворения ограничений с порогами $p_i = (\gamma_i, \delta_i)$, и возвращает вектор оптимальных точек p^* в пространстве ограничений. Пусть $C_i(\gamma_i, \delta_i)$ обозначает оптимальное значение целевой функции при заданных ограничениях:

$$C_i(\gamma_i, \delta_i) = \min_{X_i} \{C_i(X_i) \mid \psi_i^L(X_i) \geq \gamma_i, \psi_i^C(X_i) - \beta\varphi_i^C(X_i) \geq \delta_i\}.$$

Для $p_i^1 = (\gamma_i^1, \delta_i^1)$, $p_i^2 = (\gamma_i^2, \delta_i^2)$ положим $p_i^1 \succeq p_i^2$, если $\gamma_i^1 \geq \gamma_i^2$ и $\delta_i^2 \geq \delta_i^1$. Тогда можно сформулировать следующие утверждения.

Свойство 1. Функция $C_i(p)$ монотонна, т. е. из отношения $p^1 \succeq p^2$ следует $C_i(p^1) \geq C_i(p^2)$.

Свойство монотонности позволяет нам предложить метод для подсчёта нижней оценки для оптимального решения, если известны значения функции в некоторых точках. Пусть U_i — множества всех допустимых значений пар $(\gamma_i, \delta_i) = p_i$ и Q_i — множества точек, в которых значения функций C_i известны.

Лемма 2. Пусть $(0, -\beta\varphi_i^L) \in Q_i$. Если для каждого товара $i \in I$ доопределить функцию C_i меньшим значением, т. е. $C_i'(p) = C_i(p')$, $p' \preceq p$, $p' \in Q_i$, $p \in U_i \setminus Q_i$, точно решить полученную задачу координатора ($Q_i' = U_i, C_i'$), то полученное решение будет нижней оценкой для оптимального решения исходной задачи.

Доказательство. Действительно, пусть $p_i^*, i \in I$, — запросы для оптимального решения. Тогда решение (p_i^*) допустимо и $C_i'(p_i^*) \leq C_i(p_i^*)$. Значит, стоимость оптимального решения изменённой задачи не превосходит $C(p^*)$. Лемма 2 доказана.

Заметим, что точке $(0, -\beta\varphi_i^L)$ соответствует тривиальное решение, в котором все параметры политик равны 0.

Зная нижнюю оценку для данной задачи, нетрудно построить простой точный алгоритм, который находит оптимальное решение. Алгоритм 3 описывает такой метод. Он начинается с некоторого множества точек Q_i , в которых известны значения функции C_i . На каждом шаге алгоритм, зная точное значение C_i в точках Q_i , доопределяет функцию C_i до C_i' во всех точках $U_i \setminus Q_i$ и решает задачу координатора, получая нижнюю оценку. Если эта оценка получена в точке, в которой значение C_i известно

точно, то это решение является решением и для исходной задачи, и алгоритм возвращает это решение. Иначе алгоритм вычисляет значение C_i в полученной точке с целью улучшить оценку.

Алгоритм 3. Точный алгоритм

- 1: Выбрать и оценить начальные точки $\mathbf{Q}$;
 - 2: **while** критерий остановки не выполнен **do**
 - 3: **for all** $i \in I$, $p_i^j \in Q_i$ **do** вычислить точные значения $C_i(p_i^j)$;
 - 4: доопределить функции C_i до C'_i во всех неизвестных точках;
 - 5: $p \leftarrow \text{coordinator}(\mathbf{U}, \mathbf{C}')$;
 - 6: **if** все значения $C'_i(p_i)$ известны точно **then** СТОП; $\triangleright$ Найдено оптимальное решение
 - 7: $Q_i \leftarrow Q_i \cup \{p_i\}$.
-

Теорема 1. Алгоритм 3 находит оптимальное решение задачи.

ДОКАЗАТЕЛЬСТВО. Из доказанного ранее $\sum_i C'_i(p_i)$ является нижней оценкой на стоимость оптимального решения исходной задачи. Если все значения $C_i(p_i)$ известны на ШАГЕ 6, то $\sum_i C'_i(p_i) = \sum_i C_i(p_i)$, т. е. реальная стоимость решения p совпадает с нижней оценкой, и оно оптимально. В противном случае ходя бы для одного $i \in I$ точка p_i неизвестна, и на следующей итерации значение $C_i(p_i)$ вычисляется хотя бы в одной новой точке. Так как множество возможных точек конечно, алгоритм конечен. Теорема 1 доказана.

Кроме того, можно заметить, что способ получения нижних оценок годится в случае, когда рассматривается не всё пространство допустимых значений, а некоторое его подмножество.

Замечание 1. Лемма 2 остаётся верной, если рассматривать не всё пространство решений U_i , а некоторое его прямоугольное подмножество $V_i = \{(\gamma_i, \delta_i) \in U_i \mid a \leq \gamma_i \leq b, c \leq \delta_i \leq d\}$ при условии, что известно значение C_i в точке $\min V_i = (\min_{(\gamma_i, \delta_i) \in V_i} \gamma_i, \min_{(\gamma_i, \delta_i) \in V_i} \delta_i)$. В таком случае полученное решение будет являться нижней оценкой для оптимального решения среди всех $p_i \in V_i$. Доказательство этого факта аналогично доказательству леммы 2.

Наличие способа построения верхних и нижних оценок для подмножеств пространства решений позволяет построить для данной задачи алгоритм, основанный на методе ветвей и границ. Компоненты такого алгоритма выглядят следующим образом. Для текущей подобласти $\mathbf{V} = (V_1, \dots, V_n)$ имеем следующее.

- Верхняя граница: оптимальное решение $\text{coordinator}(\mathbf{V} \cap \mathbf{Q}, \mathbf{C})$.
- Нижняя граница: оптимальное решение $\text{coordinator}(\mathbf{V}, \mathbf{C}')$.
- Ветвление: например, для некоторого товара i выбираем порог γ'_i и разбиваем область на две:

$$V'_i = \{(\gamma_i, \delta_i) \in V_i \mid \gamma_i \leq \gamma'_i\}, \quad V''_i = \{(\gamma_i, \delta_i) \in V_i \mid \gamma_i > \gamma'_i\},$$

или аналогично для δ_i .

- Дополнительно на каждом шаге происходит вычисление функции $C_i(\min V_i)$ для обеспечения корректности леммы 2, а для получения перспективных решений опционально вычисляется $C(\text{coordinator}(\mathbf{V}, \mathbf{C}'))$.

Описанный алгоритм может демонстрировать более высокую производительность, чем алгоритм 3.

Заметим, что в процессе работы алгоритма приходится много раз решать задачу целочисленного программирования (16)–(20), в которой множества P_i состоят из всех возможных точек. Это может оказаться трудоёмкой задачей, поэтому для уменьшения размерности задачи координатора можно использовать следующее изменение.

Замечание 2. Для некоторых точек p_i^1, p_i^2 и функции оценки стоимости H определим полное доминирование относительно H :

$$p_i^1 \preceq_H p_i^2, \quad \text{если } p_i^1 \preceq p_i^2 \text{ и } H(p_i^1) \geq H(p_i^2),$$

т. е. точка p_i^1 хуже точки p_i^2 по всем параметрам, включая значение целевой функции. Тогда в лемме 2 достаточно оставить только не полностью доминируемые точки относительно C'_i . Обозначим это сужение через U'_i .

ДОКАЗАТЕЛЬСТВО. Если $p_i^* \notin U'_i$, то существует точка $p_i^2 \in U'_i$ такая, что $p_i^2 \succeq_{C'_i} p_i^*$. Значит, $p_i^2 \succeq p_i^*$, и решение с p_i^2 допустимо для исходной задачи, а также $C'_i(p_i^2) \leq C'_i(p_i^*) \leq C_i(p_i^*)$. Замечание 2 доказано.

Замечание 3. Введём функции для получения «предыдущих» значений: $\text{prev}_i(\gamma) = \gamma - 1$, $\text{prev}_i(\delta) = \max\{\delta' \in \Delta_i \mid \gamma' < \gamma\}$. Пусть

$$Q_i^\gamma = \{\gamma \mid (\gamma, \delta) \in Q_i\} \cup \{\max \Gamma_i\}, \quad Q_i^\delta = \{\delta \mid (\gamma, \delta) \in Q_i\} \cup \{\max \Delta_i\}.$$

Тогда для корректности леммы 2 функцию C_i достаточно доопределить в точках

$$U'_i = \{(\text{prev}_i(\gamma), \text{prev}_i(\delta)), (\text{prev}_i(\gamma), \delta), \\ (\gamma, \text{prev}_i(\delta)), (\gamma, \delta) \mid \gamma \in Q_i^\gamma, \delta \in Q_i^\delta\}.$$

ДОКАЗАТЕЛЬСТВО. Выберем оптимальную точку $p_i^* = (\gamma_i^*, \delta_i^*)$ и положим $D = \{p \in U'_i \mid p \succeq p_i^*\}$ — множество всех точек, доминирующих p_i^* .

Множество D непусто, так как $\max U'_i \in D$. Достаточно показать, что существует точка $p' \in D$ такая, что $C'_i(p') \leq C_i(p_i^*)$. Рассмотрим множество «самых левых» точек из D :

$$D^l = \{(\gamma_0, \delta) \in D \mid \gamma_0 = \min \{\gamma' \mid (\gamma', \delta') \in D\}\}$$

и возьмём «самую нижнюю» из них: $p_0 = (\gamma_0, \delta_0 = \min\{\delta' \mid (\gamma, \delta') \in D^l\})$. Тогда $C'_i(p_0) = C_i(p_1)$ для некоторой известной точки p_1 . Если $p_1 \preceq p_i^*$, то условие выполнено. Значит, $\delta_1 > \delta^*$ или $\gamma_1 > \gamma^*$.

При $\delta_1 > \delta^*$ с необходимостью известно значение функции C_i в точке $p_2 = (\gamma_0, \text{prev}_i(\delta_1)) \succeq p^*$. Если это не так, то указанное значение должно быть доопределено; противоречие с условием «минимальности» p_0 . Возможны несколько вариантов расположения точки p_2 .

1) В случае $p_2 = p^*$ известно значение $C_i(p^*)$, и требуемое условие выполнено.

2) Если $\delta_2 > \delta^*$ и $\gamma_2 > \gamma^*$, то по условию утверждения должна быть доопределена точка $p_3 = (\text{prev}_i(\delta_2), \text{prev}_i(\gamma_2))$. Повторяем рассуждения для неё.

3) Если $\delta_2 > \delta^*$ и $\gamma_2 = \gamma^*$, то аналогично должна быть доопределена точка $p_4 = (\text{prev}(\delta_2), \gamma_2)$.

4) Случай $\delta_2 = \delta^*$ и $\gamma_2 > \gamma^*$ рассматривается аналогично.

При $\gamma_1 > \gamma^*$ аналогично должна была быть доопределена точка $p_5 = (\text{prev}_i(\gamma_0), \delta_1) \succeq p^*$.

Таким образом, существует точка $p' \in U'_i$, $p' \succeq p^*$, такая, что значение $C'_i(p')$ доопределено значением из некоторой точки $p'_2 \preceq p^*$, а значит, $C'_i(p') \leq C_i(p^*)$. Замечание 3 доказано.

4. Вычислительные эксперименты

Алгоритм MAS был реализован на языке Julia [21]. Все эксперименты проводились на компьютере, оснащённом процессором Intel Core i7-8700 с частотой 3,20 ГГц и имеющем 32 ГБ оперативной памяти, под управлением операционной системы Microsoft Windows 10 Pro.

4.1. Описание тестовых примеров. Тестовые примеры сгенерированы аналогично [22]. Для формирования запросов клиентов на товары применяется распределение Пуассона. Чтобы смоделировать неравномерность спроса и цен, используется процедура, предложенная в [23]. Сначала для каждого товара выбирается непрерывная случайная величина u_{di} из равномерного распределения на отрезке $[0, 1]$. Затем по формуле

$$\nu_i = \frac{\lambda}{\rho_d} u_{di}^{\frac{1-\rho_d}{\rho_d}},$$

где λ — средняя интенсивность спроса, а ρ_d — параметр скошенности для спроса, вычисляется интенсивность спроса на товар i . Затем для каждого склада вновь выбирается значение h_{dif} из равномерного распределения на отрезке $[0, 2]$. Итоговая интенсивность спроса в распределении Пуассона на товар i на складе f равна $\lambda_{if} = \nu_i h_{dif}$.

Стоимость хранения товара i на складе f вычисляется аналогично:

$$c_{if} = \frac{c}{\rho_c} u_{ci}^{\frac{1-\rho_c}{\rho_c}} h_{cif},$$

где c — средняя стоимость хранения, ρ_c — параметр скошенности для стоимости, u_{ci} и h_{cif} — реализации случайных величин из отрезков $[0, 1]$ и $[0, 2]$ соответственно. В проведённых экспериментах использованы параметры $\rho_d = 0,139$, $\rho_c = 0,097$. Кроме того, величины λ_{if} и c_{if} не могут принимать значения меньше 10^{-3} .

Времена поставок для каждого предмета и склада выбираются случайно из равномерного распределения в диапазоне от 1 до 7 дней для местных складов и от 10 до 16 для центрального склада. Для тестирования сгенерированы два семейства примеров: малые и большие. Малые примеры имеют 100 типов товаров и 10 местных складов. У товара есть вероятность 0,025 наличия спроса на каждом складе. Принимая во внимание центральный склад, получаем около 500 переменных на пример. Большие примеры имеют 1000 типов товаров, 100 местных складов и вероятность в 0,045 получения запросов с определённого склада. Это даёт около 10000 переменных. Уровни целевых показателей удовлетворённости были установлены на уровне $\alpha = 0,95$, $\beta = 0,95$.

4.2. Сравнительный анализ. Предложенный алгоритм исследован в сравнении с универсальным пакетом Nevergrad [18] и многоэшелонным оптимизатором МЕО. Для Nevergrad используем оптимизатор NGOpt39 с бюджетом в $3 \cdot 10^5$ оцениваний и предоставляем ему первоначальное решение, найденное с помощью нашей эвристики для случая удовлетворения всех заказов. Если не предоставить первоначального решения, то Nevergrad потребует гораздо больший вычислительный бюджет, чтобы показать результаты, аналогичные представленным далее. Nevergrad и MAS использовали все доступные потоки, а МЕО работает только в однопоточном режиме. Кроме того, Nevergrad использует полную симуляцию, в то время как MAS и МЕО используют симуляции одного товара. Для MAS установлен параметр $\varepsilon_0 = 0,04$. Критерием остановки является достижение $\varepsilon = 1$ и $z^t = z^{t-1}$ (п. 2.2). В локальном поиске $r = 21$, $T = \max\{10, n\}$, где n — число переменных в X_i . Алгоритм локального поиска завершается после 500 оцениваний целевой функции или когда лучшее найденное решение не меняется в течение $10T$ итераций. В алгоритме MAS для товара $i \in I$ и каждой пары (γ_i, δ_i) мы не тратим в общей

Таблица 2

Сравнение MAS, MEO и Nevergrad на малых примерах

№	MAS _{min}	MAS _{avg}	MEO	NG _{min}	NG _{avg}	t _{MAS}	t _{MEO}
1	6485	6487	6912	11 857	12 562	36	20
2	5421	5431	5581	11 536	12 181	33	18
3	5056	5061	5503	8413	8954	25	18
4	5590	5592	5868	8705	9475	20	18
5	12 230	12 258	12 870	16 993	18 540	25	24
6	7122	7125	7518	13 493	14 253	25	19
7	10 846	10 852	11 565	15 747	16 472	22	18
8	11 560	11 571	11 882	17 443	18 028	29	17
9	13 383	13 389	13 743	20 084	21 290	23	20
10	8968	8999	9356	12 999	13 291	26	24

сложности более 2000 оцениваний при решении задач $\text{OneItem}_i(\gamma_i, \delta_i)$. Хотя следует отметить, что изменение параметров может сместить баланс между временем вычислений и качеством результата. В табл. 2 и 3 представлены значения затрат на хранение запасов для соответствующих примеров и алгоритмов.

В табл. 2 приведены результаты для малых примеров. Для каждого примера было выполнено 5 запусков алгоритма MAS и 5 запусков

Таблица 3

Сравнение MAS, MEO и Nevergrad на больших примерах

№	MAS _{min}	MAS _{avg}	MAS _t	MEO	NG _{min}	NG _{avg}	t _{MAS}	t _{MEO}
1	133 591	133 664	135 809	140 153	209 120	215 842	1625,790	617
2	127 924	128 103	129 725	136 991	217 382	219 338	1462,747	622
3	118 993	125 220	120 654	127 671	204 631	208 680	1405,229	892
4	145 825	146 274	147 470	154 442	231 940	233 772	1792,402	892
5	125 180	125 220	126 123	134 837	210 121	213 743	1579,075	903
6	126 049	126 097	128 579	133 663	217 160	219 995	1728,637	620
7	96 095	96 160	102 472	104 676	183 820	186 964	1332,990	356
8	134 922	134 999	137 072	141 286	226 715	229 002	1466,459	879
9	127 997	128 043	133 151	137 233	222 444	224 208	1637,495	366
10	134 605	134 713	136 435	144 226	238 641	240 459	1773,003	909

Nevergrad. В столбцах «MAS_{min}», «MAS_{avg}», «NG_{min}» и «NG_{avg}» представлены наилучшие и средние результаты для этих запусков. Результаты МЕО приведены в колонке «МЕО». Столбцы « t_{MAS} » и « $t_{\text{МЕО}}$ » показывают среднее время выполнения в секундах за один запуск MAS и МЕО.

MAS демонстрирует результаты в среднем на 6% лучше, чем МЕО, за аналогичное время выполнения, а Nevergrad показывает значительно более плохие результаты, хотя его время выполнения было примерно в 100 раз больше, чем у MAS и МЕО. Можно заметить, что разница между минимальными и средними значениями не велика, и алгоритм демонстрирует стабильные результаты.

В табл. 3 представлены результаты для больших примеров. Значение столбцов такое же, как и для табл. 2. Результаты, достигнутые MAS за время выполнения $t = t_{\text{МЕО}}$, представлены в столбце MAS_t. Время работы Nevergrad составляло примерно 4 ч на запуск.

Результаты в столбцах MAS_{avg} и MAS_t на 5% и 4% лучше, чем соответствующие для МЕО. Nevergrad не смог найти сопоставимого решения ни на одном из примеров. По сравнению с предыдущей версией алгоритма результаты улучшились в среднем на 0,6%.

Как оказалось, вспомогательная подзадача **coordinator(P)** достаточно лёгкая для точного решения пакетом целочисленного программирования. Её решение занимает менее 1 с. Отметим также, что из-за особенностей устройства имитационной модели почти всё время тратится на её работу, а время работы остальной части алгоритма незначительно. Время полной симуляции равно сумме времён, затраченных на симуляцию каждого товара.

Таблица 4

Сравнение результатов MAS с точным решением

№	MAS _{min}	MAS _{max}	MAS _{avg}	Точное значение
1	5514	5514	5514	5514
2	5539	5539	5539	5539
3	5827	5827	5827	5825
4	6456	6461	6459	6455
5	5265	5265	5265	5265
6	6402	6402	6402	6391
7	4774	4775	4775	4750
8	5175	5175	5175	5170
9	8292	8292	8292	8292
10	4208	4208	4208	4208

Таблица 5

**Сравнение результатов MAS без перебора
с точным решением**

№	MAS _{min}	MAS _{max}	MAS _{avg}	Точное значение
1	5547	5547	5547	5514
2	5598	5610	5601	5539
3	5849	5849	5849	5825
4	6540	6576	6566	6455
5	5265	5272	5271	5265
6	6421	6421	6421	6391
7	4785	4805	4789	4750
8	5264	5175	5292	5170
9	8308	8308	8308	8292
10	4208	4208	4208	4208

4.3. Сравнение с точным решением. Также проведено сравнение решений полученных алгоритмом MAS с точными решениям на маленьких примерах, на которых возможен полный перебор в подзадачах одного товара (около 70 складов и 400 переменных в сумме). Результаты данного сравнения приведены в табл. 4.

Видно, что в 5 из 10 примеров алгоритм MAS нашёл точное решение, а в остальных случаях отклонение составило не более 0,5%.

Однако стоит заметить, что алгоритм MAS включает в себя полный перебор в тех подзадачах одного товара, в которых число вариантов мало. Если полностью отключить перебор, то получатся результаты, представленные в табл. 5.

В данном случае оптимальное решение находится только в двух примерах, а максимальное отклонение возрастает до 1,88%.

Заключение

Для задачи управления запасами в двухэшелонной системе складов разработан алгоритм декомпозиции, основанный на многоагентной схеме. Алгоритм использует задачу координатора и однотипные задачи рабочих агентов. Каждый рабочий агент оптимизирует затраты на хранение одного товара на всех складах при заданных уровнях удовлетворения заказов в обоих эшелонах. Координатор задаёт эти уровни агентам и строит решение исходной задачи из решений рабочих агентов, используя модель о многовариантном рюкзаке. Для вычисления целевой функции и ограничений рабочие агенты используют детерминированную имитационную модель. Показано, что такой подход гарантирует получение

точного решения задачи, если точно решать подзадачи для каждого товара и координатор правильно выбрал уровни удовлетворения заказов. Разработан итерационный алгоритм, который последовательно уточняет эти уровни. Приводится нижняя оценка минимума суммарных затрат, которая позволяет строить точные алгоритмы. Проведены вычислительные эксперименты на примерах большой размерности. Эксперименты показали высокую эффективность предложенного подхода и его превосходство над пакетами Nevergrad и МЕО. На примерах с известным точным решением в двух случаях удалось найти оптимум, в остальных случаях отклонение от оптимума составило не более 1,9%.

Финансирование работы

Исследование выполнено в рамках гос. задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Wassick J. M., Agarwal A., Akiya N. [et al.]. Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products // *Comput. Chem. Eng.* 2012. V. 47. P. 157–169. DOI: 10.1016/j.compchemeng.2012.06.041.
2. Grossmann I. Enterprise-wide optimization: A new frontier in process systems engineering // *AIChE J.* 2005. V. 51, No. 7. P. 1846–1857. DOI: 10.1002/aic.10617.
3. Zhang S., Huang K., Yuan Y. Spare parts inventory management: A literature review // *Sustainability (Switz.)*. 2021. V. 13, No. 5. Paper ID 2460. 23 p. DOI: 10.3390/su13052460.
4. Lee H. L. A multi-echelon inventory model for repairable items with emergency lateral transshipments // *Manag. Sci.* 1987. V. 33, No. 10. P. 1302–1316. DOI: 10.1287/mnsc.33.10.1302.
5. Axsäter S. Modelling emergency lateral transshipments in inventory systems // *Manag. Sci.* 1990. V. 36, No. 11. P. 1329–1338. DOI: 10.1287/mnsc.36.11.1329.
6. Wong H., van Houtum G. J., Cattrysse D., Van Oudheusden D. Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints // *J. Oper. Res. Soc.* 2005. V. 56, No. 12. P. 1419–1430. DOI: 10.1057/palgrave.jors.2601952.
7. Yue D., You F. Planning and scheduling of flexible process networks under uncertainty with stochastic inventory: MINLP models and algorithm // *AIChE J.* 2013. V. 59, No. 5. P. 1511–1532. DOI: 10.1002/aic.13924.

8. **Zapata J. C., Pekny J., Reklaitis G. V.** Simulation-optimization in support of tactical and strategic enterprise decisions // Planning production and inventories in the extended enterprise. A state of the art handbook. V. 1. New York: Springer, 2011. P. 593–627. (Int. Ser. Oper. Res. Manag. Sci.; V. 151). DOI: 10.1007/978-1-4419-6485-4_20.
9. **Peidro D., Mula J., Poler R., Lario F. C.** Quantitative models for supply chain planning under uncertainty // Int. J. Adv. Manuf. Technol. 2009. V. 43. P. 400–420. DOI: 10.1007/s00170-008-1715-y.
10. **Köchel P., Nieländer U.** Simulation-based optimisation of multi-echelon inventory systems // Int. J. Prod. Econ. 2005. V. 93–94. P. 505–513. DOI: 10.1016/j.ijpe.2004.06.046.
11. **Chu Y., You F., Wassick J. M., Agarwal A.** Simulation-based optimization framework for multi-echelon inventory systems under uncertainty // Comput. Chem. Eng. 2015. V. 73. P. 1–16. DOI: 10.1016/j.compchemeng.2014.10.008.
12. **Mele F. D., Guillén G., Espuña A, Puigjaner L.** A simulation-based optimization framework for parameter optimization of supply-chain networks // Ind. Eng. Chem. Res. 2006. V. 45, No. 9. P. 3133–3148. DOI: 10.1021/ie051121g.
13. **Moncayo-Martínez L. A., Zhang D. Z.** Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design // Int. J. Prod. Econ. 2011. V. 131, No. 1. P. 407–420. DOI: 10.1016/j.ijpe.2010.11.026.
14. **Nikolopoulou A., Ierapetritou M. G.** Hybrid simulation based optimization approach for supply chain management // Comput. Chem. Eng. 2012. V. 47. P. 183–193. DOI: 10.1016/j.compchemeng.2012.06.045.
15. **Blum C., Puchinger J., Raidl G. R., Roli A.** Hybrid metaheuristics in combinatorial optimization: A survey // Appl. Soft Comput. 2011. V. 11, No. 6. P. 4135–4151. DOI: 10.1016/j.asoc.2011.02.032.
16. **Noordhoek M., Dullaert W., Lai D. S. W., de Leeuw S.** A simulation-optimization approach for a service-constrained multi-echelon distribution network // Transp. Res. Part E: Logist. Transp. Rev. 2018. V. 114. P. 292–311. DOI: 10.1016/j.tre.2018.02.006.
17. **Chen H., Dai B., Li Y.** [et al.]. Stock allocation in a two-echelon distribution system controlled by (s, S) policies // Int. J. Prod. Res. 2022. V. 60, No. 3. P. 894–911. DOI: 10.1080/00207543.2020.1845915.
18. **Rapin J., Teytaud O.** Nevergrad — A gradient-free optimization platform. 2018. URL: [GitHub.com/FacebookResearch/Nevergrad](https://github.com/facebookresearch/nevergrad) (accessed: 24.01.2024).
19. **Yuskov A. D., Kulachenko I. N., Melnikov A. A., Kochetov Yu. A.** Decomposition approach for simulation-based optimization of inventory management // Mathematical optimization theory and operations research: Recent trends. Rev. Sel. Pap. 22nd Int. Conf. (Yekaterinburg, Russia, July 2–8, 2023). Cham: Springer, 2023. P. 259–273. (Commun. Comput. Inf. Sci.; V. 1881). DOI: 10.1007/978-3-031-43257-6_20.

20. **Bajaj I., Arora A., Hasan M. M. F.** Black-box optimization: Methods and applications // Black box optimization, machine learning, and no-free lunch theorems. Cham: Springer, 2021. P. 35–65. (Springer Optim. Its Appl.; V. 170). DOI: 10.1007/978-3-030-66515-9_2.
21. **Bezanson J., Edelman A., Karpinski S., Shah V. B.** Julia: A fresh approach to numerical computing // SIAM Rev. 2017. V. 59, No. 1. P. 65–98. DOI: 10.1137/141000671.
22. **Topan E., Bayındır Z. P., Tan T.** Heuristics for multi-item two-echelon spare parts inventory control subject to aggregate and individual service measures // Eur. J. Oper. Res. 2017. V. 256. P. 126–138. DOI: 10.1016/j.ejor.2016.06.012.
23. **Thonemann U. W., Brown A. O., Hausman W. H.** Easy quantification of improved spare parts inventory policies // Manag. Sci. 2002. V. 48, No. 9. P. 1213–1225. DOI: 10.1287/mnsc.48.9.1213.173.

Юськов Александр Дмитриевич
Кулаченко Игорь Николаевич
Мельников Андрей Андреевич
Кочетов Юрий Андреевич

Статья поступила
25 января 2024 г.
После доработки —
10 марта 2024 г.
Принята к публикации
22 июня 2024 г.

DECOMPOSITION APPROACH FOR A TWO ECHELON INVENTORY MANAGEMENT SYSTEM

A. D. Yuskov^{1,a}, I. N. Kulachenko^{2,b},
A. A. Melnikov^{2,c}, and Yu. A. Kochetov^{2,d}

¹ Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia

² Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia

E-mail: ^aa.yuskov@ng.su.ru, ^bink@math.nsc.ru,
^cmelnikov@math.nsc.ru, ^djkochet@math.nsc.ru

Abstract. Warehouses of the first echelon in a two-echelon system are designed to satisfy customer orders. In the second echelon, we have a central warehouse for restocking the first echelon warehouses. Customer orders can be partially satisfied, but the total fraction of completed orders should not be less than the specified threshold. We need to minimize the total cost of storing the items in all warehouses. We use a deterministic simulation to calculate the order satisfaction ratio and the storage cost during the planning period. The simulation depends on inventory management policies at each warehouse for each type of items. We develop a decomposition method for solving the problem. It is based on solution of subproblems for each type of items. Also, we propose some approaches for exact solution of the problem. The results of numerical experiments with instances with 100 warehouses and 1000 types of items are presented. On instances with known exact solutions, we have the optimum in two cases, while in the other cases the deviation from the optimal values is at most 1.9%. Tab. 5, illustr. 1, bibliogr. 23.

Keywords: gray-box optimization, knapsack problem, local search.

References

1. **J. M. Wassick, A. Agarwal, N. Akiya, [et al.]**, Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products, *Comput. Chem. Eng.* **47**, 157–169 (2012), DOI: 10.1016/j.compchemeng.2012.06.041.
2. **I. Grossmann**, Enterprise-wide optimization: A new frontier in process systems engineering, *AIChE J.* **51** (7), 1846–1857 (2005), DOI: 10.1002/aic.10617.
3. **S. Zhang, K. Huang, and Y. Yuan**, Spare parts inventory management: A literature review, *Sustainability (Switz.)* **13** (5), ID 2460 (2021), DOI: 10.3390/su13052460.
4. **H. L. Lee**, A multi-echelon inventory model for repairable items with emergency lateral transshipments, *Manag. Sci.* **33** (10), 1302–1316 (1987), DOI: 10.1287/mnsc.33.10.1302.
5. **S. Axsäter**, Modelling emergency lateral transshipments in inventory systems, *Manag. Sci.* **36** (11), 1329–1338 (1990), DOI: 10.1287/mnsc.36.11.1329.
6. **H. Wong, G. J. van Houtum, D. Cattrysse, and D. Van Oudheusden**, Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints, *J. Oper. Res. Soc.* **56** (12), 1419–1430 (2005), DOI: 10.1057/palgrave.jors.2601952.
7. **D. Yue and F. You**, Planning and scheduling of flexible process networks under uncertainty with stochastic inventory: MINLP models and algorithm, *AIChE J.* **59** (5), 1511–1532 (2013), DOI: 10.1002/aic.13924.
8. **J. C. Zapata, J. Pekny, and G. V. Reklaitis**, Simulation-optimization in support of tactical and strategic enterprise decisions, in *Planning Production and Inventories in the Extended Enterprise. A State of the Art Handbook*, Vol. 1 (Springer, New York, 2011), pp. 593–627 (Int. Ser. Oper. Res. Manag. Sci., Vol. 151), DOI: 10.1007/978-1-4419-6485-4_20.
9. **D. Peidro, J. Mula, R. Poler, F. C. Lario**, Quantitative models for supply chain planning under uncertainty, *Int. J. Adv. Manuf. Technol.* **43**, 400–420 (2009), DOI: 10.1007/s00170-008-1715-y.
10. **P. Köchel and U. Nieländer**, Simulation-based optimisation of multi-echelon inventory systems, *Int. J. Prod. Econ.* **93–94**, 505–513 (2005), DOI: 10.1016/j.ijpe.2004.06.046.
11. **Y. Chu, F. You, J. M. Wassick, and A. Agarwal**, Simulation-based optimization framework for multi-echelon inventory systems under uncertainty, *Comput. Chem. Eng.* **73**, 1–16 (2015), DOI: 10.1016/j.compchemeng.2014.10.008.
12. **F. D. Mele, G. Guillén, España A and L. Puigjaner**, A simulation-based optimization framework for parameter optimization of supply-chain networks, *Ind. Eng. Chem. Res.* **45** (9), 3133–3148 (2006), DOI: 10.1021/ie051121g.
13. **L. A. Moncayo-Martínez and D. Z. Zhang**, Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design, *Int. J. Prod. Econ.* **131** (1), 407–420 (2011), DOI: 10.1016/j.ijpe.2010.11.026.

14. **A. Nikolopoulou** and **M. G. Ierapetritou**, Hybrid simulation based optimization approach for supply chain management, *Comput. Chem. Eng.* **47**, 183–193 (2012), DOI: 10.1016/j.compchemeng.2012.06.045.
15. **C. Blum**, **J. Puchinger**, **G. R. Raidl**, and **A. Roli**, Hybrid metaheuristics in combinatorial optimization: A survey, *Appl. Soft Comput.* **11** (6), 4135–4151 (2011), DOI: 10.1016/j.asoc.2011.02.032.
16. **M. Noordhoek**, **W. Dullaert**, **D. S. W. Lai**, and **S. de Leeuw**, A simulation–optimization approach for a service-constrained multi-echelon distribution network, *Transp. Res., Part E: Logist. Transp. Rev.* **114**, 292–311 (2018), DOI: 10.1016/j.tre.2018.02.006.
17. **H. Chen**, **B. Dai**, **Y. Li**, [et al.]. Stock allocation in a two-echelon distribution system controlled by (s, S) policies, *Int. J. Prod. Res.* **60** (3), 894–911 (2022), DOI: 10.1080/00207543.2020.1845915.
18. **J. Rapin** and **O. Teytaud**, Nevergrad — A gradient-free optimization platform. 2018. URL: [GitHub.com/FacebookResearch/Nevergrad](https://github.com/facebookresearch/nevergrad) (accessed: 24.01.2024).
19. **A. D. Yuskov**, **I. N. Kulachenko**, **A. A. Melnikov**, and **Yu. A. Kochetov**, Decomposition approach for simulation-based optimization of inventory management, in *Mathematical Optimization Theory and Operations Research: Recent Trends* (Rev. Sel. Pap. 22nd Int. Conf., Yekaterinburg, Russia, July 2–8, 2023) (Springer, Cham, 2023), pp. 259–273 (Commun. Comput. Inf. Sci., Vol. 1881), DOI: 10.1007/978-3-031-43257-6_20.
20. **I. Bajaj**, **A. Arora**, and **M. M. F. Hasan**, Black-box optimization: Methods and applications, in *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems* (Springer, Cham, 2021), pp. 35–65 (Springer Optim. Its Appl., Vol. 170), DOI: 10.1007/978-3-030-66515-9_2.
21. **J. Bezanson**, **A. Edelman**, **S. Karpinski**, and **V. B. Shah**, Julia: A fresh approach to numerical computing, *SIAM Rev.* **59** (1), 65–98 (2017), DOI: 10.1137/141000671.
22. **E. Topan**, **Z. P. Bayındır** and **T. Tan**, Heuristics for multi-item two-echelon spare parts inventory control subject to aggregate and individual service measures, *Eur. J. Oper. Res.* **256**, 126–138 (2017), DOI: 10.1016/j.ejor.2016.06.012.
23. **U. W. Thonemann**, **A. O. Brown**, and **W. H. Hausman**, Easy quantification of improved spare parts inventory policies, *Manag. Sci.* **48** (9), 1213–1225 (2002), DOI: 10.1287/mnsc.48.9.1213.173.

Aleksandr D. Yuskov
Igor N. Kulachenko
Andrey A. Melnikov
Yury A. Kochetov

Received January 25, 2024
Revised March 10, 2024
Accepted June 22, 2024

СОДЕРЖАНИЕ ТОМА 31

№ 1

Баротов Д. Н. Выпуклое продолжение булевой функции и его приложения	5
Ключин А. А., Кожухов И. Б., Манилов Д. Ю., Решетников А. В. Определяемость отношений полугруппами изотонных преобразований	19
Ковалевский А. П. Вероятностный подход к игре в угадывание в случайной среде	35
Малыгина Е. С., Куценко А. В., Новосёлов С. А., Колесников Н. С., Бахарев А. О., Хильчук И. С., Шапоренко А. С., Токарева Н. Н. Постквантовые криптосистемы: открытые вопросы и существующие решения. Криптосистемы на изогениях и кодах, исправляющих ошибки	52
Монахова Э. А., Монахов О. Г. Метод автоматического поиска семейств оптимальных хордальных кольцевых сетей	85
Талецкий Д. С. О количестве k -доминирующих независимых множеств в планарных графах	109

№ 2

Балашов В. В., Абрамов А. В., Чупахин А. А., Туркин А. В., Гао Ц., Сунь Ш., Чжоу Л., Сунь Ц. Муравьиный алгоритм для построения однопроцессорного расписания с минимизацией пикового использования ресурса	5
Береснев В. Л., Мельников А. А., Утюпин С. Ю. Устойчивость вершинных покрытий в игре с конечным числом шагов	28
Гусев В. В. Кооперативные игры с предпочтениями: приложение веса правила к проблемам общественного пространства в городе Санкт-Петербург	46
Ефимов Д. Б. Перечисление чётных и нечётных хордовых диаграмм ..	63
Зоркальцев В. И., Князев А. С. Сравнительный анализ алгоритмов для оценки динамики численности рыб	80

Зуев Ю. А., Ключарёв П. Г. Способ оптимального измерения значений некоторых параметров fault-атак на криптографические алгоритмы	96
Мокроусов А. С., Коломеец Н. А. Разности по модулю 2^n для ARX-преобразований, вероятность которых больше $1/4$	108
Пяткин А. В. О сложности задачи выбора кластеров большого размера	136
Фомичёв В. М. О сложности метода последовательного опробования ..	144

№ 3

Баротов Д. Н. Выпуклые продолжения некоторых дискретных функций	5
Бахарев А. О. Новая модель квантового оракула для гибридной квантово-классической атаки на постквантовые криптосистемы, основанные на решётках	24
Бородин С. О., Тараненко А. А. Совершенные раскраски гиперграфа подматриц	54
Водян М. Е., Панин А. А., Плясунов А. В. Исследование пороговой устойчивости двухуровневой задачи размещения производства и дискриминационного ценообразования	79
Ляшкова К. А., Сервах В. В. Задача одного станка с равными длительностями работ и возможностью прерываний	105
Маматов А. Р. Поиск локально оптимальных стратегий в линейной игровой задаче с благоприятными ситуациями	123
Пяткин А. В. О максимальном числе открытых треугольников в графах с малым числом рёбер	144

№ 4

Береснев В. Л., Мельников А. А. Алгоритм ветвей, границ и отсечений для неявной динамической задачи конкурентного размещения	5
Гончаров Е. Н. Жадный алгоритм для задачи календарного планирования с ограниченными ресурсами	27
Ильев В. П., Ильева С. Д., Кононов А. В. Приближённые алгоритмы для задач кластеризации на графах с кластерами небольшого размера	40
Коломеец Н. А. Число невозможных разностей по модулю 2^n композиции XOR и циклического сдвига битов	58

Назаренко С. А. Уточнённые оценки для алгоритмов упаковки 2-гистограмм в полосу	90
Панин А. А., Пискеева Д. А., Плясунов А. В. Спуск с чередующимися окрестностями для поиска радиуса пороговой устойчивости в задаче размещения и дискриминационного ценообразования	116
Савенкова Н. П., Мокин А. Ю., Дряженков А. А., Артемова Л. А. Алгоритм поиска решения задачи календарного планирования на производстве	134
Сергеев И. С. О сложности параллельных префиксных схем с ограниченным ветвлением	151
Тамасян Г. Ш., Шульга Г. С. Методы негладкого анализа в задаче минимизации суммы модулей от аффинных функций	168
Юськов А. Д., Кулаченко И. Н., Мельников А. А., Кочетов Ю. А. Метод декомпозиции для управления запасами в двухэшелонной системе складов	186

DISKRETNYYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/

CONTENTS OF VOLUME 31

No. 1

D. N. Barotov. <i>Convex continuation of a Boolean function and its applications</i>	5
A. A. Klyushin, I. B. Kozhukhov, D. Yu. Manilov, and A. V. Reshetnikov. <i>Definability of relations by semigroups of isotone transformations</i>	19
A. P. Kovalevskii. <i>A probabilistic approach to the game of guessing in a random environment</i>	35
E. S. Malygina, A. V. Kutsenko, S. A. Novoselov, N. S. Kolesnikov, A. O. Bakharev, I. S. Khilchuk, A. S. Shaporenko, and N. N. Tokareva. <i>Post-quantum cryptosystems: Open problems and solutions. Isogeny-based and code-based cryptosystems</i>	52
E. A. Monakhova and O. G. Monakhov. <i>A method for automatic search for families of optimal chordal ring networks</i>	85
D. S. Taletskii. <i>On the number of k-dominating independent sets in planar graphs</i>	109

No. 2

V. V. Balashov, A. V. Abramov, A. A. Chupakhin, A. V. Turkin, J. Gao, C. Sun, L. Zhou, and J. Sun. <i>Ant colony algorithm for single processor scheduling with minimization of peak resource usage</i>	5
V. L. Beresnev, A. A. Melnikov, and S. Yu. Utyupin. <i>Stability of vertex covers in a game with finitely many steps</i>	28
V. V. Gusev. <i>Cooperative games with preferences: Application of the weight rule to problems of public space in St. Petersburg</i>	46
D. B. Efimov. <i>Enumeration of even and odd chord diagrams</i>	63
V. I. Zorkaltsev and A. S. Knyazev. <i>Comparative analysis of algorithms for estimating fish population dynamics</i>	80
Yu. A. Zuev and P. G. Klyucharev. <i>A method for optimal measurement of some parameters of fault attacks on cryptographic algorithms</i>	96
A. S. Mokrousov and N. A. Kolomeec. <i>Additive differentials for ARX mappings with probability exceeding $1/4$</i>	108

A. V. Pyatkin. <i>On the complexity of the problem of choice of large clusters</i> . . .	136
V. M. Fomichev. <i>On the complexity of the sequential sampling method . . .</i>	144

No. 3

D. N. Barotov. <i>Convex continuations of some discrete functions</i>	5
A. O. Bakharev. <i>A new quantum oracle model for a hybrid quantum-classical attack on post-quantum lattice-based cryptosystems .</i>	24
S. O. Borodin and A. A. Taranenko. <i>Perfect colorings of submatrix hypergraphs</i>	54
M. E. Vodyan, A. A. Panin, and A. V. Plyasunov. <i>A study of the threshold stability of the bilevel problem of facility location and discriminatory pricing</i>	79
K. A. Lyashkova and V. V. Servakh. <i>The problem of one machine with equal processing time and preemption</i>	105
A. R. Mamatov. <i>Search for locally optimal strategies in a linear game problem with favorable situations</i>	123
A. V. Pyatkin. <i>On the maximum number of open triangles in graphs with few edges</i>	144

No. 4

V. L. Beresnev and A. A. Melnikov. <i>A branch, bound, and cuts algorithm for the dynamic competitive facility location problem</i>	5
E. N. Goncharov. <i>Greedy algorithm for the resource-constrained project scheduling problem</i>	27
V. P. Il'ev, S. D. Il'eva, and A. V. Kononov. <i>Approximation algorithms for graph clustering problems with clusters of bounded size</i>	40
N. A. Kolomeec. <i>The number of impossible additive differentials for the composition of XOR and bit rotation</i>	58
S. A. Nazarenko. <i>Updated estimates for algorithms for packing 2-bar charts in a strip</i>	90
A. A. Panin, D. A. Piskeeva, and A. V. Plyasunov. <i>Variable neighborhood descent for finding the threshold stability radius in the facility location and discriminatory pricing problem</i>	116
N. P. Savenkova, A. Yu. Mokin, A. A. Dryazhenkov, and L. A. Artemyeva. <i>An algorithm to search a solution to the production scheduling problem</i>	134

I. S. Sergeev. <i>On the complexity of fanout-bounded parallel prefix circuits</i>	151
G. Sh. Tamasyan and G. S. Shulga. <i>Methods of nonsmooth analysis in application to the problem of minimizing the sum of affine functions' modules</i>	168
A. D. Yuskov, I. N. Kulachenko, A. A. Melnikov, and Yu. A. Kochetov. <i>Decomposition approach for a two echelon inventory management system</i>	186

ДИСКРЕТНЫЙ АНАЛИЗ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

2024. Том 31, № 4

Зав. редакцией Ю. В. Шамардин

Журнал подготовлен с использованием макропакета $\text{\LaTeX 2}_{\epsilon}$.

The present publication has been typeset using $\text{\LaTeX 2}_{\epsilon}$.

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 18.06.2025 г.

Формат 70×100 1/16. Усл. печ. л. 17,8. Объём 1,97 МБ.

Издательство Института математики,
пр. Академика Коптюга, 4, 630090 Новосибирск, Россия