

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 31 № 2 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, № 2 (160)	Апрель–июнь 2024	

СОДЕРЖАНИЕ

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 весового 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

НОВОСИБИРСК
ИЗДАТЕЛЬСТВО ИНСТИТУТА МАТЕМАТИКИ

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

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

Адрес редакции:

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

© 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. 2 (160)		April–June, 2024

CONTENTS

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

NOVOSIBIRSK
SOBOLEV INSTITUTE PRESS

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

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

Editorial office address:

Sobolev Institute of Mathematics,
4 Acad. Koptug 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

МУРАВЬИНЫЙ АЛГОРИТМ ДЛЯ ПОСТРОЕНИЯ
ОДНОПРОЦЕССОРНОГО РАСПИСАНИЯ
С МИНИМИЗАЦИЕЙ ПИКОВОГО ИСПОЛЬЗОВАНИЯ
РЕСУРСА

В. В. Балашов^{1, a}, А. В. Абрамов^{1, b}, А. А. Чупахин^{1, c},
А. В. Туркин^{2, d}, Ц. Гао^{2, e}, Ш. Сунь^{3, f}, Л. Чжоу^{3, g}, Ц. Сунь^{3, h}

¹ Московский гос. университет им. М. В. Ломоносова,
Ленинские горы, 1, 119991 Москва, Россия

² Московский исследовательский центр Хуавэй,
Смоленская пл., 7-9, 121099 Москва, Россия

³ Гонконгский исследовательский центр Хуавэй,
Сайнс Парк Вест Авеню, 2, 8/F, 999077 Гонконг, КНР

E-mail: ^ahbd@cs.msu.ru, ^bavabramov@lvk.cs.msu.ru,
^candrewchup@lvk.cs.msu.ru, ^dandrei.turkin@huawei.com,
^egaojiexing@huawei.com, ^fsunchumin@huawei.com,
^gzhouli107@huawei.com, ^hj.sun@huawei.com

Аннотация. Рассматривается задача построения однопроцессорного расписания с минимизацией пикового использования ресурса вычислителя. В качестве ресурса может выступать оперативная память. Набор заданий для планирования представлен в виде ориентированного ациклического графа, в каждой вершине которого указан объём занимаемого соответствующим заданием ресурса. Высвобождение ресурса, выделенного заданию, происходит по завершении последнего (согласно расписанию) непосредственного потомка этого задания в графе. Ограничением корректности на расписание является соблюдение частичного порядка, определённого графом заданий. Длительности заданий не рассматриваются. Приводится формальная постановка задачи. В качестве алгоритма для её решения предлагается муравьиный алгоритм, модифицированный так, что матрица феромона отражает желательность взаимного расположения (порядка) в расписании для любой пары заданий, а не только для пар соседних заданий. При построении расписания алгоритм для каждого задания выбирает его позицию в расписании, в отличие от известных муравьиных алгоритмов, которые строят расписание в порядке увеличения номеров позиций («слева направо») и для каждой очередной позиции выбирают задание.

Выполнено экспериментальное исследование алгоритма на двух наборах заданий. Графы из первого набора построены так, чтобы априорно была известна оценка оптимального значения целевой функции. Графы из второго набора «слоистые», что соответствует структуре приложений по многостадийной обработке данных. В рамках каждого из наборов графы сформированы случайным образом, с соблюдением заданных параметров генерации и ограничений на структуру графа. Экспериментальное исследование показало высокую точность и стабильность предложенного муравьиного алгоритма. Табл. 1, ил. 12, библиогр. 17.

Ключевые слова: комбинаторная оптимизация, однопроцессорное расписание, минимизация ресурса, муравьиный алгоритм.

Введение

Эффективность использования ресурсов в вычислительных системах (ВС) тесно связана с планированием заданий. Частью указанного процесса является решение задачи составления расписаний. Традиционно решение таких задач направлено на минимизацию общего времени выполнения всех заданий, но в ряде случаев наиболее критичными являются другие ресурсы, такие как оперативная память. Таким образом, при построении ВС возникает вопрос: как определить необходимый для ВС объём ресурсов, которого точно хватит для организации вычислительного процесса и не окажется в избытке?

В настоящей работе рассматривается проблема построения расписания, оптимального с точки зрения уменьшения пикового потребления памяти ВС. ВС состоит из одного процессора, способного в каждый момент времени обрабатывать лишь одно задание, и некоторого объёма памяти. На вход алгоритму планирования подаётся связный ориентированный ациклический граф заданий, в котором рёбра представляют собой зависимости по данным между ними. Таким образом, граф задаёт отношение частичного порядка на множестве заданий. Каждой вершине в графе сопоставлено неотрицательное целое число — объём памяти, занимаемый результатом выполнения задания, соответствующего этой вершине. Результат выполнения задания хранится в памяти до тех пор, пока не выполнятся все непосредственные потомки данного задания, т. е. такие вершины в графе, которые связаны ребром с вершиной, соответствующей данному заданию. Все задания обрабатываются на процессоре без прерываний. Необходимо определить сквозной порядок заданий, который минимизирует пиковое потребление памяти ВС. При этом сквозной порядок должен удовлетворять отношению частичного порядка, заданному графом. Длительности заданий при планировании не учитываются.

Описанная задача является задачей о минимизации максимальной совокупной стоимости (ММСС) [1]. В [2, 3] показано, что в общем случае задача ММСС NP-трудна. В настоящий момент неизвестны полиномиальные алгоритмы точного решения NP-трудных задач. В этой работе предложен алгоритм, относящийся к классу алгоритмов муравьиных колоний, которые берут за основу принципы взаимодействия муравьёв в естественной среде, а именно их роевой интеллект. Каждый отдельный муравей обладает информацией только о локальной обстановке, ни один из них не имеет представления обо всей ситуации в целом — только о том, что узнал от своих сородичей явно или неявно. На неявных взаимодействиях муравьёв основаны механизмы поиска кратчайшего пути от муравейника до источника пищи. Каждый раз, проходя по такому пути, муравей оставляет за собой дорожку феромонов. Другие муравьи, почувствовав такие следы на земле, будут рефлекторно устремляться к нему. Эти муравьи тоже оставляют за собой дорожки феромонов, а значит, чем больше муравьёв проходит по определённому пути, тем более привлекательным он становится для их сородичей. При этом чем короче путь до источника пищи, тем меньше времени требуется муравьям на него, а следовательно, тем быстрее оставленные на нём следы становятся заметными.

Муравьиные алгоритмы (МА) нашли широкое применение в приближённом решении NP-трудных задач, особенно задач на графах. Так, в работах [4, 5] рассматривается задача минимизации суммарного запаздывания на одном приборе при помощи муравьиного алгоритма, описываются подходы локального поиска (перестановки и замены) для улучшения решений, найденных при помощи МА. В [6] приведено описание сбалансированного алгоритма муравьиных колоний для планирования вычислений в Grid-системах. В [7] рассматривается гибридный муравьиный алгоритм для проектного планирования при условии ограниченности ресурсов. В [8] предложен муравьиный алгоритм для отображения графов ресурсных запросов на граф физических ресурсов центра обработки данных.

Часто МА показывает хорошие результаты, которые можно улучшить при помощи различных модификаций, например, локального поиска или правил исключения [5]. В [9] приведено сравнение МА с различными метаэвристическими алгоритмами, в числе которых алгоритм имитации отжига и генетический алгоритм, и установлено, что для задач минимизации суммарного запаздывания больших размерностей МА показывает наиболее хорошие результаты.

Ниже приводится муравьиный алгоритм для решения задачи минимизации максимальных затрат. Разд. 1 отведён математической постановке задачи. В разд. 2 приводится описание предложенного алгоритма.

Результаты численных экспериментов содержатся в разд. 3. В заключении подведены итоги и указаны направления дальнейших исследований.

1. Постановка задачи

Модель прикладной программы может быть представлена как ориентированный ациклический граф $G = (V, E)$, $|V| = n$, $|E| = m$. Каждой вершине графа соответствует задание p_i , $1 \leq i \leq n$, каждой дуге — передача данных между заданиями. Если $(p_i, p_j) \in E$, $1 \leq i, j \leq n$, то для выполнения задания p_j необходим результат выполнения задания p_i .

Вычислительная среда состоит из одного процессора SP, который в каждый момент времени способен выполнять только одно задание, и уникального ресурса SR. Каждое задание выполняется на процессоре SP без прерываний. В момент старта задание p_i , $1 \leq i \leq n$, захватывает $r_{p_i} \geq 0$ ресурса SR для хранения результата. При завершении задания p_i происходит высвобождение ресурса, занимаемого такими заданиями p_j , $1 \leq j \leq n$, $(p_j, p_i) \in E$, для которых p_i является последним выполненным потомком.

Расписание НР сформировано, если для всех заданий из G определён порядок их выполнения на процессоре SP. Фактически НР представляет собой перестановку π номеров заданий из G (будем считать, что задания пронумерованы), поэтому ниже понятия «расписание» и «перестановка заданий» будем считать взаимозаменяемыми:

$$\pi = (j_1, j_2, \dots, j_n),$$

где j_i — номер задания, находящегося в i -й позиции расписания.

Расписание НР корректно, если выполнены следующие ограничения.

1*. Каждое задание назначено на процессор SP.

2*. Процессор SP в каждый момент времени выполняет не более одного задания.

3*. Частичный порядок, заданный графом зависимостей G , сохранён в НР.

Далее будем говорить, что расписание допустимо НР $\in \text{НР}_{1-3}^*$, если оно удовлетворяет ограничениям 1*–3*.

Минимизируемой целевой функцией (стоимостью в терминах задачи ММСС) является максимальное по всему расписанию количество занятого в ВС ресурса. Длительности заданий в рамках задачи не рассматриваются.

Пусть задано расписание НР $\in \text{НР}_{1-3}^*$. Опишем процедуру расчёта целевой функции для НР. Обозначим через $f_{\text{НР}}^k$, $1 \leq k \leq n$, количество занятого в ВС ресурса для k -й позиции в расписании НР. Пусть A — множество заданий, расположенных в расписании НР на позициях $1, 2, \dots, k$. Пусть $B \subset A$ — множество заданий, расположенных в расписании НР

на позициях $1, 2, \dots, k-1$, у каждого из которых все потомки расположены на позициях от 1 до $k-1$. Тогда $f_{\text{НР}}^k$ вычисляется по формуле

$$f_{\text{НР}}^k = \sum_{p_j \in A} r_{p_j} - \sum_{p_i \in B} r_{p_i}.$$

Пусть известны значения $\{f_{\text{НР}}^k\}_{k=1}^n$ для расписания НР. Тогда значение целевой функции $f_{\text{НР}}$ для такого расписания равно

$$f_{\text{НР}} = \max_{1 \leq k \leq n} f_{\text{НР}}^k. \quad (1)$$

Требуется для модели прикладной программы G найти такое расписание $\text{НР}_* \in \text{НР}_{1-3}^*$, для которого достигается минимальное значение целевой функции:

$$f_{\text{НР}_*} = \min_{\text{НР} \in \text{НР}_{1-3}^*} f_{\text{НР}}.$$

Прежде чем переходить к предлагаемому алгоритму, ответим на вопрос о применимости для решения поставленной задачи ряда известных программ-решателей задач, как коммерческих (с доступной академической лицензией), так и свободно распространяемых. Ограничения задачи задаются графом зависимостей G , определяющим частичный порядок заданий. Если в качестве переменных задачи взять позиции заданий в расписании, то ограничения частичного порядка принимают линейный вид. Однако авторам задачи не известно представления целевой функции задачи ни в линейном, ни в квадратичном виде. В связи с указанной спецификой целевой функции для решения задачи не подходят такие программы-решатели, как Gurobi [10], MOSEK [11], SCIP [12], решатели от COIN-OR [13].

2. Описание предложенного алгоритма

2.1. Структура матрицы феромонного следа. Прежде чем перейти к описанию предложенного алгоритма, необходимо рассмотреть некоторые особенности решаемой задачи и обсудить применимость стандартного муравьиного алгоритма.

Стандартные муравьиные алгоритмы хорошо зарекомендовали себя в решении задач на графах таких, как, например, задача коммивояжёра [14]. Главной причиной, по которой МА оказываются столь эффективными, является дизайн и применение структуры, хранящей информацию о количестве феромона на рёбрах графа. Если рассматривать большинство стандартных задач построения расписаний, решаемых при помощи МА, то можно обратить внимание на то, что переход по каждому ребру вносит вполне определённый вклад в итоговую целевую функцию (ЦФ), например в длительность расписания. Другими словами, точно известно, что ребро из вершины A в вершину B имеет определённую длину,

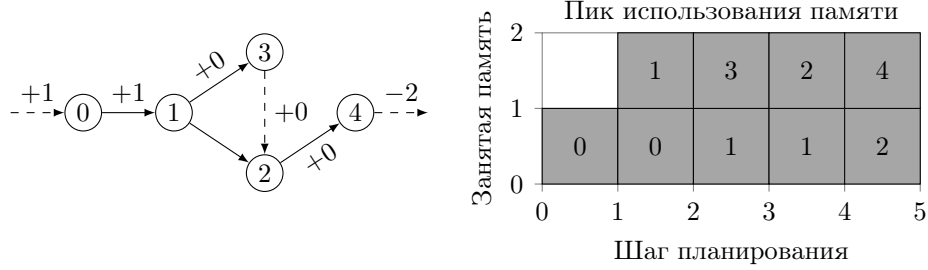


Рис. 1

поэтому вне зависимости от того, на какой позиции в последовательности переходов стоит вершина A , переход из неё в вершину B , в случае, когда он допустим, всегда будет иметь одинаковую стоимость. В случае задачи ММСС это не так. Действительно, переход по ребру $A \rightarrow B$ может оказывать различный вклад в итоговое значение ЦФ в зависимости от последовательности обхода графа. На рис. 1–2 слева изображён обход одного и того же графа в разном порядке, а справа — использование ресурса в различных позициях расписания (каждая вершина графа имеет вес 1). Сплошные линии задают частичный порядок, а пунктирные обозначают переходы между несвязанными вершинами. На каждом ребре изображено значение, которое добавляется к ЦФ при переходе по нему. Можно заметить, что переход по ребру из вершины 2 в вершину 4 оказывает различный вклад в значение ЦФ (0 и 1) в зависимости от того, была ли ранее посещена вершина 3. На диаграммах в правой части рисунков в прямоугольниках указаны номера заданий таких, что ресурс, занятый ими, в данной позиции расписания ещё не освобождён.

Из вышесказанного следует, что в рассматриваемой задаче стоимость построенного решения определяют не отдельные вершины или переходы, а их взаимодействие, т. е. их взаимный порядок. Таким образом, появляется необходимость в структуре, которая хранит информацию вида «вершина B следует после вершины A ». В этой формулировке «следует после» означает именно отношение порядка, непосредственного следования не требуется.

Как отмечено выше, расписание НР представляет собой перестановку π заданий из G . Пусть π_i — позиция вершины i в расписании НР. Рассмотрим матрицу $R \in \mathbb{R}^{n \times n}$ [15], в которой элемент на месте $(i, j) \in \{1, 2, \dots, n\}^2$ равен

$$r_{ij} = \begin{cases} 1, & \text{если } \pi_i < \pi_j, \\ 0, & \text{если } \pi_i > \pi_j. \end{cases}$$

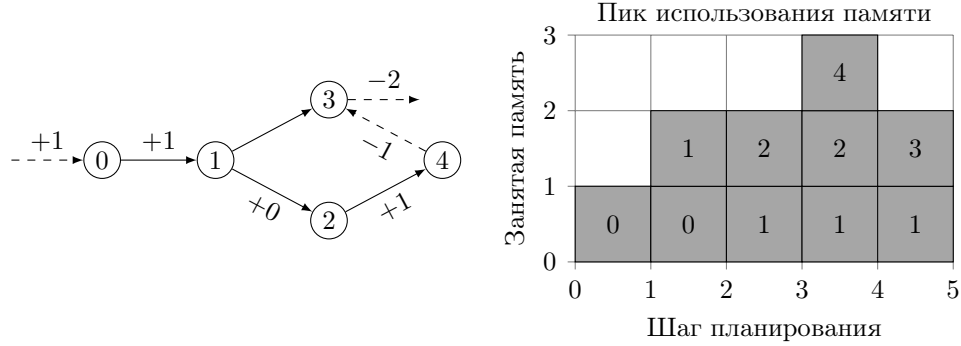


Рис. 2

Можно показать, что существует биективное отображение из множества перестановок π в множество матриц R , если абстрагироваться от допустимости конкретной перестановки (в противном случае можно составить матрицу, которая не будет удовлетворять ни одной допустимой перестановке). Пусть π изначально пуста, и задана матрица R . Будем рассматривать вершины в порядке увеличения меток и добавлять их в π . Вершина с меткой 1 первой включается в перестановку π . Для каждой следующей вершины p_i последовательно проходим по вершинам p_k из π в порядке возрастания меток; если $r_{ik} = 1$, то вставляем вершину p_i в перестановку π перед вершиной p_k , иначе переходим к следующей вершине в перестановке. Если достигнут конец перестановки π , то добавляем вершину p_i в конец перестановки.

Структура матрицы T феромонного следа в предложенном МА основывается на матрице R . Каждая ячейка матрицы τ_{ij} содержит вещественное число — информацию о том, насколько выгодным оказалось расположение заданий, при котором $\pi_i < \pi_j$, на предыдущих итерациях. Таким образом, пройдясь по строкам матрицы, для каждого задания можно определить его наиболее выгодный порядок относительно других заданий. Предложенный МА основывается на данной идее.

Более широко используемая в муравьиных алгоритмах [14] схема, в которой феромон отражает выгодность непосредственных переходов между вершинами (т. е. наличия пар вида $\pi_i \pi_j$ в составе решения), также исследовалась авторами, но продемонстрировала отсутствие сходимости. На рис. 3 показана динамика целевой функции в зависимости от номера итерации алгоритма. Слева график для алгоритма с традиционной схемой работы с феромонным следом, справа — для алгоритма со схемой, описанной в данной статье.

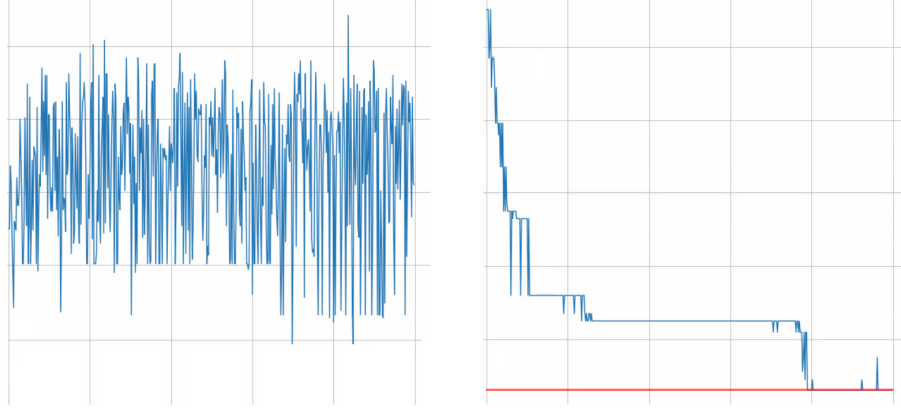


Рис. 3. Сходимость муравьиного алгоритма для различных схем работы с феромонным следом

2.2. Жадная эвристика. Как известно, МА строит решение задачи не только на основе исторической информации, содержащейся в матрице феромонного следа. При построении решения МА опирается также на некоторую жадную эвристику, которая содержит локальную информацию о выгодности текущего перехода. Предложенный МА не стал исключением. В качестве жадной эвристики в нём используется следующая стратегия. Пусть на вход алгоритму подаётся ориентированный ациклический граф $G = (V, E)$, $|V| = n$, $|E| = m$. Пусть также для каждого задания p_i значение i , $1 \leq i \leq n$, которое задаёт индекс задания, будет меткой вершины, соответствующей данному заданию. Произведём некоторую топологическую сортировку графа G и получим граф $G_s = (V_s, E_s)$, $|V_s| = n$, $|E_s| = m$. Очевидно, что для любой вершины p_j^s графа G_s если $(p_i^s, p_j^s) \in E_s$, то $i < j$, т. е. для любой вершины p_i^s её метка i меньше метки j любого её потомка p_j^s . Таким образом, если рассматривать вершины в порядке возрастания значений меток, то очередь рассмотрения вершины p_k^s наступает лишь после рассмотрения всех её предков.

Будем рассматривать вершины в порядке увеличения значений меток, назначенных вершинам графа при топологической сортировке. Пусть на шаге k , $1 \leq k \leq n$, длина уже построенного расписания НР^k равна $k-1$ и позиция последнего предка задания p_k^s равна l_k , т. е. $l_k = \max_{p_i^s: (p_i^s, p_k^s) \in E_s} \pi_i$, где π_i — позиция вершины p_i^s в расписании НР^k . Ввиду строения графа G_s все предки вершины p_k^s уже содержатся в частичном расписании НР^k . Рассмотрим расписания $\{\text{НР}_d^k\}_{d=l_k}^k$, каждое из которых получается путём вставки вершины p_k^s в НР^k на позицию d (вершины, для которых

$\pi_i \geq d$, сдвинутся на 1 позицию вправо). Введём вещественные значения $\eta_{kd} = \frac{1}{f_{\text{НР}_d^k}}$ — локальную информацию о выгодности вставки вершины p_k^s на позицию d в расписание НР^k . Здесь $f_{\text{НР}_d^k}$ — значение целевой функции для частично построенного расписания, вычисленное по формуле (1). Из множества $\{\text{НР}_d^k\}_{d=l_k}^k$ выбирается такое расписание, для которого значение η_{kd} максимально, т. е. $\text{НР}^{k+1} = \text{НР}_b^k$, $b = \operatorname{argmax}_{l_k \leq d \leq k} \eta_{kd}$.

Другими словами, на каждом шаге новая вершина вставляется в уже построенное расписание таким образом, чтобы локально минимизировать целевую функцию, а выбор вершин для вставки выполняется в порядке, определённом топологической сортировкой.

2.3. Муравьиный алгоритм. В МА используются следующие параметры:

- 1) критерий останова;
- 2) N — численность муравьёв, каждый из которых строит своё решение на каждой итерации;
- 3) ρ — коэффициент распада феромона, $\rho \in [0, 1]$;
- 4) q_0 — пороговое значение для вероятности перехода, $q_0 \in [0, 1]$;
- 5) α, β — влияние на вероятность перехода следа и жадной эвристики соответственно;
- 6) K — число выбираемых наилучших решений.

Пусть также $T = (\tau_{ij}) \in \mathbb{R}^{n \times n}$ — матрица феромонного следа и B — множество наилучших найденных решений, $|B| = K$; матрица T и множество B корректируются на каждой итерации алгоритма; η_{kd} — значение жадной эвристики, описанной в п. 2.2.

На вход алгоритму подаётся ориентированный ациклический граф $G = (V, E)$, $|V| = n$, $|E| = m$. Положим, что к нему уже применена топологическая сортировка, в противном случае применим её. Тогда МА может быть представлен в следующем виде.

Шаг 1. Сгенерировать начальное решение π_0 , используя только жадную эвристику. Пусть f^0 — значение целевой функции для такого расписания.

Шаг 2. Положить $\tau_0 := \frac{1}{f^0}$, $T := \tau_0$, $B := \{\pi_0\}$, $f_{\text{best}} := f^0$.

Шаг 3. Сгенерировать $2K$ случайных допустимых расписаний (перестановок), оставить среди них $K - 1$ лучших (с меньшим значением целевой функции) и пополнить ими множество B .

Шаг 4. Обновить $T := (1 - \rho)T$.

Шаг 5. Выполнить процедуру глобального обновления матрицы феромонного следа T .

ШАГ 6. Для каждого муравья $a = 1, \dots, N$ построить допустимое расписание в соответствии с операцией генерации решения и добавить его в множество B , при этом сохраняя его размер $|B| = K$ путём удаления наихудшего решения (решения с наибольшим значением целевой функции).

ШАГ 7. Положить $f_{\text{best}} = \min(f_{\text{best}}, \min_{\pi \in B} f_{\pi})$.

ШАГ 8. Если критерий останова не выполнен, то перейти на шаг 4.

ШАГ 9. Вернуть f_{best} .

Процедура генерации решения муравьём заключается в следующем. Как и при использовании жадной эвристики, расписание строится последовательной вставкой заданий на некоторые позиции, только выбор позиции происходит иным образом. Пусть на шаге k имеется множество $\{\text{HP}_d^k\}_{d=l_k}^k$ и вычислены значения $\{\eta_{kd}\}_{d=l_k}^k$. Для расчёта исторической выгоды c_d , $l_k \leq d \leq k$, т. е. того, насколько хорошим кажется расписание HP_d^k с точки зрения матрицы T , используется произведение

$$c_d = \prod_{p_i \in \text{HP}_d^k} \begin{cases} \tau_{ik}, & \text{если } \pi_i < d, \\ \tau_{ki}, & \text{если } \pi_i > d. \end{cases} \quad (2)$$

Фактически c_d — информация о том, насколько выгодным кажется определённый порядок вершины p_k относительно уже входящих в расписание HP_d^k вершин, исходя из истории, хранящейся в матрице T .

Выбор позиции d для вставки вершины p_k основывается на вероятностной схеме. Рассмотрим распределение вероятностей

$$P_{kd} = \frac{[c_d]^\alpha [\eta_{kd}]^\beta}{\sum_{h \in [l_k, k]} [c_h]^\alpha [\eta_{kh}]^\beta}, \quad d \in [l_k, k]. \quad (3)$$

Пусть $q \in [0, 1]$ — число, случайно выбранное в соответствии с равномерным распределением на указанном отрезке. Если $q \leq q_0$, то положим $d = \arg\max_{h \in [l_k, k]} [c_h]^\alpha [\eta_{kh}]^\beta$. Иначе d выбирается в соответствии с распределением вероятностей (3).

После выбора d и обновления π ($\text{HP}^{k+1} := \text{HP}_d^k$) происходит локальное обновление следа: все значения τ_{ij} , которые появились в произведении (2), изменяются по формуле

$$\tau_{ij} = (1 - \rho)\tau_{ij} - \rho\tau_0.$$

Процедура глобального обновления матрицы T выполняет для каждого $\pi \in B$ следующие действия.


```

1: for  $i = 1, 2, \dots, n - 1$  do
2:   for  $j = i + 1, i + 2, \dots, n$  do
3:     if  $\pi_i < \pi_j$  then  $\tau_{ij} := \tau_{ij} + \frac{1}{f_\pi}$ 
4:     else  $\tau_{ji} := \tau_{ji} + \frac{1}{f_\pi}$ 

```

В качестве критерия останова можно рассматривать заданное общее число итераций алгоритма или же заданное число итераций без улучшения целевой функции на наилучшем найденном решении.

3. Экспериментальное исследование свойств алгоритма

3.1. Описание инструментального компьютера и наборов данных. В рамках настоящей работы эксперименты проводились на компьютере со следующими характеристиками:

- ЦП: Intel(R) Xeon(R) CPU E5-2650 v4 @ 2,20 ГГц;
- объём дискового хранилища 450 ГБ;
- объём ОЗУ 62 ГБ;
- ОС Ubuntu 16.04.5 LTS.

Алгоритм на языке C++ реализован последовательно и использует ресурсы одного процессорного ядра.

Выполнены две серии экспериментов на синтетических наборах входных данных. В первой серии экспериментов набор данных состоит из графов, для которых по построению известно оптимальное или субоптимальное (несущественно большее чем оптимальное) значение целевой функции. Генерация таких графов основана на результатах работы [16], где предложен полиномиальный алгоритм построения расписания с минимизацией пикового использования ресурса для случая, когда граф заданий является последовательно-параллельным графом (ППГ).

С учётом того, что в [16] рассматривалась задача, в которой количество используемого ресурса сопоставлено не вершине графа, а ребру, схема генерации графа для экспериментов выглядит следующим образом.

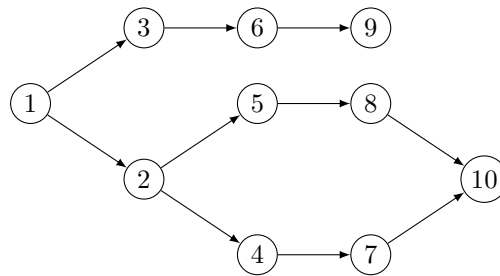


Рис. 4. Последовательно-параллельный граф

Вначале генерируется ППГ (рис. 4) для исходной задачи, т. е. с ресурсом, сопоставленным вершинам. Параметрами генерации являются число вершин и плотность графа (отношение числа рёбер к числу вершин). Затем осуществляется переход от этого ППГ к графу с ресурсом, сопоставленным рёбрам. При этом для каждой вершины исходного графа выполняется следующее. Если у вершины один потомок, то ресурс переносится из вершины на выходящее из неё ребро. Если у вершины более одного потомка, то используется преобразование, показанное (для случая двух потомков) на рис. 5 и в общем случае выводящее получаемый граф из класса ППГ.

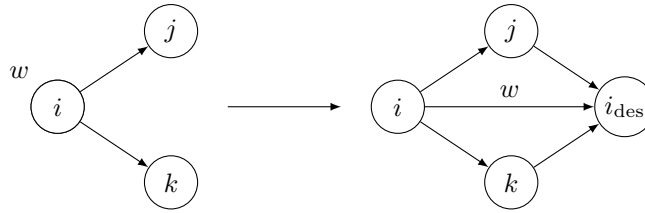


Рис. 5. Переход от ППГ с ресурсом в вершинах к ППГ с ресурсом на рёбрах

Из полученного графа выделяется ППГ (также с ресурсом на рёбрах), и для него строится оптимальное расписание при помощи алгоритма из [16]. Затем в граф возвращаются ранее отброшенные рёбра, при этом расписание корректируется для учёта соответствующих им зависимостей (на этом шаге расписание может стать субоптимальным). Затем происходит возврат к исходной постановке задачи с «возвращением» ресурса в вершины. Построенное расписание корректно и для исходной постановки, поскольку учитывает все зависимости.

Чтобы вывести формируемые графы заданий из класса последовательно-параллельных, для которых задача решается полиномиальным алгоритмом, в исходный ППГ добавляются дополнительные рёбра, которые не нарушают корректности построенного (суб-) оптимального расписания и не меняют значения целевой функции на этом расписании: если в расписании имеются задания a, b, c (именно в таком порядке, необязательно подряд), причём c — последний в расписании непосредственный потомок a , а рёбра $a \rightarrow b$ в графе нет, то это ребро можно добавить в граф, не изменив значения целевой функции на расписании. Неизменность целевой функции обусловлена тем, что ресурс, занятый заданием a , по-прежнему высвобождается после выполнения c . После добавления рёбер проверяется, что полученный граф не является ППГ.

В наборе данных для первой серии экспериментов число вершин варьируется от 60 до 250, средняя плотность графов равна 1,7, а веса вершин выбраны на отрезке $[1, 100]$ согласно равномерному распределению.

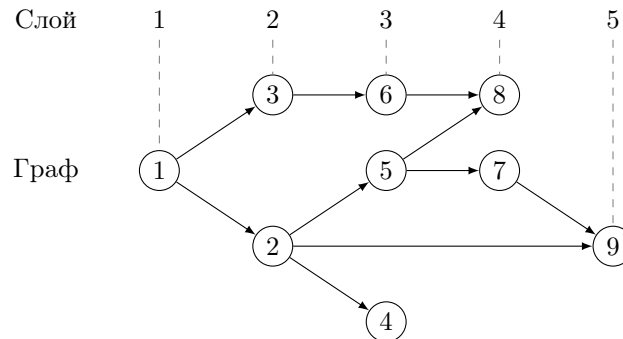


Рис. 6. Слоистый граф

Набор данных для второй серии экспериментов состоит из «слоистых» графов (рис. 6), структура которых соответствует приложениям по многостадийной обработке данных. В таких графах большинство рёбер связывают между собой вершины соседних слоёв. Для каждого из графов задаются число вершин и их веса, средняя ширина слоя, плотность связей между слоями, а также максимальное число слоёв, минуя которые можно провести ребро (на рис. 6 это ребро $2 \rightarrow 9$). Среднее отношение числа рёбер к числу вершин при генерации задано равным 3. Веса вершин выбраны на отрезке $[10, 50]$ согласно равномерному распределению. Схема генерации слоистых графов приведена в статье [17], там же дана ссылка на репозиторий с исходным кодом генератора.

Характеристики второго набора данных приведены в табл. 1. Из таблицы видно, что графы масштабированы в глубину, т. е. число вершин растёт за счёт увеличения числа слоёв (глубины графа). Для каждого числа вершин сгенерировано по 20 графов с приблизительно равными характеристиками. Результаты в таблице усреднены по графам с одинаковым числом вершин.

3.2. Настройки муравьиного алгоритма. Во всех экспериментах муравьиному алгоритму выставались заранее подобранные значения параметров: $N = 10$, $\rho = 0,2$, $q_0 = 0,8$, $\alpha = 0,6$, $\beta = 0,4$, $K = 6$. Рассматривались следующие критерии останова:

- 1) выполнение заданного числа итераций (500) без улучшения целевой функции на наилучшем найденном решении;
- 2) выполнение фиксированного числа итераций (1000).

В зависимости от выбранного критерия останова обозначим варианты алгоритма соответственно МАН и МА1000.

Фиксированное число итераций выбрано так, чтобы качество решений, находимых МАН и алгоритмом с фиксированным числом итераций,

Таблица 1

Параметры слоистых графов

Число вершин	Число рёбер	Глубина	Минимальное число вершин в слое	Среднее число вершин в слое	Максимальное число вершин в слое
50	150	5	3	10	15
55	161	6	3	10	15
60	181	6	5	11	16
65	196	9	3	7	11
70	211	10	3	7	11
75	226	10	3	8	11
80	240	11	2	7	11
85	257	10	3	9	13
100	290	15	3	7	10
120	358	15	3	8	12
130	389	19	3	7	11
140	421	20	3	7	11
150	451	21	3	7	11
160	478	23	2	7	11
170	510	24	2	7	11
180	542	26	3	7	11
190	569	24	3	8	12
200	600	25	3	8	12

было сопоставимым. На рис. 7 это видно по диаграмме, практически симметричной относительно отметки 0. Одной из целей экспериментов было выяснить, позволяет ли алгоритм МАН на рассматриваемом наборе слоистых графов заданий получить выигрыш по времени работы относительно МА1000 за счёт автоматического определения необходимости останова, когда оптимальное решение более не удаётся улучшить (при этом МА1000 может продолжать работу).

3.3. Результаты экспериментов для графов заданий с известным субоптимальным значением целевой функции. В этой серии экспериментов выполнялся прогон МАН и алгоритма, использующего только жадную эвристику (см. п. 2.2). Каждый из этих алгоритмов строил решение со значением целевой функции, не худшим (а в ряде случаев лучшим), чем заранее известное субоптимальное значение. При этом различие в значениях целевой функции на решении между муравьиным и жадным алгоритмами не превышало 1% в пользу муравьиного алгоритма.

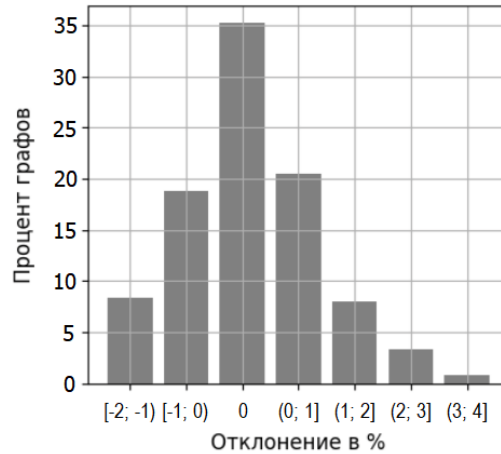


Рис. 7. Относительное отклонение целевой функции на решениях, получаемых МАН и МА1000

3.4. Результаты экспериментов для слоистых графов заданий. На рис. 7 приведена сводная информация о попарном сравнении значений целевой функции на решениях, получаемых МА с различными критериями останова, запускаемых на одних и тех же наборах данных. Например, по диаграмме видно, что целевая функция на решениях от МАН превышает целевую функцию на решениях от МА1000 на 1–2% (интервал по горизонтальной оси) примерно в 8% случаев (высота столбца диаграммы). В такой же доле случаев на решениях от МАН целевая функция меньше на 1–2%, чем целевая функция на решениях от МА1000. Основная часть (более 95%) отображённых на диаграмме случаев приходится на её практически симметричную часть в интервале от -2 до 2% по горизонтальной оси, что говорит о том, что по качеству решений на рассматриваемом наборе входных данных МАН и МА1000 выступают на равных.

На рис. 8 показано время работы МАН и МА1000. На горизонтальной оси расположены номера графов, отсортированных в лексикографическом порядке по числу вершин, затем по числу рёбер. Большой разброс по вертикальной оси у МАН вызван непостоянным числом итераций при запуске, так как алгоритм работает до тех пор, пока решения не перестанут улучшаться. Видно, что время работы МАН не превосходит, а зачастую много меньше времени работы МА с жёстко установленным числом итераций 1000.

Поскольку муравьиный алгоритм рандомизированный, необходимо исследовать стабильность его результата, т. е. разброс значений целевой функции при прогонах на одних и тех же входных данных. На рис. 9

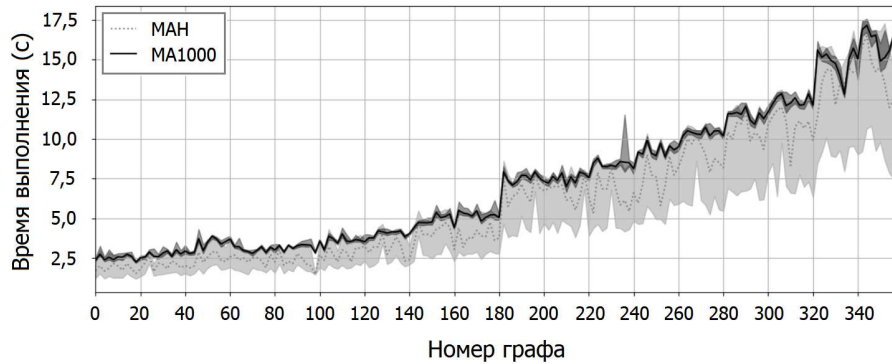


Рис. 8. Время выполнения МА с различными критериями останова

показан такой разброс для МАН и МА1000. Можно видеть, что разброс значений обоих алгоритмов не сильно различается и в среднем составляет около 5,5%.

Из графиков на рис. 7–9 следует, что МАН оказывается более гибким. Этот алгоритм способен автоматически определить, когда необходимо остановить процесс построения решения, за счёт чего выигрывает по времени работы у МА с жёстко заданным большим числом итераций (в данном случае — 1000), выдавая при этом решение сопоставимого качества.

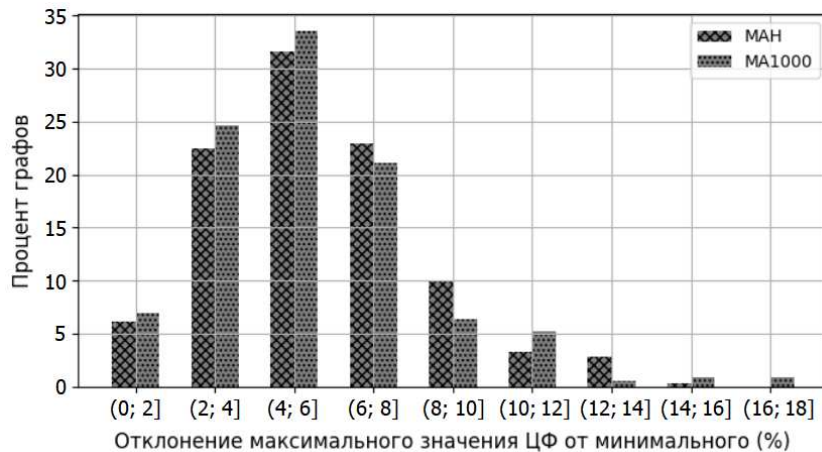


Рис. 9. Относительное отклонение максимального значения ЦФ от минимального, полученных на одном и том же графе заданий при различных запусках

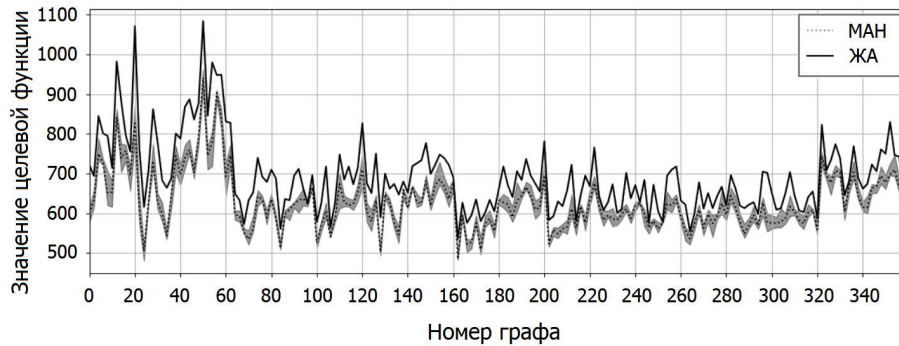


Рис. 10. Графики целевой функции на решениях, получаемых жадным алгоритмом и МАН

Рассмотрим теперь МАН на слоистых графах в сравнении с жадным алгоритмом (ЖА), использующим эвристику из п. 2.2. На рис. 10–11 показаны графики значения целевой функции на решении и времени работы алгоритмов. На всех графах жадный алгоритм отработал меньше секунды. Можно заметить, что с увеличением размера графа растёт не только время работы алгоритма МАН, но и разброс времени его работы, что связано с увеличением числа возможных перестановок и повышением вероятности продления работы МАН в связи с очередным небольшим улучшением решения.

На рис. 12 представлено относительное отклонение целевой функции на решениях, получаемых жадным алгоритмом, от целевой функции на решениях, найденных МАН. Видно, что в более чем 45% случаев МАН удалось найти решение, на 10–30% лучшее по значению целевой функции, чем решение от жадного алгоритма, что говорит о способности МА, использующего ту же жадную эвристику, существенно улучшать решение за счёт использования механизма феромонного следа.

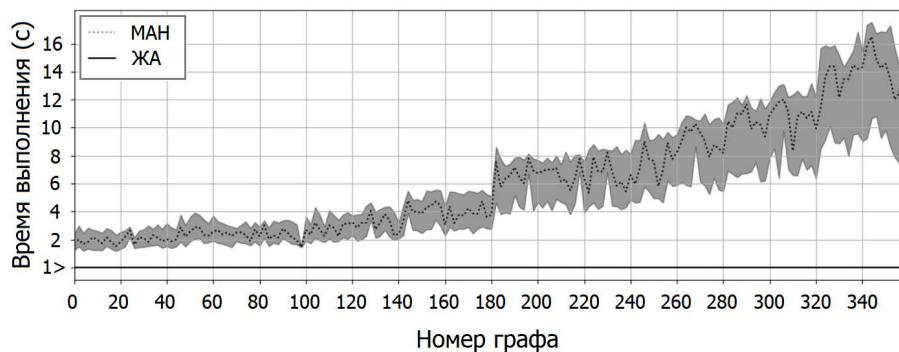


Рис. 11. Время работы жадного алгоритма и МАН

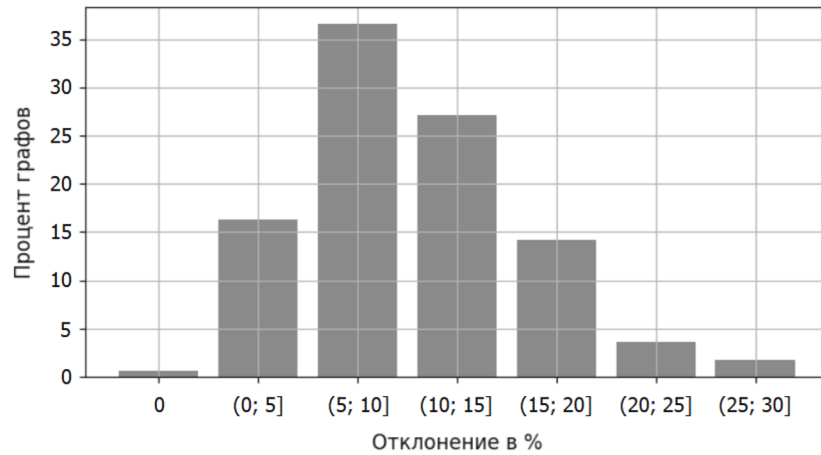


Рис. 12. Относительное отклонение целевой функции на решениях, получаемых жадным алгоритмом и МАН

Заключение

Предложен муравьиный алгоритм для решения задачи о построении расписания с минимизацией пикового использования ресурса, описаны отличия поставленной задачи от стандартных задач, решаемых при помощи МА. По итогам исследования алгоритма на рассмотренной выборке слоистых графов заданий, сделаны следующие выводы.

1. МА способен значительно улучшать качество решений, найденных жадной эвристикой, что вкупе с приемлемым временем работы делает использование МА оправданным.

2. МА, останавливающийся после выполнения заданного числа итераций (500) без улучшения целевой функции на наилучшем найденном решении, позволяет на графах рассмотренной размерности получить заметный выигрыш по времени выполнения по сравнению с МА, останавливающимся после выполнения фиксированного числа итераций (1000). При этом указанные МА выдают решения сопоставимого качества.

3. Применяемая в предложенном алгоритме схема работы с феромонным следом, учитывающая не только непосредственное следование заданий в расписании, вносит существенный вклад в сходимость МА.

В качестве направлений дальнейшего развития МА рассматриваются выполнение параллельной реализации (в рамках одной итерации могут одновременно строиться расписания для разных муравьёв), а также увеличение доли случайности в работе алгоритма при стабилизации значения целевой функции на лучшем решении с целью выхода из локального оптимума.

Финансирование работы

Исследование выполнено за счёт бюджетов организаций, обозначенных 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. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
2. Abdel-Wahab H. M. Scheduling with applications to register allocation and deadlock problems: PhD Thes. Waterloo, ON: Univ. Waterloo, 1976. 440 p.
3. Sethi R. Complete register allocation problems // SIAM J. Comput. 1975. V. 4, No. 3. P. 226–248.
4. Bauer A., Straus C., Hartle R. F. Minimizing total tardiness on a single machine using ant colony optimization // Cent. Eur. J. Oper. Res. Econ. 2000. V. 8, No. 2. P. 125–141.
5. Гафаров Е. Р. Гибридный алгоритм решения задачи минимизации суммарного запаздывания для одного прибора // Информ. технологии. 2007. № 1. С. 30–37.
6. Jha S. Balanced ant colony algorithm for scheduling DAG to grid heterogeneous system // Int. J. Sci. Eng. Res. 2011. V. 2, No. 6. 10 p.
7. Myszkowski P., Skowronski M., Olech L., Oslizlo K. Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem // Soft Comput. 2015. No. 19. P. 3599–3619.
8. Костенко В. А., Плакунов А. В. Муравьиные алгоритмы для планирования вычислений в центрах обработки данных // Вестн. Моск. ун-та. Сер. 15. Вычисл. математика и кибернетика. 2017. № 1. С. 44–50.
9. Gagne C., Price W. L., Grave M. Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times // J. Oper. Res. Soc. 2002. V. 53. P. 895–906.
10. Gurobi optimizer. Gurobi optimization, 2023. URL: www.gurobi.com/solutions/gurobi-optimizer (accessed Mar. 22, 2024).
11. MOSEK solver. MOSEK ApS, 2023. URL: www.mosek.com/products/mosek (accessed Mar. 22, 2024).
12. SCIP: Solving Constraint Integer Programs. Berlin: Zuse Inst. Berlin, 2023. URL: www.scipopt.org (accessed Mar. 22, 2024).
13. COIN-OR: Computations Infrastructure for Operations Research. Projects by Category. Towson: COIN-OR Found., 2023. URL: www.coin-or.org/projects (accessed Mar. 22, 2024).
14. Xuan H. H., Linh-Trung N., Dong D. D., Huynh T. Solving the Traveling Salesman Problem with ant colony optimization: A revisit and new efficient algorithms // J. Electron. Commun. 2012. V. 2. P. 121–129.

15. **Tong C. J., Lau H. C., Lim A.** Ant colony optimization for the Ship Berthing Problem // Advances in computing science—ASIAN'99. Proc. 5th Asian Computing Science Conf. (Phuket, Thailand, Dec. 10–12, 1999). Heidelberg: Springer, 1999. P. 359–370. (Lect. Notes Comput. Sci.; V. 1742). DOI: 10.1007/3-540-46674-6_30.
16. **Kayaaslan E., Lambert T., Marchal L., Uçar B.** Scheduling series-parallel task graphs to minimize peak memory // Theor. Comput. Sci. 2018. V. 707. P. 1–23.
17. **Canon L.-C., El Sayah M., Héam P.-C.** A comparison of random task graph generation methods for scheduling problems // Euro-Par 2019: Parallel processing. Proc. 25th Int. Conf. Parallel and Distributed Computing (Göttingen, Germany, Aug. 26–30, 2019). Cham: Springer, 2019. P. 61–73. (Lect. Notes Comput. Sci.; V. 11725). DOI: 10.1007/978-3-030-29400-7_5.

Балашов Василий Викторович
Абрамов Алексей Владимирович
Чупахин Андрей Андреевич
Туркин Андрей Владимирович
Гао Цзесин
Сунь Шумин
Чжоу Ли
Сунь Цзе

Статья поступила
19 декабря 2022 г.
После доработки —
25 октября 2023 г.
Принята к публикации
22 декабря 2023 г.

ANT COLONY ALGORITHM
FOR SINGLE PROCESSOR SCHEDULING
WITH MINIMIZATION OF PEAK RESOURCE USAGE

V. V. Balashov^{1, a}, A. V. Abramov^{1, b}, A. A. Chupakhin^{1, c},
A. V. Turkin^{2, d}, J. Gao^{2, e}, C. Sun^{3, f}, L. Zhou^{3, g}, and J. Sun^{3, h}

¹ Lomonosov Moscow State University,
1 Leninskie Gory, 119991 Moscow, Russia

² Huawei Moscow Research Center,
7-9 Smolenskaya Square, 121099 Moscow, Russia

³ Huawei Hong Kong Research Center,
8/F, 2 Science Park West Avenue, 999077 Hong Kong, PRC

E-mail: ^ahbd@cs.msu.ru, ^bavabramov@lvk.cs.msu.ru,
^candrewchup@lvk.cs.msu.ru, ^dandrei.turkin@huawei.com,
^egaojiexing@huawei.com, ^fsunchumin@huawei.com,
^gzhouli107@huawei.com, ^hj.sun@huawei.com

Abstract. We consider the problem of constructing a single processor task schedule with minimization of peak resource usage. An example of the resource is the main memory of the target computer. Task set to be scheduled is represented as a directed acyclic graph every node of which is marked with the amount of resource used by the corresponding task. The resource allocated to a task is released on completion of the last (according to the schedule) immediate successor of this task in the graph. Correctness constraint on the schedule is the partial order specified by the task graph. Task duration values are not considered. The formal statement of the problem is provided. To solve the problem, we propose an ant colony algorithm modified so that the pheromone matrix reflects the desirability of pairwise order in the schedule for every pair of tasks, not only for pairs of adjacent tasks. During the schedule construction, for every task the algorithm chooses its position in the schedule, in contrast to existing ant colony scheduling algorithms that construct schedule in increasing order of positions (left-to-right) choosing a task for every next position. Experimental evaluation of

the algorithm was conducted on two sets of task graphs. The first set contains graphs generated in such a way that the estimation for the optimum value of the goal function is known a priori. Graphs from the second set are “layered,” and their structure corresponds to the structure of multistage data processing applications. In both sets, the graphs are generated randomly with respect to specified generation parameters and constraints on the graph structure. The experiments indicate high precision and stability of the proposed ant colony algorithm. Tab. 1, illustr. 12, bibliogr. 17.

Keywords: combinatorial optimization, single-processor schedule, resource minimization, ant colony algorithm.

References

1. **M. R. Garey** and **D. S. Johnson**, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979; Mir, Moscow, 1982 [Russian]).
2. **H. M. Abdel-Wahab**, Scheduling with applications to register allocation and deadlock problems, *PhD Thesis* (Univ. Waterloo, Waterloo, ON, 1976).
3. **R. Sethi**, Complete register allocation problems, *SIAM J. Comput.* **4** (3), 226–248 (1975).
4. **A. Bauer**, **C. Straus**, and **R. F. Hartle**, Minimizing total tardiness on a single machine using ant colony optimization, *Cent. Eur. J. Oper. Res. Econ.* **8** (2), 125–141 (2000).
5. **E. R. Gafarov**, A hybrid algorithm for solution to the total delay minimization problem with one device, *Inf. Tekhnol.*, No. 1, 30–37 (2007) [Russian].
6. **S. Jha**, Balanced ant colony algorithm for scheduling DAG to grid heterogeneous system, *Int. J. Sci. Eng. Res.* **2** (6) (2011).
7. **P. Myszkowski**, **M. Skowronski**, **L. Olech**, and **K. Oslizlo**, Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem, *Soft Comput.*, No. 19, 3599–3619 (2015).
8. **V. A. Kostenko** and **A. V. Plakunov**, Ant algorithms for scheduling computations in data centers, *Vestn. Mosk. Univ., Ser. 15*, No. 1, 44–50 (2017) [Russian] [*Mosc. Univ. Comput. Math. Cybern.* **41** (1), 44–50 (2017)].
9. **C. Gagne**, **W. L. Price**, and **M. Grave**, Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times, *J. Oper. Res. Soc.* **53**, 895–906 (2002).
10. Gurobi Optimizer (Gurobi Optimization, 2023), URL: www.gurobi.com/solutions/gurobi-optimizer (accessed Mar. 22, 2024).
11. MOSEK Solver (MOSEK ApS, 2023), URL: www.mosek.com/products/mosek (accessed Mar. 22, 2024).
12. SCIP: Solving Constraint Integer Programs (Zuse Inst. Berlin, Berlin, 2023), URL: www.scipopt.org (accessed Mar. 22, 2024).
13. COIN-OR: Computations Infrastructure for Operations Research. Projects by Category (COIN-OR Found., Towson, 2023), URL: www.coin-or.org/projects (accessed Mar. 22, 2024).

14. **H. H. Xuan, N. Linh-Trung, D. D. Dong, and T. Huynh**, Solving the Traveling Salesman Problem with ant colony optimization: A revisit and new efficient algorithms, *J. Electron. Commun.* **2**, 121–129 (2012).
15. **C. J. Tong, H. C. Lau, and A. Lim**, Ant colony optimization for the Ship Berthing Problem, in *Advances in Computing Science — ASIAN’99* (Proc. 5th Asian Computing Science Conf., Phuket, Thailand, Dec. 10–12, 1999) (Springer, Heidelberg, 1999), pp. 359–370 (Lect. Notes Comput. Sci., Vol. 1742), DOI: 10.1007/3-540-46674-6_30.
16. **E. Kayaaslan, T. Lambert, L. Marchal, and B. Uçar**, Scheduling series-parallel task graphs to minimize peak memory, *Theor. Comput. Sci.* **707**, 1–23 (2018).
17. **L.-C. Canon, Sayah M. El, and P.-C. Héam**, A comparison of random task graph generation methods for scheduling problems, in *Euro-Par 2019: Parallel Processing* (Proc. 25th Int. Conf. Parallel and Distributed Computing, Göttingen, Germany, Aug. 26–30, 2019) (Springer, Cham, 2019), pp. 61–73 (Lect. Notes Comput. Sci., Vol. 11725), DOI: 10.1007/978-3-030-29400-7_5.

Vasily V. Balashov
Aleksey V. Abramov
Andrey A. Chupakhin
Andrey V. Turkin
Jiexing Gao
Chumin Sun
Li Zhou
Jie Sun

Received December 19, 2022
Revised October 25, 2023
Accepted December 22, 2023

УСТОЙЧИВОСТЬ ВЕРШИННЫХ ПОКРЫТИЙ В ИГРЕ С КОНЕЧНЫМ ЧИСЛОМ ШАГОВ

В. Л. Береснев^{1, a}, А. А. Мельников^{1, b}, С. Ю. Утюпин^{2, c}

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, ^cstepan.utyupin@gmail.com

Аннотация. Задача о вечном вершинном покрытии является вариантом задачи о вершинном покрытии графа и может рассматриваться как динамическая игра двух сторон (Атакующего и Защитника) с бесконечным числом шагов. На каждом шаге имеется размещение охранников по вершинам графа, образующее вершинное покрытие. Атакующий выбирает для атаки одно из рёбер графа, а Защитник должен переместить охранника вдоль атакованного ребра из одной вершины в другую. Кроме того, Защитник может переместить любое подмножество остальных охранников из вершин, в которых они находились до атаки, в одну из незанятых соседних вершин, чтобы получить новое вершинное покрытие.

В статье описана процедура, позволяющая ответить на вопрос, существует ли выигрышная стратегия для Защитника, позволяющая защищать вершинное покрытие в течении заданного числа шагов. Для построения стратегии Защитника задача представляется в виде динамической игры Штакельберга, на каждом шаге которой взаимодействие противоборствующих сторон формализуется как двухуровневая задача математического программирования. Ил. 6, библиогр. 11.

Ключевые слова: динамическая игра Штакельберга, вечное вершинное покрытие, защита рёбер графа, алгоритм проверки устойчивости.

Введение

Модели вечной защиты рёбер графа появились в 1990-х гг. в качестве формализации для стратегии императора Константина по защите

Римской империи [1]. В этих моделях рёбра графа защищаются от бесконечной последовательности атак. Защита осуществляется с помощью охранников, размещаемых в вершинах графа и способных перемещаться по его рёбрам после атаки.

В данной статье рассматривается задача о вечном вершинном покрытии, сформулированная в [2]. Эту задачу можно интерпретировать как бесконечную игру двух сторон — Атакующего и Защитника. Первоначально Защитник размещает в вершинах заданного графа охранников так, что вершины, в которых размещены охранники, образуют вершинное покрытие. Далее на каждом шаге Атакующий выбирает для атаки любое ребро, в одной из вершин которого нет охранника. В ответ Защитник перемещает охранника из вершины атакованного ребра в другую вершину этого ребра. Кроме того, Защитник имеет право из любой вершины переместить охранника вдоль любого ребра в соседнюю вершину, если в ней не было охранника, либо охранник был перемещён из этой вершины. В одну вершину не могут быть перемещены несколько охранников, при этом один и тот же охранник не может быть перемещён более одного раза на одном шаге. Если реконструированное размещение охранников в вершинах образует вершинное покрытие, то начинается следующий шаг с той же последовательностью действий. Если же с помощью указанной процедуры не удаётся построить вершинное покрытие, т. е. хотя бы одно ребро окажется незащищённым, то Атакующий побеждает и игра завершается. Победителем становится Защитник, если он может защититься от бесконечной последовательности атак.

Задача нахождения минимального числа охранников (eternal vertex cover number, EVCN), образующих вершинное покрытие, для которого в рассматриваемой игре может быть построена выигрышная стратегия Защитника, есть задача о вечном вершинном покрытии. В [3] установлено, что эта задача NP-трудна, а в [4] показано, что задача остаётся NP-трудной для двудольных графов. В статье [5] авторы предлагают новую нижнюю оценку для EVCN произвольного графа через мощность вершинного покрытия минимального размера, содержащего все точки сочленения данного графа. Также предлагается квадратичный по времени алгоритм для вычисления EVCN для хордальных графов. В [6] для некоторых классов графов построены полиномиальные алгоритмы вычисления EVCN. В статье [7] предложен полиномиальный алгоритм для специализированных древовидных графов. В [8] получен полиномиальный алгоритм для поиска EVCN в цепных графах и кографах, а также предложен алгоритм с линейным временем для поиска EVCN для разделённых графов, подкласса хордальных графов. В [3] авторы показывают, что для задачи о вечном вершинном покрытии существует 2-приближённый полиномиальный по времени алгоритм.

Можно заметить, что перечисленные и многие родственные работы либо предлагают алгоритмы для конкретных семейств графов, либо приводят оценки для общих или частных случаев. В данной статье мы рассматриваем ситуацию, когда задан граф, на структуру которого не наложено никаких ограничений, а также дано некоторое вершинное покрытие без каких-либо предположений о нём. Рассматривается вопрос, какой игрок выигрывает, если игра «Атакующий — Защитник» начнётся с заданного вершинного покрытия.

Ответ на этот вопрос требует анализа всех бесконечных последовательностей атак. На практике достаточно рассматривать последовательности некоторой достаточно большой длины из-за конечности множества возможных состояний игры, однако стоит ожидать, что эта длина будет неизвестной. В этой работе мы предлагаем процедуру, позволяющую ответить на вопрос, существует ли выигрышная стратегия для Защитника, позволяющая защищать заданное вершинное покрытие на протяжении наперёд заданного конечного числа шагов. Данная процедура является обобщением процедуры, предложенной в [9].

Для построения стратегии Защитника рассматриваемая игра представляется в виде динамической игры Штакельберга. На каждом шаге взаимодействие противоборствующих сторон формализуется в виде задачи двухуровневого математического программирования. Решение задачи верхнего уровня (задачи Атакующего) определяет ребро, атака которого приводит к максимальному числу незащищённых рёбер после реконструкции Защитником размещения охранников. Решение задачи нижнего уровня (задачи Защитника) при заданном решении Атакующего изменяет размещение охранников с целью минимизировать число незащищённых рёбер.

Приводятся результаты вычислительных экспериментов с использованием предложенной процедуры.

1. Формулировка задачи, рассматриваемой на шагах исследуемой игры

Пусть $G = (I, J)$ есть ориентированный граф без петель с множеством вершин I и множеством дуг J . Для всякого $j \in J$ обозначим через $i(j)$ и $k(j)$ начальную и конечную вершины дуги j соответственно и будем называть эти вершины соседними. Отметим, что рассмотрение ориентированного графа не уменьшает общности двухуровневой задачи, рассматриваемой на каждом шаге исследуемой игры. В самом деле, в неориентированном графе каждое ребро можно заменить двумя противоположно ориентированными параллельными дугами. При этом будем считать, что перемещение охранника по дуге происходит от начальной вершины к конечной вершине дуги.

Вершинным покрытием в графе G считаем всякое подмножество множества I , содержащее начальную или конечную вершину любой дуги из множества J .

Для формальной записи двухуровневой задачи введём следующие обозначения:

- $x_{i0} \in \{0, 1\}$ — заданные величины, определяющие начальное для рассматриваемого шага вершинное покрытие: $x_{i0} = 1$, если в вершине $i \in I$ размещён охранник, и $x_{i0} = 0$ в противном случае;
- $x_i \in \{0, 1\}$ — переменные, задающие размещение охранников по завершении шага: $x_i = 1$, если в вершине $i \in I$ размещён охранник, и $x_i = 0$ в противном случае;
- $y_j \in \{0, 1\}$ — переменные, обозначающие атакованную на шаге дугу: $y_j = 1$, если дуга $j \in J$ атакована, и $y_j = 0$ в противном случае;
- $u_j \in \{0, 1\}$ — переменные, обозначающие дуги, которые по завершении шага оказались незащищёнными: $u_j = 1$, если в начальной и конечной вершинах дуги $j \in J$ не размещён охранник, и $u_j = 0$ в противном случае;
- $v_j \in \{0, 1\}$ — переменные, обозначающие перемещение охранников по дугам на шаге: $v_j = 1$, если охранник перемещается по дуге $j \in J$ из вершины $i(j)$ в вершину $k(j)$, и $v_j = 0$ в противном случае.

С использованием введённых обозначений модель, формализующая взаимодействие сторон на шаге рассматриваемой игры, записывается как следующая задача двухуровневого математического программирования:

$$\max_{(y_j)} \sum_{j \in J} u_j^*, \quad (1)$$

$$\sum_{j \in J} y_j \leq 1, \quad (2)$$

$$x_{i(j)0} \geq y_j, \quad j \in J, \quad (3)$$

$$x_{i(j)0} + x_{k(j)0} + y_j \leq 2, \quad j \in J, \quad (4)$$

$$y_j \in \{0, 1\}, \quad j \in J, \quad (5)$$

$(x_j^*), (v_j^*), (u_j^*)$ — оптимальное решение задачи:

$$\min_{(x_i), (v_j), (u_j)} \sum_{j \in J} u_j, \quad (6)$$

$$v_j \geq y_j, \quad j \in J, \quad (7)$$

$$x_{i0} \geq \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (8)$$

$$1 - x_{i0} \geq \sum_{j: k(j)=i} v_j - \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (9)$$

$$x_i = x_{i0} + \sum_{j: k(j)=i} v_j - \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (10)$$

$$x_{i(j)} + x_{k(j)} + u_j \geq 1, \quad j \in J, \quad (11)$$

$$x_i, u_j, v_j \in \{0, 1\}, \quad i \in I, j \in J. \quad (12)$$

Данная модель, как и всякая двухуровневая задача, включает задачу верхнего уровня (1)–(5) (задачу Атакующего) и задачу нижнего уровня (6)–(12) (задачу Защитника).

Целевые функции (1) и (6) обеих задач показывают, сколько дуг осталось без защиты после реконструкции вершинного покрытия на рассматриваемом шаге. Неравенство (2) означает, что на рассматриваемом шаге атакованной может быть только одна дуга, а из неравенств (3)–(4) следует, что атакованной может быть только дуга, у которой охранник размещён в начальной вершине и нет охранника в конечной вершине. Ограничения (7)–(9) задачи нижнего уровня формализуют правила возможных перемещений охранников вдоль соответствующих дуг. Условие (7) означает, что если дуга атакована, то охранник перемещается в конечную вершину дуги. Из неравенства (8) следует, что если в рассматриваемом на этом шаге вершинном покрытии в вершине i размещён охранник, то его перемещение возможно не более чем по одной дуге, исходящей из этой вершины, а если в вершине i охранника нет, то перемещение охранников по дугам, исходящим из вершины i , невозможно. Аналогично из неравенства (9) с учётом условия (8) получаем, что если в рассматриваемом на этом шаге вершинном покрытии в вершине i нет охранника, то $\sum_{j: k(j)=i} v_j \leq 1$. Это означает, что если охранник не размещён в вершине i , то не более чем по одной дуге, входящей в вершину i , может быть перемещён охранник. Если же в вершине i размещён охранник, то из неравенств (8) и (9) получаем

$$1 \geq \sum_{j: i(j)=i} v_j \geq \sum_{j: k(j)=i} v_j.$$

Это означает, что по одной из дуг, входящих в вершину i , может быть перемещён охранник, но только в случае, когда охранник, размещённый в вершине i , также перемещается. Равенство (10) определяет новое размещение охранников в результате их перемещения на рассматриваемом шаге. Неравенство (11) заставляет переменную u_j принимать ненулевое значение, когда в начальной и конечной вершинах дуги j нет охранников.

Допустимым решением двухуровневой задачи на рассматриваемом шаге игры назовём пару $((y_j), ((x_i), (v_j), (u_j)))$, где (y_j) — допустимое решение задачи верхнего уровня, а $((x_i), (v_j), (u_j))$ — оптимальное решение задачи нижнего уровня. Допустимое решение будет оптимальным, если значение целевой функции на этом решении не меньше, чем на других

допустимых решениях. Вычисление оптимального решения позволяет проверить текущее вершинное покрытие на устойчивость к атаке. Если оптимальное значение целевой функции равно нулю, то ответ положительный. Если же это значение больше нуля, то существует допустимое решение задачи верхнего уровня (атакуемое ребро) для которого не существует реконструкции вершинного покрытия, приводящей к новому вершинному покрытию.

2. Устойчивость вершинных покрытий

Как следует из сказанного выше, возможность восстановления (реконструкции) вершинного покрытия после атаки определяется значениями целевых функций задач верхнего и нижнего уровней в двухуровневой модели (1)–(12), рассматриваемой на каждом шаге исследуемой игры.

Вершинное покрытие $C_0 = (x_{i0})$ будем называть *устойчивым к атаке дуги $j_0 \in J$* , если оптимальное значение целевой функции задачи Защитника равно нулю при допустимом решении (y_j) , $y_{j_0} = 1$, задачи Атакующего.

Вершинное покрытие $C_0 = (x_{i0})$ назовём *1-устойчивым*, если оптимальное значение целевой функции задачи Атакующего равно нулю, иначе покрытие C_0 будем называть *неустойчивым*.

Отсюда следует, что процедура проверки 1-устойчивости заданного вершинного покрытия $C_0 = (x_{i0})$ сводится к проверке устойчивости этого покрытия к атаке всякой допустимой для атаки дуги $j_0 \in J$. Эта проверка может быть ускорена, если воспользоваться следующей леммой.

Лемма 1. Пусть (y_j) — допустимое решение задачи Атакующего и существует решение $((x_i), (v_j), (u_j))$ задачи Защитника при решении (y_j) со значением целевой функции, равным нулю. Если $j' \in \{j \in J \mid v_j = 1\}$ и допустимое решение (y'_j) задачи Атакующего таково, что $y'_{j'} = 1$, то оптимальное значение целевой функции задачи Защитника равно нулю при решении задачи Атакующего (y'_j) .

ДОКАЗАТЕЛЬСТВО. Рассмотрим оптимальное решение $((x_i), (v_j), (u_j))$ задачи Защитника, соответствующее решению (y_j) задачи Атакующего, и заметим, что это решение есть допустимое решение задачи Защитника при решении (y'_j) . Действительно, ограничение (7) выполняется, поскольку $v_{j'} = 1$, а ограничения (8)–(12) справедливы, поскольку не зависят от решения (y'_j) . Таким образом, получаем, что $((x_i), (v_j), (u_j))$ — оптимальное решение задачи Защитника при решении (y'_j) со значением целевой функции, равным нулю.

Следствие 1. Если для заданного вершинного покрытия при атаке дуги $j \in J$ существует реконструкция, дающая 1-устойчивое покрытие,

определяемое решением $((x_i), (v_j), (u_j))$ задачи Защитника, то такая реконструкция возможна при атаке любой дуги $j' \in \{j \in J \mid v_j = 1\}$.

Резюмируя сказанное, получим следующую процедуру проверки 1-устойчивости заданного вершинного покрытия $C_0 = (x_{i0})$. Процедура состоит из начального шага и некоторого числа однотипных основных шагов.

На начальном шаге определяем множество

$$J_0 = \{j \in J \mid x_{i(j)0} = 1, x_{k(j)0} = 0\},$$

задающее допустимые решения задачи Атакующего.

На основном шаге рассматриваем множество J_0 . Если $J_0 = \emptyset$, то процедура проверки 1-устойчивости завершится с положительным результатом. Если $J_0 \neq \emptyset$, то выбираем $j_0 \in J_0$ и вычисляем оптимальное решение $((x_i), (v_j), (u_j))$ задачи Защитника при допустимом решении (y_j) , $y_{j_0} = 1$, задачи Атакующего. Если оптимальное значение целевой функции задачи не равно нулю, то процедура проверки завершается с отрицательным результатом. В противном случае полагаем $J_0 := J_0 \setminus \{j \in J \mid v_j = 1\}$ и переходим к следующему основному шагу.

3. Устойчивость вершинного покрытия в игре с конечным числом шагов

Рассмотрим игру, состоящую из $L > 1$ шагов, с начальным вершинным покрытием $C_0 = (x_{i0})$. Пусть на шагах $1, 2, \dots, L-1$ оптимальное значение целевой функции задачи Атакующего равно нулю, а на шаге L оно больше нуля. Это означает, что в результате $L-1$ шагов получено вершинное покрытие такое, что после атаки некоторой дуги успешная реконструкция этого вершинного покрытия невозможна. Заметим, что отсюда не следует, что при начальном покрытии C_0 у Защитника нет выигрышной стратегии. Действительно, полученное к шагу L вершинное покрытие есть результат последовательных изменений вершинных покрытий, построенных на предыдущих шагах. Каждое такое изменение определяется оптимальным решением задачи Защитника, которое может быть не единственным. Следовательно, при соответствующем выборе оптимальных решений на шагах, предшествующих шагу L , может быть получено вершинное покрытие, которое будет устойчивым к атаке любого ребра на шаге L .

Вершинное покрытие $C_0 = (x_{i0})$ назовём L -устойчивым, если оно 1-устойчиво и для любого допустимого решения задачи Атакующего существует оптимальное решение задачи Защитника, дающее $(L-1)$ -устойчивое вершинное покрытие. Если покрытие L -устойчиво, то число L будем называть *степенью устойчивости*.

Следовательно, при проверке L -устойчивости заданного вершинного покрытия для каждого допустимого решения задачи Атакующего может потребоваться исследование, вообще говоря, всех оптимальных решений задачи Защитника. Таким образом, алгоритм проверки L -устойчивости исходного вершинного покрытия $C_0 = (x_{i0})$ включает два этапа. На первом этапе проверяем 1-устойчивость исходного вершинного покрытия, а на втором этапе для допустимых решений задачи Атакующего проверяем существование оптимального решения задачи Защитника, дающего $(L - 1)$ -устойчивое вершинное покрытие.

При этом в ходе работы алгоритма формируем и используем список D проверенных вершинных покрытий, содержащий информацию о степенях устойчивости этих покрытий. Для вершинного покрытия $C \in D$ определяем величины $v(C)$ и $u(C)$, при этом $v(C) = 0$, если C не 1-устойчиво, и $v(C) = l$, если C — l -устойчивое вершинное покрытие; $u(C) = l$, если C не является l -устойчивым покрытием.

В начале первого этапа алгоритма проверяем наличие вершинного покрытия C_0 в списке D . Если $C_0 \in D$ и $v(C_0) = 0$, то вершинное покрытие не 1-устойчиво, и алгоритм проверки L -устойчивости завершает работу. Если $C_0 \in D$ и $u(C_0) \leq L$, то вершинное покрытие не L -устойчиво и алгоритм также завершает работу. Если $C_0 \in D$ и $v(C_0) > 0$, то начинается второй этап алгоритма. Если $C_0 \notin D$, то выполняем процедуру проверки 1-устойчивости C_0 , и в зависимости от результата завершаем проверку L -устойчивости C_0 с отрицательным исходом или полагаем $v(C_0) = 1$, включаем C_0 в список D и начинаем второй этап алгоритма.

Второй этап алгоритма аналогичен процедуре проверки 1-устойчивости вершинного покрытия. На начальном шаге формируем множество $J_0 \subseteq J$, задающее допустимые решения задачи Атакующего. Далее идёт последовательность основных шагов, на каждом из которых для некоторого $j \in J_0$ проверяем существование $(L - 1)$ -устойчивого вершинного покрытия, полученного после атаки дуги j . Несложно увидеть, что число таких шагов можно уменьшить аналогично тому, как это сделано для случая проверки 1-устойчивости в разд. 2.

На основном шаге рассматриваем множество J_0 , задающее допустимые и не просмотренные решения задачи Атакующего. Если $J_0 = \emptyset$, то исходное вершинное покрытие C_0 L -устойчиво, и алгоритм завершается с положительным результатом. Если $J_0 \neq \emptyset$, то выбираем $j_0 \in J_0$ и выполняем процедуру поиска оптимального решения задачи Защитника при атаке дуги j_0 , дающего $(L - 1)$ -устойчивое вершинное покрытие. В ходе этой процедуры, включающей некоторое число однотипных итераций, строим множество *запрещённых* вершинных покрытий $C(j_0)$, включающее рассмотренные и отброшенные покрытия C' , не являющиеся $(L - 1)$ -устойчивыми. На первой итерации $C(j_0) = \emptyset$.

На каждой итерации рассматриваем задачу Защитника с дополнительными ограничениями, отсекающими решения, приводящие к вершинным покрытиям из множества $C(j_0)$. Если $C' = (x'_i)$ и $C' \in C(j_0)$, то соответствующее дополнительное ограничение имеет вид

$$\sum_{i \in I} x'_i x_i \leq \sum_{i \in I} x'_i - 1.$$

Пусть на очередной итерации оптимальное значение целевой функции задачи Защитника с дополнительными ограничениями, отсекающими запрещённые вершинные покрытия из множества $C(J_0)$, равно нулю. Если $((x_i), (v_j), (u_j))$ — оптимальное решение, то $C = (x_i)$ — новое вершинное покрытие. Если $C \in D$ и $v(C) = 0$ или $u(C) \leq L - 1$, то включаем C в множество $C(j_0)$ и начинаем следующую итерацию на текущем шаге. Если $C \in D$ и $v(C) \geq L - 1$, то полагаем $J_0 = J_0 \setminus \{j \mid v_j = 1\}$ и переходим к следующему шагу. Если $C \in D$ и $0 \leq v(C) < L - 1$, либо $C \notin D$, то обращаемся к алгоритму проверки $(L - 1)$ -устойчивости покрытия C . Если алгоритм проверки завершается с положительным результатом, то полагаем $J_0 = J_0 \setminus \{j \mid v_j = 1\}$ и включаем C в список D , полагая $v(C) = L - 1$. После этого переходим к следующему основному шагу. Если алгоритм проверки $(L - 1)$ -устойчивости покрытия C завершается с отрицательным результатом, то включаем покрытие в множество запрещённых покрытий $C(j_0)$ и начинаем следующую итерацию на текущем шаге.

Пусть на очередной итерации текущего шага значение целевой функции задачи Защитника с дополнительными ограничениями больше нуля. Это означает, что вершинных покрытий, кроме рассмотренных и включённых в множество запрещённых, не существует, поэтому исходное вершинное покрытие не L -устойчиво, и алгоритм проверки L -устойчивости завершается с отрицательным результатом.

Описанный алгоритм рекурсивный, и в ходе его работы при проверке l -устойчивости покрытия C проводится проверка $(l - 1)$ -устойчивости некоторого множества вершинных покрытий C' , порождённых покрытием C . Однако число вызовов функции проверки устойчивости и в целом трудоёмкость алгоритма значительно снижаются за счёт использования списка D проверенных вершинных покрытий. В результате каждое вершинное покрытие, рассмотренное в ходе работы алгоритма, проверяется на l -устойчивость, $l = 1, 2, \dots, L - 1$, не более одного раза.

4. Показательный пример

На рис. 1 приведён пример графа и его вершинного покрытия, которое 1-устойчиво, но не 2-устойчиво.

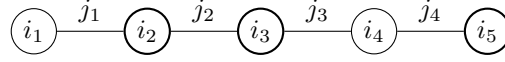


Рис. 1. Исходное покрытие

На рис. 1 множество выделенных вершин $\{i_2, i_3, i_5\}$, в которых размещены охранники, образует вершинное покрытие. На первом шаге у Атакующего есть возможность атаковать одно из рёбер j_1, j_3, j_4 . Легко видеть, что при любой атаке перемещение охранников по рёбрам j_1, j_3, j_4 соответственно приведёт к новому вершинному покрытию. Вместе с тем у Атакующего есть выигрышная стратегия в игре из двух шагов. Если на первом шаге атаковать ребро j_3 , то Защитник переместит охранника из вершины i_3 в вершину i_4 , что приведёт к размещению охранников, показанному на рис. 2.

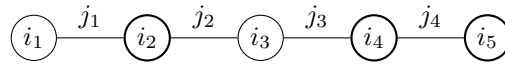


Рис. 2. Результат первой атаки

У Защитника есть также возможность переместить охранника из вершины i_2 в вершину i_1 или i_3 . Однако любое из этих перемещений нарушит вершинное покрытие. Перемещение охранника из вершины i_5 недопустимо, так как в единственную соседнюю вершину i_4 уже был перемещён охранник. Таким образом, единственное возможно перемещение охранников на первом шаге приводит к вершинному покрытию $\{i_2, i_4, i_5\}$.

На втором шаге Атакующий атакует ребро j_2 . Защитник обязан переместить охранника из вершины i_2 в вершину i_3 . Тогда получаем размещение охранников, показанное на рис. 3.

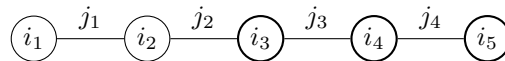


Рис. 3. Результат второй атаки

Других возможностей для перемещения охранников у Защитника нет, так как у охранников, находящихся в вершинах i_4 и i_5 , соседние вершины заняты. При этом ребро j_1 оказалось не защищённым. Таким образом, исходное покрытие $\{i_2, i_3, i_5\}$ не 2-устойчиво.

5. Вычислительные эксперименты

В ходе проведённых экспериментов исследованы вычислительные возможности предложенного алгоритма проверки устойчивости вершинных

покрытий, а также влияние на его производительность некоторых используемых процедур. Расчёты проводились на компьютере с четырёхъядерным процессором 3,3 ГГц и 16 ГБ оперативной памяти. Алгоритмы написаны на языке программирования Julia [10], а для поиска оптимальных решений возникающих задач целочисленного линейного программирования использовался решатель Cbc COIN-OR [11].

5.1. Генерация графов. В экспериментах используем случайно генерируемые связные графы разной плотности. Плотность определяется как отношение числа рёбер в графе к числу рёбер в клике с тем же числом вершин.

Построение графа с заданным числом вершин n начинается с построения остовного дерева. Такое дерево может иметь специальный вид (путь, звезда и т. д.), а может быть случайно сгенерированным. Случайно сгенерированное дерево получается в результате последовательного просмотра вершин и соединения рассматриваемой вершины со случайно выбранной ранее просмотренной вершиной. Затем случайным образом выбирается некоторое число дополнительных рёбер, каждое из которых соединяет некоторую пару случайно выбранных вершин.

5.2. Генерация покрытий. Поскольку нахождение минимального покрытия есть труднорешаемая задача, в экспериментах используются начальные покрытия, у которых ни одно из собственных подмножеств не остаётся покрытием. Такие покрытия назовём *неприводимыми*. Граф может иметь несколько неприводимых покрытий разных размеров, и генерация различных неприводимых покрытий в наших экспериментах организована как рандомизированный жадный алгоритм.

Алгоритм строит неприводимое покрытие в два этапа. В начале первого этапа все вершины графа считаются непомеченными. Далее последовательно случайным образом выбираются тройки непомеченных вершин и помечается та вершина, которая инцидентна наибольшему числу рёбер с непомеченными концевыми вершинами. Этот процесс повторяется до тех пор, пока не останется рёбер, у которых не помечены обе концевые вершины. На втором этапе в некотором случайно выбранном порядке просматриваются помеченные вершины. Если у всех рёбер, инцидентных рассматриваемой вершине, обе концевые вершины помечены, то рассматриваемая вершина становится непомеченной. Вершины, оставшиеся помеченными после второго этапа, образуют неприводимое вершинное покрытие.

5.3. Влияние списка проверенных покрытий. Здесь приводятся результаты вычислительного эксперимента, позволяющего оценить влияние использования списка проверенных вершинных покрытий на время

работы алгоритма проверки устойчивости покрытий. Рассмотрено 300 примеров вершинных покрытий различных графов с числом вершин 15, которые проверялись на 3-устойчивость. Для рассматриваемых примеров покрытий, которые оказались 3-устойчивыми, сравнивается время работы алгоритма с использованием списка проверенных покрытий и без использования списка. Время исчисляется числом обращений в ходе работы алгоритма к процедуре проверки устойчивости покрытия к атаке ребра, т. е. числом вычислений оптимального решения задачи Защитника. Для каждого примера вычисляется величина $s = (1 - \frac{t_1}{t_2}) \cdot 100$, где t_1 — время работы алгоритма с использованием списка, t_2 — время работы алгоритма без использования списка. Эта величина характеризует то, насколько велико уменьшение времени работы алгоритма при использовании списка по сравнению со временем работы алгоритма без использования списка.

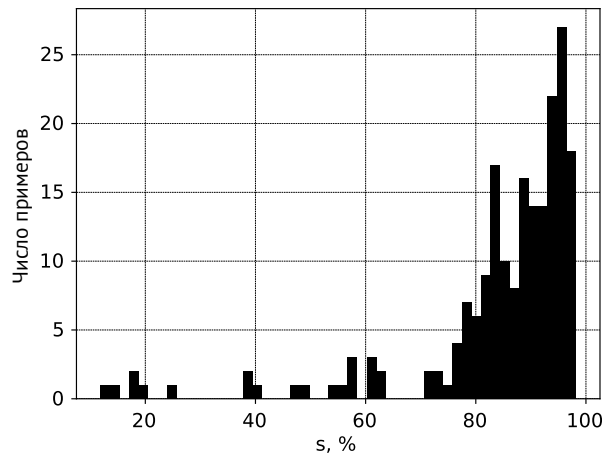


Рис. 4. Ускорение алгоритма при использовании списка

На рис. 4 представлена гистограмма распределения полученных значений s . По горизонтали указаны возможные значения s , а по вертикали число примеров, соответствующих данному s . Как можно видеть, в большинстве случаев использование списка сокращает время работы алгоритма более чем на 80%. Это, в частности, свидетельствует о большой повторяемости вершинных покрытий, рассматриваемых в ходе работы алгоритма.

5.4. Время работы алгоритма. Проведённые вычислительные эксперименты позволяют сделать некоторые выводы о времени работы алгоритма при проверке L -устойчивости вершинных покрытий в зависимости

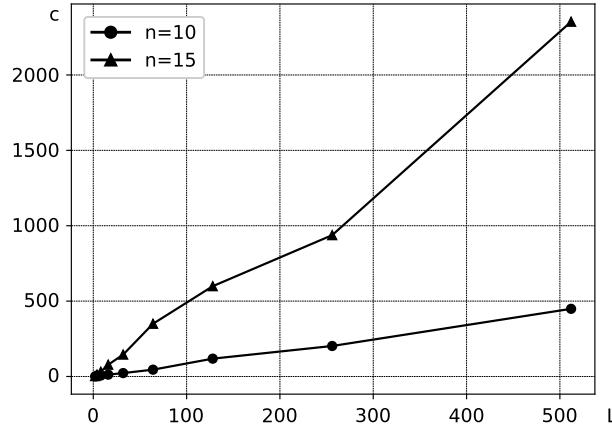


Рис. 5. Зависимость времени работы алгоритма от степени устойчивости L

от числа вершин графа и проверяемой степени устойчивости вершинного покрытия.

На рис. 5 показана зависимость времени работы алгоритма (в секундах) от степени устойчивости L для числа вершин $n \in \{10, 15\}$. В ходе эксперимента для каждого значения L и каждого числа вершин рассмотрено 75 примеров вершинных покрытий различных графов. Из этих покрытий были удалены неустойчивые, а для остальных вычислено среднее время работы алгоритма. Как можно видеть, временная сложность алгоритма возрастает линейно при увеличении L .

На рис. 6 показана зависимость времени работы алгоритма в секундах от числа вершин графа n для значения устойчивости $L \in \{1, 2, 3\}$. В ходе эксперимента для каждого n и каждого L сгенерировано 75 примеров

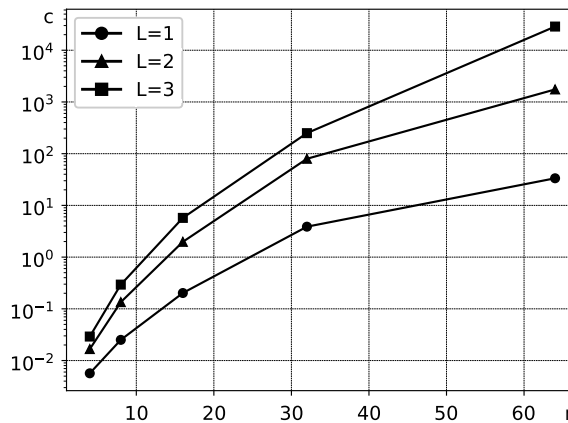


Рис. 6. Зависимость времени работы алгоритма от числа вершин n

вершинных покрытий различных графов с числом вершин n . Из этих покрытий удалены неустойчивые, а для остальных вычислено среднее время работы алгоритма. Заметим, что время проверки 3-устойчивости для графов размера до 32 вершин не превышает 16 минут.

5.5. Распределение степеней устойчивости. В этом пункте приведём сводные результаты численных экспериментов. В экспериментах рассмотрены графы с числом вершин до 64 и различной плотностью рёбер. Проверена L -устойчивость вершинных покрытий для значений L от 2 до 512. При каждом эксперименте, состоящем в проверке L -устойчивости рассматриваемого вершинного покрытия и завершённом установлением фактической степени устойчивости l , $0 \leq l \leq L$, этого вершинного покрытия, вычислена величина $r = \frac{l}{L}$. Значение $r = 0$ означает, что проверенное покрытие неустойчиво, а $r = 1$ — что проверка L -устойчивости рассматриваемого покрытия завершилась успешно. Результат $0 < r < 1$ означает, что степень устойчивости покрытия меньше проверяемой.

Из 2714 случайно сгенерированных и проверенных вершинных покрытий, 1207 попали в категорию неустойчивых с $r = 0$, а 87 — в промежуточную категорию с $0 < r < 1$: степень устойчивости оказалась ниже проверяемой. У оставшихся 1420 покрытий степень устойчивости совпала с проверяемой. Таким образом, подавляющее число рассматриваемых вершинных покрытий либо оказались неустойчивыми, либо имели степень устойчивости не меньше той, которая проверялась.

Заключение

В настоящей работе предложена процедура, позволяющая ответить на вопрос, существует ли выигрышная стратегия для Защитника, следуя которой он может защищать заданное вершинное покрытие на протяжении определённого конечного числа шагов. Для построения стратегии Защитника задача о вечном вершинном покрытии представляется как динамическая игра Штакельберга, на каждом шаге которой взаимодействие противоборствующих сторон формализуется в виде двухуровневой задачи математического программирования. Предложенная процедура протестирована на различных случайно сгенерированных графах и вершинных покрытиях.

Вычислительные эксперименты показали результаты, позволяющие предположить, что большая часть случайных покрытий либо неустойчивы, либо имеют проверяемую степень устойчивости. Таким образом, следующий шаг в исследовании рассматриваемой игры состоит в создании алгоритма, который за конечное число шагов сможет определить, является ли вершинное покрытие бесконечно устойчивым или нет.

Финансирование работы

Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева СО РАН (проект № FWNF-2022-0019).

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Klostermeyer W. F., Mynhardt C. M. Protecting a graph with mobile guards // *Appl. Anal. Discrete Math.* 2016. V. 10, No. 1. P. 1–29. DOI: 10.2298/AADM151109021K.
2. Klostermeyer W. F., Mynhardt C. M. Edge protection in graphs // *Australas. J. Comb.* 2009. V. 45. P. 235–250.
3. Fomin F. V., Gaspers S., Golovach P. A., Kratsch D., Saurabh S. Parameterized algorithm for eternal vertex cover // *Inf. Process. Lett.* 2010. V. 110, No. 17. P. 702–706. DOI: 10.1016/j.ipl.2010.05.029.
4. Babu J., Misra N., Nanoti S. G. Eternal vertex cover in bipartite graphs // *Computer science — Theory and applications. Proc. 17th Int. Comp. Sci. Symp. in Russia (St. Petersburg, Russia, June 29–July 1, 2022)*. Cham: Springer, 2022. P. 64–76. (Lect. Notes Comput. Sci.; Vol. 13296). DOI: 10.1007/978-3-031-09574-0_5.
5. Babu J., Prabhakaran V. A new lower bound for the eternal vertex cover number of graphs // *J. Comb. Opt.* 2022. V. 44. P. 2482–2498. DOI: 10.1007/s10878-021-00764-8.
6. Babu J., Prabhakaran V., Sharma A. A substructure based lower bound for eternal vertex cover number // *Theor. Comput. Sci.* 2021. V. 890. P. 87–104. DOI: 10.1016/j.tcs.2021.08.018.
7. Araki H., Fujito T., Inoue S. On the eternal vertex cover numbers of generalized trees // *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* 2015. V. E98-A, No. 6. P. 1153–1160. DOI: 10.1587/transfun.E98.A.1153.
8. Paul K., Pandey A. Some algorithmic results for eternal vertex cover problem in graphs // *WALCOM: Algorithms and computation. Proc. 17th Int. Conf. and Workshops (Hsinchu, Taiwan, Mar. 22–24, 2023)*. Cham: Springer, 2023. P. 242–253. (Lect. Notes Comput. Sci.; Vol. 13973). DOI: 10.1007/978-3-031-27051-2_21.
9. Beresnev V. L., Melnikov A. A., Utyupin S. Yu. Representation of the eternal vertex cover problem as a dynamic Stackelberg game // *Optimization and applications. Rev. Sel. Pap. 14th Int. Conf. (Petrovac, Montenegro, Sept. 18–22, 2023)* Cham: Springer, 2023. P. 3–13. (Lect. Notes Comput. Sci.; V. 14395). DOI: 10.1007/978-3-031-47859-8_1.
10. 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.

-
11. COIN-OR Branch-and-cut solver. Towson: COIN-OR Found., 2023. URL: github.com/coin-or/Cbc (accessed Mar. 27, 2024).

Береснев Владимир Леонидович
Мельников Андрей Андреевич
Утюпин Степан Юрьевич

Статья поступила
26 февраля 2024 г.
После доработки —
14 марта 2024 г.
Принята к публикации
21 марта 2024 г.

STABILITY OF VERTEX COVERS IN A GAME
WITH FINITELY MANY STEPSV. L. Beresnev^{1, a}, A. A. Melnikov^{1, b}, and S. Yu. Utyupin^{2, c}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, ^cstepan.utyupin@gmail.com

Abstract. The eternal vertex cover problem is a version of the graph vertex cover problem that can be represented as a dynamic game between two players (the Attacker and the Defender) with an infinite number of steps. At each step, there is an arrangement of guards over the vertices of the graph forming a vertex cover. When the Attacker attacks one of the graph's edges, the Defender must move the guard along the attacked edge from one vertex to the other. In addition, the Defender can move any number of other guards from their current vertices to some adjacent ones to obtain a new vertex cover. The Attacker wins if the Defender cannot build a new vertex cover after the attack.

In this paper, we propose a procedure that allows us to answer the question, whether there exists a winning Defender strategy that permits protecting a given vertex cover for a given finite number of steps. To construct the Defender strategy, the problem is represented as a dynamic Stackelberg game in which at each step the interaction of the opposing sides is formalized as a two-level mathematical programming problem. The idea of the procedure is to recursively check the 1-stability of vertex covers obtained as a result of solving lower-level problems and to use some information about the covers already considered. Illustr. 6, bibliogr. 11.

Keywords: dynamic Stackelberg game, eternal vertex cover, graph edge protection, stability check algorithm.

References

1. **W. F. Klostermeyer** and **C. M. Mynhardt**, Protecting a graph with mobile guards, *Appl. Anal. Discrete Math.* **10** (1), 1–29 (2016), DOI: 10.2298/AADM151109021K.
2. **W. F. Klostermeyer** and **C. M. Mynhardt**, Edge protection in graphs, *Australas. J. Comb.* **45**, 235–250 (2009).
3. **F. V. Fomin**, **S. Gaspers**, **P. A. Golovach**, **D. Kratsch**, and **S. Saurabh**, Parameterized algorithm for eternal vertex cover, *Inf. Process. Lett.* **110** (17), 702–706 (2010), DOI: 10.1016/j.ipl.2010.05.029.
4. **J. Babu**, **N. Misra**, and **S. G. Nanoti**, Eternal vertex cover in bipartite graphs, in *Computer Science — Theory and Applications* (Proc. 17th Int. Comp. Sci. Symp. in Russia, St. Petersburg, Russia, June 29 – July 1, 2022) (Springer, Cham, 2022), pp. 64–76 (Lect. Notes Comput. Sci., Vol. 13296), DOI: 10.1007/978-3-031-09574-0_5.
5. **J. Babu** and **V. Prabhakaran**, A new lower bound for the eternal vertex cover number of graphs, *J. Comb. Opt.* **44**, 2482–2498 (2022), DOI: 10.1007/s10878-021-00764-8.
6. **J. Babu**, **V. Prabhakaran**, and **A. Sharma**, A substructure based lower bound for eternal vertex cover number, *Theor. Comput. Sci.* **890**, 87–104 (2021), DOI: 10.1016/j.tcs.2021.08.018.
7. **H. Araki**, **T. Fujito**, and **S. Inoue**, On the eternal vertex cover numbers of generalized trees, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E98-A** (6), 1153–1160 (2015), DOI: 10.1587/transfun.E98.A.1153.
8. **K. Paul** and **A. Pandey**, Some algorithmic results for eternal vertex cover problem in graphs, in *WALCOM: Algorithms and Computation* (Proc. 17th Int. Conf. and Workshops, Hsinchu, Taiwan, Mar. 22–24, 2023) (Springer, Cham, 2023), pp. 242–253 (Lect. Notes Comput. Sci., Vol. 13973), DOI: 10.1007/978-3-031-27051-2_21.
9. **V. L. Beresnev**, **A. A. Melnikov**, and **S. Yu. Utyupin**, Representation of the eternal vertex cover problem as a dynamic Stackelberg game, in *Optimization and Applications* (Rev. Sel. Pap. 14th Int. Conf., Petrovac, Montenegro, Sept. 18–22, 2023) (Springer, Cham, 2023), pp. 3–13 (Lect. Notes Comput. Sci., Vol. 14395), DOI: 10.1007/978-3-031-47859-8_1.
10. **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.
11. COIN-OR Branch-and-Cut solver (COIN-OR Found., Towson, 2023). URL: github.com/coin-or/Cbc (accessed Mar. 27, 2024).

Vladimir L. Beresnev
 Andrey A. Melnikov
 Stepan Yu. Utyupin

Received February 26, 2024
 Revised March 14, 2024
 Accepted March 21, 2024

КООПЕРАТИВНЫЕ ИГРЫ С ПРЕДПОЧТЕНИЯМИ:
ПРИЛОЖЕНИЕ ВЕСОВОГО ПРАВИЛА
К ПРОБЛЕМАМ ОБЩЕСТВЕННОГО ПРОСТРАНСТВА
В ГОРОДЕ САНКТ-ПЕТЕРБУРГ

В. В. Гусев

Национальный исследовательский университет «Высшая школа экономики»,
ул. Кантемировская, 3, 194100, Санкт-Петербург, Россия

E-mail: vgusev@hse.ru

Аннотация. Исследуется проблема распределения общественного пространства с использованием методов кооперативной теории игр для её решения. Игроки — это районы, значение характеристической функции — это суммарное число людей, которые заинтересованы в конкретном виде общественного пространства в рассматриваемых районах. Составлены аксиомы, характерные для проблемы дележа. Выведено специальное значение кооперативной игры, которое зависит от весов игроков. Показано, как подобрать веса оптимизационными методами. Табл. 1, ил. 2, библиогр. 13.

Ключевые слова: теория игр, кооперативная игра, весовое правило, игра с предпочтениями.

Введение

Описание проблемы и метод её решения. По данным администрации г. Санкт-Петербург численность города достигает 7 млн человек на 2021 год. Из-за многомиллионной численности населения город сталкивается с проблемой распределения общественного пространства. Общественные пространства включают в себя все места, являющиеся общественной собственностью. Сюда входят места общественного пользования, открытые и доступные для всех жителей и гостей города. Примерами общественных мест являются детские площадки, парки, автомобильные стоянки, скверы, стадионы и т. п. Инфраструктура города расширяется, в районах города строятся новые жилые комплексы. Это делает актуальным вопрос о выборе типа общественного пространства в конкретном месте.

Предположим, что в некотором новом районе есть место под общественное пространство. Что лучше построить: детскую площадку или автостоянку? От выбора общественного пространства зависит привлекательность района для разных групп населения. Автомобилисты хотят автостоянку, молодые семьи предпочитают детские площадки, и те и другие были бы не против парка. Возникает вопрос: как именно распределить имеющийся ресурс (парки, автостоянки, детские площадки и т. п.) между районами?

Предположим, что требуется определить, какую площадь общественного пространства нужно занять под парки в каждом районе. Чтобы решить проблему распределения ресурса, пользуемся методами кооперативной теории игр. Кооперативная игра задаётся множеством игроков $N = \{1, 2, \dots, n\}$ и характеристической функцией $v: 2^N \rightarrow \mathbb{R}$ такой, что $v(\emptyset) = 0$. Множество игроков — это множество районов, а характеристическая функция показывает число людей (измеряется в сотнях или тысячах человек), которые будут пользоваться общественным пространством.

Например, для $N = \{1, 2, 3\}$ характеристическую функцию вида

$$\begin{aligned} v(\{1\}) = 5, \quad v(\{2\}) = 6, \quad v(\{3\}) = 10, \quad v(\{1, 2\}) = 15, \\ v(\{1, 3\}) = 20, \quad v(\{2, 3\}) = 25, \quad v(\{1, 2, 3\}) = 50 \end{aligned}$$

интерпретируем следующим образом. Если в районе 1 обустроить парк, то по прогнозам им будут пользоваться в среднем 5 сотен человек. Если расположить парки в районах 1 и 2, то ими будут пользоваться в среднем 15 сотен человек.

Монотонность и супераддитивность характеристической функции зависят от районов и их расположения. Иногда жители города готовы ездить в центральные районы для проведения своего досуга. Если в районе, близком к центральному, разместить общественное пространство, то это может значительно привлечь отдыхающих в район. Это говорит о том, что значение $v(K)$ может превышать число $\sum_{i \in K} v(\{i\})$ для некоторых

$K \subseteq N$, $K \neq \emptyset$.

Рассчитывая значение кооперативной игры, можно определить, какой район наиболее предпочтителен для размещения в нём конкретного общественного пространства. Однако, у самого города есть предпочтения относительно размещения общественных мест в городе. Может оказаться так, что город хочет разгрузить центральный район и намеренно размещает общественное пространство в близко расположенных районах. Тем самым в кооперативной модели нам нужно учесть предпочтения города, которые в статье обозначаем отношением $\succ$. Запись $i \succ j$ означает, что игрок (район) i предпочтительнее игрока j , где $i, j \in N$.

Характеристическая функция кооперативной игры хранит в себе информацию о числе жителей, которые заинтересованы в конкретном общественном пространстве, а предпочтения $\succ$ отражают намерения и способности города разместить в том или ином районе общественное пространство.

Для решения задачи распределения ресурса используется следующий подход. Сначала формулируем аксиомы, которые характерны для задачи распределения общественного пространства. С помощью аксиом выводим значение кооперативной игры, в котором у каждого игрока есть свой вес в коалиции. Вес одного и того же игрока в разных коалициях разный. Далее, используя аксиоматические и оптимизационные методы, выбираем веса игроков некоторым справедливым или оптимальным способом.

Отметим, что большинство работ в области исследования операций, урбанизации, индустриальной организации используют только оптимизационные и эмпирические методы решения проблем общественных пространств [1–3]. Здесь мы показываем, как можно решить задачу распределения общественного пространства по районам города с помощью методов кооперативной теории игр, учитывая предпочтения общества и города.

Результаты. В ходе исследования проблемы распределения общественного пространства получены следующие основные результаты:

- составлена аксиоматика для весового экспертного значения кооперативной игры (теорема 1);
- рассчитана оптимальная площадь для единственного вида общественного пространства (теорема 2).

С помощью методов кооперативной теории игр в статье выводится специальное значение кооперативной игры. Сформулированные аксиомы имеют физический смысл для решения задачи распределения общественного пространства. В теореме 1 показано, что аксиомам линейности, эффективности и предпочтения удовлетворяет значение кооперативной игры такое, что у каждого игрока существует вес для каждой коалиции. Стоит вопрос о том, как подобрать веса игроков. Для этого в статье предлагается два подхода. Первый подход аксиоматический, второй — оптимизационный. С помощью оптимизационного подхода показано, как можно рассчитать оптимальную площадь для единственного вида общественного пространства.

1. Основные обозначения и определения

Кооперативная игра — это пара (N, v) , где $N = \{1, 2, \dots, n\}$ — множество игроков и $v: 2^N \rightarrow \mathbb{R}$, $v(\emptyset) = 0$, — характеристическая функция.

Обозначим через $G(N)$ множество всех характеристических функций для множества игроков N , т. е. $G(N) = \{v: 2^N \rightarrow \mathbb{R} \mid v(\emptyset) = 0\}$. *Значением кооперативной игры* называется отображение $\varphi: G(N) \rightarrow \mathbb{R}^n$. Число $\varphi_i(v)$ называется *значением игрока $i \in N$ в кооперативной игре (N, v)* .

Пусть $\emptyset \subset K \subseteq N$. Единичная кооперативная игра — это пара (N, v_K) , где

$$v_K(S) = \begin{cases} 1, & \text{если } S = K, \\ 0, & \text{если } S \neq K. \end{cases}$$

Множество характеристических функций $\{v_K \mid \emptyset \subset K \subseteq N\}$ образует базис в пространстве $G(N)$. Характеристическая функция $v \in G(N)$ может быть представлена единственным образом в виде линейной комбинации характеристических функций единичных кооперативных игр следующим образом:

$$v(S) = \sum_{\emptyset \subset K \subseteq N} v(K) v_K(S), \quad S \subseteq N.$$

Будем предполагать, что существует общий порядок предпочтений игроков $1 \succeq 2 \succeq \dots \succeq n$.

2. Весовое экспертное значение

В этом разделе вводим аксиомы, которые учитывают предпочтения в кооперативной игре.

Аксиома предпочтения. Для любых $i, j \in N$, $K \subseteq N$, $K \neq \emptyset$, если $i \succ j$, то $\varphi_i(v_K) \geq \varphi_j(v_K)$.

В аксиоме предпочтения задано отношение предпочтительности между игроками для любой единичной игры. Обычно в игре (N, v_K) агенты из коалиции K получают выигрыш не меньше нуля, а выигрыш каждого игрока из множества $N \setminus K$ — это нуль. Такое свойство дележа обусловлено только математическими свойствами единичных кооперативных игр. В действительности нулевые игроки могут получать ненулевой выигрыш. Аксиома предпочтения это учитывает. Например, если значение игрока i в игре (N, v_K) неотрицательно и для игрока $j \in N \setminus K$ имеет место $j \succ i$, то согласно аксиоме предпочтения $\varphi_j(v_K) \geq 0$.

Аксиома линейности. Для любых характеристических функций $v, w \in G(N)$ и констант $\alpha, \beta \in \mathbb{R}$ имеем

$$\varphi(\alpha v + \beta w) = \alpha \varphi(v) + \beta \varphi(w).$$

Аксиома эффективности. Для любой характеристической функции $v \in G(N)$ имеем

$$\sum_{i \in N} \varphi_i(v) = v(N).$$

Теорема 1. Значение φ удовлетворяет аксиомам предпочтения, линейности и эффективности, если и только если существуют числа w_S^i , $i \in N$, $\emptyset \subset S \subseteq N$, такие, что

$$\sum_{j \in N} w_S^j = \begin{cases} 1, & \text{если } S = N, \\ 0, & \text{если } S \neq N, \end{cases} \quad (1)$$

и значение игрока $i \in N$ в кооперативной игре (N, v) равно

$$\varphi_i(v) = \sum_{\emptyset \subset S \subset N} w_S^i \cdot v(S) + \frac{w_N^i}{\sum_{j \in N} w_N^j} \cdot v(N).$$

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ. Пусть существуют числа w_S^i , $\emptyset \subset S \subseteq N$, такие, что имеет место (1), а значение $\varphi_i(v)$ определено в условии теоремы. Покажем, что выполняются аксиомы линейности, эффективности и предпочтения.

ЛИНЕЙНОСТЬ. Так как $\varphi_i(v)$ задано линейным выражением относительно переменных $v(S)$, $\emptyset \subset S \subseteq N$, то $\varphi_i(v)$ удовлетворяет условию линейности.

ЭФФЕКТИВНОСТЬ. Покажем, что сумма значений игроков равна $v(N)$:

$$\begin{aligned} \sum_{i \in N} \varphi_i(v) &= \sum_{i \in N} \left(\sum_{\emptyset \subset S \subset N} w_S^i \cdot v(S) + \frac{w_N^i}{\sum_{j \in N} w_N^j} \cdot v(N) \right) = \\ &= \sum_{\emptyset \subset S \subset N} v(S) \cdot \underbrace{\sum_{i \in N} w_S^i}_0 + \frac{\sum_{i \in N} w_N^i}{\sum_{j \in N} w_N^j} \cdot v(N) = v(N). \end{aligned}$$

АКСИОМА ПРЕДПОЧТЕНИЯ. Пусть $\emptyset \subset K \subseteq N$. Вычислим

$$\varphi_i(v_K) = \sum_{\emptyset \subset S \subset N} w_S^i \cdot v_K(S) + \frac{w_N^i}{\sum_{j \in N} w_N^j} \cdot v_K(N) = \begin{cases} w_K^i, & K \neq N, \\ \frac{w_N^i}{\sum_{j \in N} w_N^j}, & K = N. \end{cases}$$

Согласно аксиоме предпочтения если $i \succ j$, то должно выполняться $\varphi_i(v_K) \geq \varphi_j(v_K)$. Учитывая явный вид значения $\varphi_i(v_K)$, неравенство $\varphi_i(v_K) \geq \varphi_j(v_K)$ верно, если и только если $w_K^i \geq w_K^j$ для любого $K \subseteq N$, $K \neq \emptyset$. Следовательно, существуют веса w_K^i , $i \in N$, $\emptyset \subset K \subseteq N$, при которых аксиома предпочтения выполняется.

ДОСТАТОЧНОСТЬ. Пусть φ удовлетворяет аксиомам предпочтения, линейности и эффективности. Покажем, что существуют веса игроков, для которых φ определено в условии теоремы.

Можем представить любую характеристическую функцию в виде линейной комбинации единичных характеристических функций:

$$v(S) = \sum_{\emptyset \subset K \subseteq N} v(S)v_K(S), \quad S \subseteq N.$$

Так как φ удовлетворяет свойству линейности, имеем

$$\varphi(v) = \sum_{\emptyset \subset K \subseteq N} v\varphi(v_K).$$

Пусть $v = v_K$ и $i_1 \succ_K i_2 \succ_K \dots \succ_K i_n$, где $(i_1, i_2, \dots, i_n)$ — некоторая перестановка игроков. Из аксиомы предпочтения и эффективности получаем

$$\begin{aligned} \varphi_{i_1}(v_K) &\geq \varphi_{i_2}(v_K) \geq \dots \geq \varphi_{i_n}(v_K), \\ \sum_{i \in N} \varphi_i(v_K) &= v(K) = \begin{cases} 0, & \text{если } K \neq N, \\ 1, & \text{если } K = N. \end{cases} \end{aligned}$$

Следовательно, существуют описанные веса игроков такие, что выигрыш игрока i в единичной кооперативной игре (N, v_K) можно представить в виде

$$\varphi_i(v_K) = \begin{cases} w_K^i, & \text{если } K \neq N, \\ \frac{w_N^i}{\sum_{j \in N} w_N^j}, & \text{если } K = N. \end{cases}$$

Значит, значение $\varphi_i(v)$ можно записать следующим образом:

$$\varphi_i(v) = \sum_{\emptyset \subset K \subseteq N} v\varphi_i(v_K) = \sum_{\emptyset \subset S \subseteq N} w_S^i \cdot v(S) + \frac{w_N^i}{\sum_{j \in N} w_N^j} \cdot v(N).$$

Теорема 1 доказана.

Значение $\varphi_i(v)$, определённое в теореме 1, будем называть *весовым экспертным значением* игрока i в кооперативной игре (N, v) . Значение $\varphi(v)$ называем экспертным, поскольку некий эксперт даёт нам информацию о предпочтениях для каждой единичной кооперативной игры. Обычно такую информацию исследователи задают самостоятельно путём введения некоторых аксиом (нулевые игроки получают нуль, выигрыши симметричных игроков равны и т. п.). Наш подход шире, так как веса игроков можно задать исходя из аксиом или с помощью эксперта.

Заметим, что если $w_K^i = 0$ для любых $i \in N$ и $K \subset N$, то получим пропорциональный делёж (proportional rule) [4, 5]

$$\varphi_i(v) = \frac{w_N^i}{\sum_{j \in N} w_N^j} \cdot v(N),$$

частным случаем которого является равномерный делёж (equal division value) [6–8]

$$\varphi_i(v) = \frac{v(N)}{|N|}.$$

Аксиоматика равномерного дележа приведена в работе [9], в основе которой лежит аксиома аннулирующего игрока (nullifying players). Игрок $i \in N$ в кооперативной игре (N, v) называется *аннулирующим*, если $v(S \cup \{i\}) = 0$ для любого $S \subseteq N \setminus \{i\}$. Тогда согласно рассматриваемой аксиоме выигрыш аннулирующего игрока — это нуль. В зависимости от весов игроков весовое экспертное значение аннулирующего игрока может быть как равно нулю, так и нет.

Подобрав веса специальным образом, так же можем получить другие классические дележи. Например, когда

$$w_K^i = \begin{cases} \frac{1}{|N|}, & \text{если } K = N, \\ -\frac{1}{|N|}, & \text{если } K = \{j\}, j \neq i, \\ 1 - \frac{1}{|N|}, & \text{если } K = \{i\}, \\ 0 & \text{иначе,} \end{cases}$$

получим делёж с равно пропорциональным делением остатка (equal surplus division solution) [10, 11]

$$\varphi_i(v) = v(\{i\}) + \frac{1}{|N|} \left(v(N) - \sum_{j \in N} v(\{j\}) \right).$$

Когда веса имеют вид

$$w_K^i = \begin{cases} \frac{(|K|-1)! (|N|-|K|)!}{|N|!}, & \text{если } i \in K, \\ -\frac{(|K|-1)! (|N|-|K|)!}{|N|!}, & \text{если } i \notin K, \end{cases}$$

весовое экспертное значение — это значение Шепли [12, 13]

$$\varphi_i(v) = \sum_{i \in K \subseteq N} \frac{(|K|-1)! (|N|-|K|)!}{|N|!} \cdot (v(K) - v(K \setminus \{i\})).$$

Заметим, что для перечисленных дележей при любом $\emptyset \subset K \subseteq N$ веса удовлетворяют ограничению теоремы 1, а именно:

$$\sum_{j \in N} w_K^j = \begin{cases} 1, & \text{если } K = N, \\ 0, & \text{если } K \neq N. \end{cases}$$

Весовое экспертное значение охватывает достаточно широкий класс дележей из-за того, что вес одного и того же игрока в разных коалициях различен. Однако мы не можем подбирать веса произвольным образом, так как они хранят в себе информацию о предпочтениях в единичных

кооперативных играх. Возникает вопрос: как задать числа w_K^i , $i \in N$, $\emptyset \subset K \subseteq N$, зная предпочтения эксперта. Далее сформулируем аксиомы, которые позволяют определить веса игроков единственным образом.

3. Аксиоматические методы нахождения весов

В этом разделе вводим аксиомы, которые характерны для задачи распределения ресурса и позволяют упростить весовое экспертное значение каждого игрока.

Пусть имеется только два игрока, т. е. $N = \{1, 2\}$. Напомним, что игрок $i \in N$ аннулирующий в игре (N, v) , если $v(K \cup \{i\}) = 0$ для любого $K \subseteq N \setminus \{i\}$.

Аксиома сверхнулевого игрока. Если в кооперативной игре (N, v) игрок $i \in N$ аннулирующий и $j \succeq i$ для любого $j \in N \setminus \{i\}$, то $\varphi_i(v) = 0$.

Если игрок является одновременно аннулирующим и самым не предпочтительным, то согласно аксиоме сверхнулевого игрока его выигрыш равен нулю.

Аксиома равенства критериев. Если $i, j \in N$ — аннулирующий и не аннулирующий игроки соответственно в кооперативной игре (N, v) , причём $i \succeq j$, то $\varphi_i(v) = \varphi_j(v)$.

С одной стороны, если игрок $i \in N$ аннулирующий, а $j \in N$ нет, то $\varphi_j(v) \geq \varphi_i(v)$. С другой стороны, игрок i предпочтительнее игрока j , т. е. $i \succeq j$, и должно выполняться $\varphi_i(v) \geq \varphi_j(v)$. Если выполняется $\varphi_j(v) > \varphi_i(v)$ или $\varphi_j(v) < \varphi_i(v)$, то свойство аннулирования доминирует свойство предпочтений, или наоборот. В аксиоме равенства критериев предполагается, что $\varphi_i(v) = \varphi_j(v)$, т. е. нельзя выделить доминирующий критерий.

Утверждение. Пусть $N = \{1, 2\}$, $1 \succeq 2$. Значение $\varphi: G(\{1, 2\}) \rightarrow \mathbb{R}^2$ удовлетворяет аксиомам сверхнулевого игрока, равенства критериев, линейности и эффективности, если и только если существуют $w_1, w_2 \in \mathbb{R}$, $w_1 \geq w_2$, такие, что

$$\varphi_1(v) = \frac{w_1}{w_1 + w_2} \cdot v(\{1, 2\}), \quad \varphi_2(v) = \frac{w_2}{w_1 + w_2} \cdot v(\{1, 2\}).$$

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ. Покажем, что

$$\varphi(v) = \left(\frac{w_1}{w_1 + w_2} \cdot v(\{1, 2\}), \frac{w_2}{w_1 + w_2} \cdot v(\{1, 2\}) \right)$$

удовлетворяет аксиомам из утверждения. Аксиома линейности выполняется, так как $\varphi_i(v)$, $i \in \{1, 2\}$, выражается линейно через $v(\{1, 2\})$. Поскольку $\varphi_1(\{1, 2\}) + \varphi_2(\{1, 2\}) = v(\{1, 2\})$, верна аксиома эффективности.

Пусть игрок 2 аннулирующий, тогда $v(\{2\}) = v(\{1, 2\}) = 0$. Так как $1 \succ 2$, то 2 — сверхнулевой игрок и должно выполняться $\varphi_2(v) = 0$. Аксиома сверхнулевого игрока выполняется, поскольку

$$\varphi_2(v) = \frac{w_2}{w_1 + w_2} \cdot v(\{1, 2\}) = 0.$$

Пусть только игрок 1 аннулирующий, тогда $v(\{1\}) = v(\{1, 2\}) = 0$. Так как $1 \succ 2$, то согласно аксиоме равенства критериев должно выполняться $\varphi_1(v) = \varphi_2(v)$, что верно, поскольку $\varphi_1(v) = \varphi_2(v) = 0$. Таким образом, φ удовлетворяет аксиомам из утверждения.

ДОСТАТОЧНОСТЬ. Покажем, что значение $\varphi(v)$, удовлетворяющее перечисленным аксиомам, единственно.

Из аксиомы линейности следует, что

$$\begin{aligned}\varphi_1(v) &= v(\{1\})\varphi_1(v_1) + v(\{2\})\varphi_1(v_2) + v(\{1, 2\})\varphi_1(v_{1,2}), \\ \varphi_2(v) &= v(\{1\})\varphi_2(v_1) + v(\{2\})\varphi_2(v_2) + v(\{1, 2\})\varphi_2(v_{1,2}).\end{aligned}$$

В игре (N, v_1) игрок 2 сверхнулевой, поэтому $\varphi_2(v_1) = 0$. Из аксиомы эффективности следует, что $\varphi_1(v_1) + \varphi_2(v_1) = v_1(\{1, 2\}) = 0$, значит, $\varphi_1(v_1) = \varphi_2(v_1) = 0$.

Для игры (N, v_2) из выполнения аксиом равенства критериев и эффективности следует, что $\varphi_1(v_2) = \varphi_2(v_2)$ и

$$\varphi_1(v_2) + \varphi_2(v_2) = v_2(\{1, 2\}) = 0,$$

значит, $\varphi_1(v_2) = \varphi_2(v_2) = 0$.

Для игры (N, v_{12}) игроки 1 и 2 не аннулирующие. Из аксиомы эффективности следует, что $\varphi_1(v_{12}) + \varphi_2(v_{12}) = v_{12}(\{1, 2\}) = 1$. Следовательно, существуют $w_1, w_2 \in \mathbb{R}$ такие, что

$$\varphi_1(v_{12}) = \frac{w_1}{w_1 + w_2}, \quad \varphi_2(v_{12}) = \frac{w_2}{w_1 + w_2}.$$

Так как разложение характеристической функции v по базису единственно и значения $\varphi_i(v_K)$, $i \in N$, $\emptyset \subset K \subseteq N$, вычисляются однозначно, значения $\varphi_i(v)$, $i \in \{1, 2\}$, определяются единственным образом:

$$\begin{aligned}\varphi_1(v) &= v(\{1\})\varphi_1(v_1) + v(\{2\})\varphi_1(v_2) + \\ &\quad + v(\{1, 2\})\varphi_1(v_{1,2}) = \frac{w_1}{w_1 + w_2} \cdot v(\{1, 2\}), \\ \varphi_2(v) &= v(\{1\})\varphi_2(v_1) + v(\{2\})\varphi_2(v_2) + \\ &\quad + v(\{1, 2\})\varphi_2(v_{1,2}) = \frac{w_2}{w_1 + w_2} \cdot v(\{1, 2\}).\end{aligned}$$

Утверждение доказано.

Упрощая весовое экспертное значение путём расширения списка аксиом, можно получить известные значения игры или их аналоги. Так, например, в утверждении 1 получено весовое значение для двух игроков. Добавление новых аксиом в аксиоматику весового экспертного значения будет приводить к обнулению или равенству весов либо получится некоторое другое элементарное упрощение. Если действительно нужно добавить новые аксиомы, то желательно проверить, подойдут ли классические дележи в качестве решения задачи распределения ресурса.

Далее фокусируемся на оптимизационных методах нахождения весов.

4. Оптимизационные методы нахождения весов

4.1. Задача выпуклого программирования для одного типа общественного пространства. В этом разделе показываем, как можно найти веса игроков с помощью методов оптимизации.

Так как игрок — это район, а значение характеристической функции $v(K)$, $K \subseteq N$, — это число людей, которые заинтересованы в общественном пространстве в районах из K , значение $\varphi_i(v)$ показывает число игроков, которые будут пользоваться общественным пространством в районе i .

Можем учитывать значения $\varphi_i(v)$, $i \in N$, при распределении общественного пространства. Площадь, выделяемую под общественное пространство, можно считать равной $\alpha_i \varphi_i^2(v)$, где $\alpha_i \in \mathbb{R}^+$, $i \in N$. Мы заинтересованы в таком распределении, которое удовлетворяет человеческим нуждам и при этом занимает минимальную площадь. Стало быть, для задачи распределения общественного пространства достаточно минимизировать суммарную площадь районов, выделяемую под общественное пространство:

$$F(w) = \sum_{i \in N} \alpha_i \varphi_i^2(v) \rightarrow \min_{\substack{w_S^i, \\ i \in S \subseteq N}},$$

$$\sum_{i \in S} w_S^i = \begin{cases} 0, & \text{если } S \subset N, \\ 1, & \text{если } S = N, \end{cases}$$

$$w_N^i \geq 0, \quad i \in N.$$

Получили оптимизационную модель для нахождения весов игроков. Числа $\varphi_i(v)$, $i \in N$, зависят от w_S^i , $\emptyset \subset S \subseteq N$, а $w = \{w_S^i\}_{i \in N, S \subseteq N}$.

Теорема 2. Суммарная оптимальная площадь, занятая под единственный тип общественного пространства, равна

$$F(w^*) = \frac{v(N)}{\sum_{i \in N} \alpha_i^{-1}}.$$

ДОКАЗАТЕЛЬСТВО. Функция Лагранжа задачи минимизации $F(w)$ имеет вид

$$L(w, \lambda) = \sum_{i \in N} \alpha_i \left(\sum_{\emptyset \subset S \subseteq N} w_S^i \cdot v(S) \right)^2 + \sum_{\emptyset \subset S \subset N} \lambda_S \sum_{i \in S} w_S^i + \lambda_N \left(\sum_{i \in N} w_N^i - 1 \right).$$

Целевая функция выпуклая, а ограничения системы линейны. Запишем необходимые условия для нахождения стационарной точки функции Лагранжа:

$$\begin{aligned} \frac{\partial L}{\partial w_K^i} &= 2\alpha_i v(K) \sum_{\emptyset \subset S \subseteq N} w_S^i v(S) + \lambda_K = 0, \quad \emptyset \subset K \subseteq N, \\ \sum_{i \in S} w_S^i &= \begin{cases} 0, & \text{если } S \subset N, \\ 1, & \text{если } S = N, \end{cases} \\ w_N^i &\geq 0, \quad i \in N. \end{aligned}$$

Пусть $w_K^i = 0$ для любых $i \in K$, $\emptyset \subset K \subset N$. Найдём значения w_N^i из системы

$$\begin{cases} 2w_N^i \alpha_i v(N) + \lambda = 0, & i \in N, \\ \sum_{i \in N} w_N^i = 1. \end{cases}$$

Получим

$$w_N^i = \frac{\alpha_i^{-1}}{\sum_{j \in N} \alpha_j^{-1}}, \quad i \in N,$$

и числа λ_K , $\emptyset \subset K \subseteq N$ имеют вид

$$\lambda_K = -\frac{2v(N)}{\sum_{j \in N} \alpha_j^{-1}} \cdot v(K).$$

Числа w_N^i удовлетворяют ограничению неотрицательности. Таким образом, веса игроков имеют вид

$$\begin{aligned} w_K^i &= 0, \quad i \in K, \emptyset \subset K \subset N, \\ w_N^i &= \frac{\alpha_i^{-1}}{\sum_{j \in N} \alpha_j^{-1}}, \quad i \in N. \end{aligned}$$

Поставляя найденные веса игроков в целевую функцию, получим

$$F(w^*) = \frac{v(N)}{\sum_{i \in N} \alpha_i^{-1}}.$$

Теорема 2 доказана.

4.2. Числовой пример. Парки, сады, парковки могут расширяться или, наоборот, их территория может уменьшиться. Так, например, парковку могут закрыть полностью или отдать под неё близлежащие территории. Рассмотрим задачу расширения парков. Анализируем только парки Калининского района города Санкт-Петербург, изображённые на рис. 1.

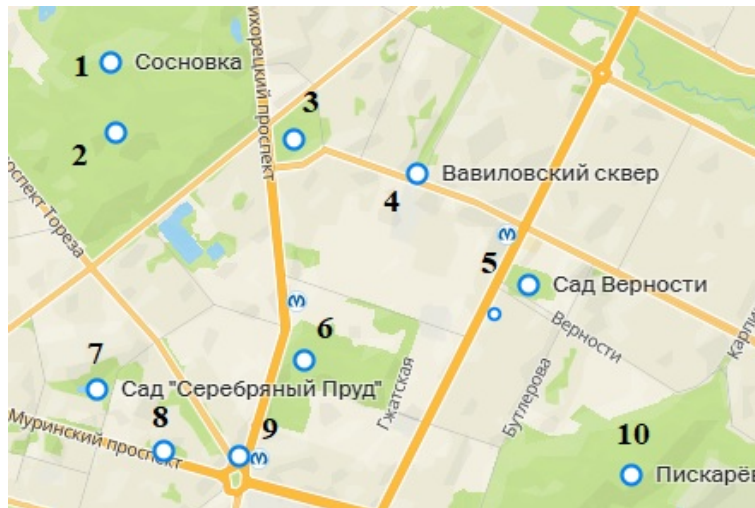


Рис. 1. Карта парков

Названия парков и нормированное число жителей вблизи парка (a_i) приведены в табл. 1.

Если некоторый человек находится в парке, то он может перейти в расположенный рядом парк. Также жители могут выбрать парк, который расположен не рядом с ними. Предполагаем, что переходы людей между парками происходят, если между соответствующими парками существует ребро в весовом графе парков, изображённом на рис. 2.

Вес ребра показывает связь между графами. Чем меньше вес ребра, тем менее вероятно, что люди будут переходить между графами.

Таблица 1

Парки и число жителей вблизи них

№	Парк	a_i
1	Сосновка	0,14
2	Центральная аллея парка «Сосновка»	0,14
3	Сад Бенуа	0,07
4	Вавилонский сквер	0,06
5	Сад Верности	0,09
6	Парк Политехнического университета	0,13
7	Сад «Серебряный пруд»	0,08
8	Сквер Академика Михачёва	0,08
9	Сквер Блокадников	0,08
10	Пискаревский парк	0,13

Построим характеристическую функцию кооперативной игры (N, v) , $N = \{1, 2, \dots, 10\}$, следующим образом:

$$v(\{i\}) = a_i - \sum_{j \in N_i} r_{ij} a_i + \sum_{j \in N_i} r_{ji} a_j, \quad i \in N,$$

$$v(K) = \sum_{i \in K} v(\{i\}), \quad \emptyset \subset K \subseteq N,$$

где N_i , $i \in N$, — множество парков таких, что для любого $j \in N_i$ существует ребро $\{i, j\}$ в графе на рис. 2. Число r_{ij} — это доля людей, которые переходят из парка i в парк j . Считаем, что $r_{ij} = r_{ji}$. Можно сказать, что числа r_{ij} хранят в себе только информацию о расстоянии между парками i и j . Значения чисел r_{ij} отображены на рис. 2.

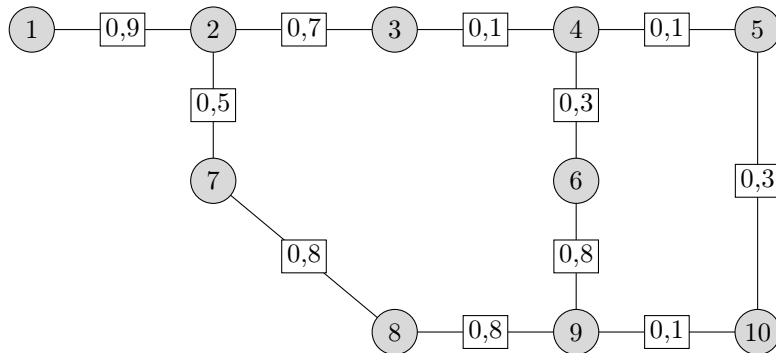


Рис. 2. Весовой граф

Пусть $\alpha_i = a_i^{-1}$, $i \in N$. Тогда согласно теореме 2 оптимальная площадь парков для коалиции N равна

$$\frac{v(N)}{\sum_{i \in N} \alpha_i^{-1}} = \frac{v(N)}{\sum_{i \in N} a_i} = 1.$$

В силу нормировки и свойства $v(K) = \sum_{i \in K} v(\{i\})$ получаем единичную оптимальную площадь. Решая оптимизационную задачу для парков 1 и 2, получаем

$$\frac{v(\{1\})}{\alpha_1^{-1}} = \frac{a_1 - 0,9a_1 + 0,9a_2}{a_1} = 1,$$

$$\frac{v(\{2\})}{\alpha_2^{-1}} = \frac{a_2 - 0,9a_2 - 0,7a_2 - 0,5a_2 + 0,9a_1 + 0,7a_3 + 0,5a_7}{a_2} = 0,025.$$

Таким образом, площадь парка 1 (Сосновка) должна быть в среднем в 40 раз больше площади парка 2 (Центральная аллея), что соответствует действительности. Рассчитывая оптимальные значения площадей парков можно понять, какие парки нуждаются в расширении.

Заключение

Проблемы распределения общественных пространств актуальны для больших городов и требуют математических подходов к решению. Для решения таких проблем нужно учитывать мнения жителей города, его инфраструктуру, бюджетные ограничения. В настоящей статье показано, что мнение жителей можно учесть с помощью предпочтений в кооперативной игре и вывести соответствующий делёж. Чтобы среди допустимых дележей выделить единственный делёж, можно использовать как аксиоматические, так и оптимизационные подходы.

Финансирование работы

Исследование выполнено за счёт гранта Российского научного фонда (проект № 22-21-20070, rscf.ru/project/22-21-20070), а также поддержано грантом Санкт-Петербургского научного фонда (соглашение № 65/2022).

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Garcia-Ramon M. D., Ortiz A., Prats M.** Urban planning, gender and the use of public space in a peripheral neighbourhood of Barcelona // *Cities*. 2004. V. 21, No. 3. P. 215–223. DOI: 10.1016/j.cities.2004.03.006.

2. **Amin A.** Collective culture and urban public space // *City*. 2008. V. 12, No. 1. P. 5–24. DOI: 10.1080/13604810801933495.
3. **Gaffikin F., Mceldowney M., Sterrett K.** Creating shared public space in the contested city: The role of urban design // *J. Urban Des.* 2010. V. 15, No. 4. P. 493–513. DOI: 10.1080/13574809.2010.502338.
4. **Bergantinos G., Lorenzo L., Lorenzo-Freire S.** A characterization of the proportional rule in multi-issue allocation situations // *Oper. Res. Lett.* 2010. V. 38, No. 1. P. 17–19.
5. **Csóka P., Herings P. J. J.** An axiomatization of the proportional rule in financial networks // *Manag. Sci.* 2021. V. 67, No. 5. P. 2799–2812.
6. **Béal S., Casajus A., Huettner F., Rémila E., Solal P.** Characterizations of weighted and equal division values // *Theory Decis.* 2016. V. 80, No. 4. P. 649–667.
7. **Casajus A., Huettner F.** Null, nullifying, or dummifying players: The difference between the Shapley value, the equal division value, and the equal surplus division value // *Econ. Lett.* 2014. V. 122, No. 2. P. 167–169.
8. **Moulin H.** Equal or proportional division of a surplus, and other methods // *Int. J. Game Theory.* 1987. V. 16. P. 161–186.
9. **Van den Brink R.** Null or nullifying players: the difference between the Shapley value and equal division solutions // *J. Econ. Theory.* 2007. V. 136, No. 1. P. 767–775.
10. **Moulin H.** Equal or proportional division of a surplus, and other methods // *Int. J. Game Theory.* 1987. V. 16. P. 161–186.
11. **Van den Brink R., Funaki Y.** Axiomatizations of a class of equal surplus sharing solutions for TU-games // *Theory Decis.* 2009. V. 67. P. 303–340.
12. **Winter E.** The Shapley value // *Handbook of game theory with economic applications*. V. 3. Amsterdam: North Holland, 2002. P. 2025–2054.
13. **Hart S.** Shapley value // *Game theory*. London: Macmillan Press, 1989. P. 210–216. DOI: 10.1007/978-1-349-20181-5_25.

Гусев Василий Васильевич

Статья поступила

8 ноября 2023 г.

После доработки —

4 декабря 2023 г.

Принята к публикации

22 декабря 2023 г.

COOPERATIVE GAMES WITH PREFERENCES:
APPLICATION OF THE WEIGHT RULE TO PROBLEMS
OF PUBLIC SPACE IN ST. PETERSBURG

V. V. Gusev

HSE University,
3 Kantemirovskaya Street, 194100 St. Petersburg, Russian Federation
E-mail: vgusev@hse.ru

Abstract. The paper examines the problem of the distribution of public space. We use the methods of cooperative game theory to solve this problem. Players are districts, while the value of the characteristic function is the total number of people interested in a particular type of public space in the areas under consideration. The axioms that are characteristic of the problem of division are compiled. A special value of the cooperative game is derived that depends on the weights of the players. It is shown how to choose the weights by optimization methods. Tab. 1, illustr. 2, bibliogr. 13.

Keywords: game theory, cooperative game, weight rule, preference game.

References

1. M. D. Garcia-Ramon, A. Ortiz, and M. Prats, Urban planning, gender and the use of public space in a peripheral neighbourhood of Barcelona, *Cities* **21** (3), 215–223 (2004), DOI: 10.1016/j.cities.2004.03.006.
2. A. Amin, Collective culture and urban public space, *City* **12** (1), 5–24 (2008), DOI: 10.1080/13604810801933495.
3. F. Gaffikin, M. Mceldowney, and K. Sterrett, Creating shared public space in the contested city: The role of urban design, *J. Urban Des.* **15** (4), 493–513 (2010), DOI: 10.1080/13574809.2010.502338.
4. G. Bergantinos, L. Lorenzo, and S. Lorenzo-Freire, A characterization of the proportional rule in multi-issue allocation situations, *Oper. Res. Lett.* **38** (1), 17–19 (2010).

English transl.: *Journal of Applied and Industrial Mathematics* **18** (2), 271–281 (2024), DOI: 10.1134/S1990478924020091.

5. **P. Csóka** and **P. J. J. Herings**, An axiomatization of the proportional rule in financial networks, *Manag. Sci.* **67** (5), 2799–2812 (2021).
6. **S. Béal**, **A. Casajus**, **F. Huettner**, **E. Rémila**, and **P. Solal**, Characterizations of weighted and equal division values, *Theory Decis.* **80** (4), 649–667 (2016).
7. **A. Casajus** and **F. Huettner**, Null, nullifying, or dummifying players: The difference between the Shapley value, the equal division value, and the equal surplus division value, *Econ. Lett.* **122** (2), 167–169 (2014).
8. **H. Moulin**, Equal or proportional division of a surplus, and other methods, *Int. J. Game Theory* **16**, 161–186 (1987).
9. **R. van den Brink**, Null or nullifying players: the difference between the Shapley value and equal division solutions, *J. Econ. Theory* **136** (1), 767–775 (2007).
10. **H. Moulin**, Equal or proportional division of a surplus, and other methods, *Int. J. Game Theory* **16**, 161–186 (1987).
11. **R. van den Brink** and **Y. Funaki**, Axiomatizations of a class of equal surplus sharing solutions for TU-games, *Theory Decis.* **67**, 303–340 (2009).
12. **E. Winter**, The Shapley value, in *Handbook of Game Theory with Economic Applications*, Vol. 3 (North Holland, Amsterdam, 2002), pp. 2025–2054.
13. **S. Hart**, Shapley value, in *Game Theory* (Macmillan Press, London, 1989), pp. 210–216, DOI: 10.1007/978-1-349-20181-5_25.

Vasily V. Gusev

Received November 8, 2023

Revised December 4, 2023

Accepted December 22, 2023

ПЕРЕЧИСЛЕНИЕ ЧЁТНЫХ И НЕЧЁТНЫХ ХОРДОВЫХ ДИАГРАММ

Д. Б. Ефимов

Физико-математический институт ФИЦ Коми НЦ УрО РАН,
ул. Коммунистическая, 24, 167982 Сыктывкар, Россия
E-mail: defimov@ipm.komisc.ru

Аннотация. Рассматривается общий метод перечисления различных классов хордовых диаграмм с чётным и нечётным числом пересечений хорд. В основе метода лежит вычисление пфаффиана и гафниана матрицы ограничений, характеризующей класс диаграмм. Табл. 3, ил. 6, библиогр. 23.

Ключевые слова: хордовая диаграмма, пфаффиан, гафниан.

Введение

Пусть n — чётное натуральное число. *Хордовой диаграммой* на n вершинах называется простой граф, все вершины которого расположены на окружности, пронумерованы последовательно числами от 1 до n по часовой стрелке, и каждая вершина соединена ребром (хордой окружности) ровно с одной другой вершиной (рис. 1). Наряду с хордовой диаграммой также часто рассматривают её линейный эквивалент, располагая вершины не вдоль окружности, а вдоль отрезка горизонтальной прямой, и изображая рёбра в виде дуг. При этом говорят о линейной хордовой диаграмме.

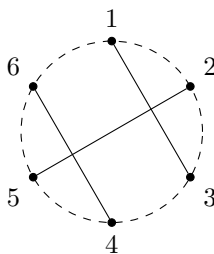


Рис. 1. Хордовая диаграмма на шести вершинах

С хордовыми диаграммами связано много интересных теоретико-вероятностных и комбинаторных задач [1–5]. Они также выступают в качестве удобной содержательной модели в задачах других разделов математики (топологии, теории представлений), а также в задачах других научных дисциплин (например биологии). Довольно подробный обзор применения хордовых диаграмм можно найти в [6].

Один из аспектов изучения хордовых диаграмм связан с комбинаторикой и статистикой числа пересечений их рёбер. Впервые задача перечисления хордовых диаграмм с заданным числом хорд и с заданным числом пересечений хорд была рассмотрена в работах Тушара [7] и Риордана [8]. В последнее время появился ряд работ, посвящённых изучению хордовых диаграмм с точки зрения их графов пересечений (chord intersection graph) [6, 9, 10]. Вершины такого графа соответствуют хордам диаграммы, а роль ребра играет факт пересечения хорд в диаграмме, т. е. две вершины соединяются ребром в том и только том случае, когда соответствующие им хорды пересекаются.

Настоящая работа также посвящена комбинаторике пересечений рёбер хордовых диаграмм, а именно рассматривается вопрос перечисления чётных и нечётных хордовых диаграмм, т. е. хордовых диаграмм с чётным и нечётным числом пересечений рёбер. В разд. 1 приводится общий метод перечисления чётных и нечётных хордовых диаграмм с заданными свойствами, основанный на комбинации таких функций от матриц, как пфаффиан и гафниан. В разд. 2 доказываются некоторые общие утверждения, а полученные результаты применяются для перечисления чётных и нечётных хордовых диаграмм некоторых конкретных классов.

1. Описание метода перечисления

Рассмотрим сначала общий метод перечисления чётных и нечётных перестановок с ограниченными позициями, основанный на комбинации таких функций от матриц, как определитель и перманент [11, 12]. Напомним, что *перманент* матрицы $A = (a_{ij})$ порядка n определяется следующим образом [13]:

$$\text{per } A = \sum_{\sigma \in S_n} \prod_{i=1}^n a_{i, \sigma(i)},$$

где S_n — множество всех перестановок порядка n . Таким образом, перманент матрицы в отличие от её определителя игнорирует знаки перестановок, по которым идёт суммирование. Предположим, что A — это $(0, 1)$ -матрица. Тогда она задаёт некоторый класс $\mathcal{P}_A$ (ограниченных) перестановок порядка n : в этот класс попадают те и только те перестановки, матрицы которых не превосходят поэлементно матрицу A . Обозначим

через $\mathcal{EP}_A$ и $\mathcal{OP}_A$ множество чётных и нечётных перестановок из $\mathcal{P}_A$ соответственно. Нетрудно видеть, что перманент матрицы A равен общему числу перестановок в классе $\mathcal{P}_A$, а определитель матрицы A равен разности между числом чётных и нечётных перестановок в $\mathcal{P}_A$:

$$\text{per } A = |\mathcal{P}_A|, \quad \det A = |\mathcal{EP}_A| - |\mathcal{OP}_A|.$$

Отсюда следует, что число чётных и нечётных перестановок в классе $\mathcal{P}_A$ можно определить с помощью следующих формул:

$$|\mathcal{EP}_A| = \frac{\text{per } A + \det A}{2}, \quad |\mathcal{OP}_A| = \frac{\text{per } A - \det A}{2}. \quad (1)$$

Введём понятие *двудольной хордовой диаграммы*. Под такой диаграммой будем понимать простой двудольный граф на $2n$ вершинах степени 1, расположенных на окружности, причём вершины $u_1, u_2, \dots, u_n$ первой доли расположены в правой части окружности по часовой стрелке, а вершины $v_1, v_2, \dots, v_n$ — в левой части против часовой стрелки. Назовём такую диаграмму *чётной*, если её рёбра имеют чётное число взаимных пересечений, и *нечётной* в противном случае (рис. 2).

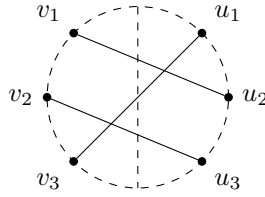


Рис. 2. Чётная двудольная хордовая диаграмма на шести вершинах

Отметим, что любая $(0,1)$ -матрица A порядка n однозначно задаёт семейство $\mathcal{BD}_A$ двудольных хордовых диаграмм на $2n$ вершинах, в которое входят те и только те диаграммы, матрица бисмежности которых не превосходит поэлементно A . Любой перестановке из класса $\mathcal{P}_A$ с матрицей M можно однозначно поставить в соответствие двудольную хордовую диаграмму из $\mathcal{BD}_A$ с матрицей бисмежности M , при этом чётности перестановки и диаграммы будут совпадать. Отсюда следует, что формулы (1) можно также применять для перечисления чётных и нечётных двудольных хордовых диаграмм в заданных классах. Отметим также, что если рассматривать A как матрицу бисмежности двудольного графа, то каждой диаграмме из $\mathcal{BD}_A$ будет однозначно соответствовать некоторое совершенное паросочетание данного графа.

Аналогичную роль в задачах перечисления хордовых диаграмм может играть комбинация таких функций от матриц как пфаффиан и гаффиан. Сделаем сначала несколько предварительных замечаний. Вполне

очевидно, что существует взаимно однозначное соответствие между множеством хордовых диаграмм на $n = 2m$ вершинах и множеством неупорядоченных разбиений множества $I = \{1, 2, \dots, n\}$ на непересекающиеся неупорядоченные пары $\{(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)\}$. Паре (a_i, b_i) однозначно соответствует ребро хордовой диаграммы, соединяющее вершины с номерами a_i и b_i . В дальнейшем не будем различать эти два множества и обозначим их через $\mathcal{D}_n$. В зависимости от контекста под данным обозначением будем понимать ту или иную интерпретацию. Также условимся, что при записи ребра в виде пары чисел на первом месте будет стоять меньшее число. Пусть (i, j) и (k, l) — два ребра хордовой диаграммы. Нетрудно видеть, что эти два ребра геометрически пересекаются тогда и только тогда, когда выполняется одно из двух неравенств: $i < k < j < l$ или $k < i < l < j$. Знаком $\operatorname{sgn} \alpha$ хордовой диаграммы α назовём число $(-1)^p$, где p — число пересечений рёбер из α . Будем говорить, что хордовая диаграмма α *чётная* (*нечётная*), если $\operatorname{sgn} \alpha = 1$ ($\operatorname{sgn} \alpha = -1$), т. е. если α имеет чётное (нечётное) число пересечений рёбер. Множество всех чётных хордовых диаграмм на n вершинах обозначим через $\mathcal{ED}_n$, а нечётных — через $\mathcal{OD}_n$.

Пфаффиан кососимметричной матрицы $A = (a_{ij})$ чётного порядка n можно определить следующим образом [14]:

$$\operatorname{Pf} A = \sum_{\alpha \in \mathcal{D}_n} \operatorname{sgn} \alpha \prod_{(i,j) \in \alpha} a_{ij}.$$

Если не учитывать знаки хордовых диаграмм и вместо кососимметричной матрицы взять симметричную, получим определение *гафниана* [15, гл. 4]:

$$\operatorname{Hf} A = \sum_{\alpha \in \mathcal{D}_n} \prod_{(i,j) \in \alpha} a_{ij}.$$

Произвольная симметричная $(0, 1)$ -матрица A чётного порядка с нулевой главной диагональю задаёт некоторое семейство $\mathcal{D}_A$ хордовых диаграмм: семейство $\mathcal{D}_A$ состоит из хордовых диаграмм, матрицы смежности которых поэлементно не превосходят A . Если рассматривать A как матрицу смежности некоторого графа, то любой хордовой диаграмме из $\mathcal{D}_A$ с матрицей смежности M однозначно соответствует совершенное паросочетание данного графа с такой же матрицей смежности. Имея в виду это соответствие, будем иногда говорить о хордовых диаграммах графа, подразумевая соответствующие совершенные паросочетания. Нетрудно заметить, что гафниан матрицы A равен общему числу диаграмм в семействе $\mathcal{D}_A$. Действительно, при вычислении гафниана суммирование идёт по всевозможным хордовым диаграммам, но ненулевые слагаемые (каждое из которых равно 1) возникают на тех и только

тех диаграммах, все рёбра которых соответствуют ненулевым элементам матрицы A .

Обозначим через A' кососимметричную матрицу, которая получается из матрицы A заменой всех ненулевых элементов ниже главной диагонали на -1 . Если вычислить пфаффиан такой матрицы, то каждая чётная хордовая диаграмма из семейства $\mathcal{D}_A$ будет вносить вклад 1 в общую сумму, а нечётная диаграмма — вклад -1 . Таким образом, пфаффиан такой матрицы будет равен разности между числом чётных и нечётных хордовых диаграмм из $\mathcal{D}_A$.

Пусть A — симметричная $(0, 1)$ -матрица чётного порядка. Обозначим через $\mathcal{ED}_A$ и $\mathcal{OD}_A$ множества чётных и нечётных хордовых диаграмм из семейства $\mathcal{D}_A$ соответственно. Из вышесказанного следует, что

$$\text{Hf } A = |\mathcal{D}_A| = |\mathcal{ED}_A| + |\mathcal{OD}_A|, \quad \text{Pf } A' = |\mathcal{ED}_A| - |\mathcal{OD}_A|.$$

Отсюда

$$|\mathcal{ED}_A| = \frac{\text{Hf } A + \text{Pf } A'}{2}, \quad |\mathcal{OD}_A| = \frac{\text{Hf } A - \text{Pf } A'}{2}. \quad (2)$$

В терминах графов пересечений формулы (2) позволяют найти общее число графов пересечений данного набора хордовых диаграмм с чётным (нечётным) числом рёбер.

Пример 1. Рассмотрим матрицу

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

Единственный её нулевой элемент выше главной диагонали находится на позиции $(1, 3)$, поэтому данной матрице соответствует семейство хордовых диаграмм на четырёх вершинах, в которых нет ребра $(1, 3)$. Нетрудно подсчитать, что $\text{Hf } A = \text{Pf } A' = 2$. Это согласуется с тем, что семейство $\mathcal{D}_A$ состоит из двух диаграмм, каждая из которых имеет нулевое число пересечений рёбер (рис. 3).

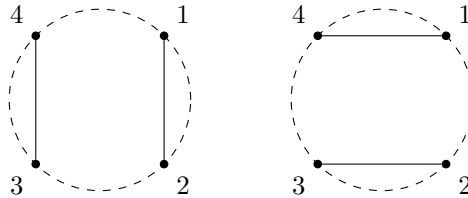


Рис. 3. Семейство хордовых диаграмм на четырёх вершинах, в которых нет ребра $(1, 3)$

Отметим, что чётным и нечётным хордовым диаграммам можно дать интерпретацию на языке диаграмм Брауэра [16–18], что является обобщением интерпретации двудольных хордовых диаграмм через перестановки.

Чтобы применять формулы (2), необходимо, как бы банально это ни звучало, уметь вычислять пфаффиан и гафниан. Наибольшим камнем преткновения в этой связке является гафниан. Сложность его вычисления заведомо не меньше сложности вычисления перманента, а последняя задача, как известно, принадлежит как минимум к классу NP-полных задач. Тем не менее, существует ряд в той или иной степени эффективных алгоритмов для вычисления гафниана специальных типов матриц (см., например, [19]). Особо отметим изящный метод эффективного вычисления гафниана матриц смежности планарных графов, разработанный Кастелейном в 1960-х гг. в рамках решения задач статистической физики (см., например, [20, гл. 8]). К этому методу ещё вернёмся в разд. 2. Существуют также алгоритмы для эффективного вычисления пфаффиана [21, 22].

В данной работе нас больше интересует не алгоритмический, а аналитический подход. В разд. 2 докажем несколько несложных, но полезных утверждений, характеризующих число чётных и нечётных хордовых диаграмм в различных классах диаграмм, а также рассмотрим некоторые примеры, когда на основе общих свойств пфаффиана и гафниана можно получить точные аналитические формулы, выражающие число чётных и нечётных хордовых диаграмм в классе диаграмм (в графе).

2. Утверждения и примеры

2.1. Свойства пфаффиана и гафниана. Прежде чем переходить к доказательству утверждений и рассмотрению примеров, перечислим некоторые известные свойства пфаффиана и гафниана, используемые в дальнейшем.

Пусть дана кососимметричная матрица $A = (a_{ij})$ n -го порядка. Пфаффиан матрицы A обладает следующим свойством разложения по i -й строке:

$$\text{Pf } A = \sum_{j=1, j \neq i}^n (-1)^{i+j+1+\theta(i-j)} a_{ij} \text{Pf } A(i, j),$$

где $A(i, j)$ — матрица, которая получается из матрицы A удалением строк и столбцов с номерами i и j , θ — функция Хевисайда. Применительно к первой строке данная формула приобретает более простой вид:

$$\text{Pf } A = \sum_{j=2}^n (-1)^j a_{1j} \text{Pf } A(1, j). \quad (3)$$

Пфаффиан кососимметричной матрицы тесно связан с её определителем, а именно, справедливо следующее соотношение:

$$|\operatorname{Pf} A| = \sqrt{\det A}. \quad (4)$$

Пусть B — произвольная матрица порядка n . Имеет место следующее свойство:

$$\operatorname{Pf}(BAB^T) = \det(B) \operatorname{Pf}(A). \quad (5)$$

В частности, отсюда следует, что пфаффиан матрицы не меняется при таких элементарных преобразованиях, как перестановка строк совместно с аналогичной перестановкой столбцов, а также прибавление к одной строке другой, умноженной на произвольное число, соединённое с аналогичной операцией над столбцами с такими же номерами.

Пусть A и B — две кососимметричные матрицы чётного порядка n . Тогда справедлива следующая формула, выражающая пфаффиан суммы матриц через сумму произведений пфаффианов подматриц [14]:

$$\operatorname{Pf}(A + B) = \sum_J (-1)^{|J|/2} (-1)^{s(J)} \operatorname{Pf}(A[J]) \operatorname{Pf}(B(J)), \quad (6)$$

где суммирование ведётся по всем подмножествам чётной мощности $J \subseteq \{1, 2, \dots, n\}$, $s(J) = \sum_{j \in J} j$ (при этом $s(\emptyset) = 0$), $A[J]$ — матрица, полученная из матрицы A выбором строк и столбцов с номерами из J , $B(J)$ — матрица, полученная из матрицы B вычёркиванием строк и столбцов с номерами из J . По определению считаем, что пфаффиан (0×0) -матрицы равен 1.

Для гафниана справедливы аналогичные формулы, причём, выглядят они проще за счёт того, что гафниан не учитывает знаки хордовых диаграмм, по которым идёт суммирование. При этом «платой» за простоту является значительное сужение диапазона применения аналогов формул (4) и (5) по сравнению со случаем пфаффиана. Имеет место следующий аналог формулы (6) для гафниана суммы симметричных матриц:

$$\operatorname{Hf}(A + B) = \sum_J \operatorname{Hf}(A[J]) \operatorname{Hf}(B(J)). \quad (7)$$

2.2. Хордовые диаграммы полного графа. Рассмотрим множество всех хордовых диаграмм с $n = 2m$ вершинами. Напомним, что мы обозначили его через $\mathcal{D}_n$. Мощность данного множества равна общему

числу совершенных паросочетаний полного графа с n вершинами, а также равна гаффиану матрицы A n -го порядка с нулевой главной диагональю и единичными элементами вне главной диагонали:

$$A = \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix},$$

$$|\mathcal{D}_n| = \text{Hf } A = \frac{(2m)!}{m!2^m} = (n-1)!!.$$

Утверждение 1. В полном графе на $2m$ вершинах разница между числом чётных и нечётных хордовых диаграмм равна 1.

ДОКАЗАТЕЛЬСТВО. Рассмотрим кососимметричную матрицу A' , которая получается из матрицы A умножением всех элементов ниже главной диагонали на -1 . Разница между числом чётных и нечётных хордовых диаграмм в полном графе равна пфаффиану данной матрицы. Обозначим пфаффиан матрицы A' порядка $2m$ через π_m . Раскладывая пфаффиан матрицы A' по первой строке, получаем рекуррентное соотношение $\pi_m = \pi_{m-1}$. Так как $\pi_1 = 1$, то $\pi_m = 1$ для любого $m \geq 1$. Утверждение 1 доказано.

Таким образом, с учётом (2) число чётных и нечётных хордовых диаграмм полного графа на n вершинах (n — чётное) равно соответственно

$$|\mathcal{ED}_n| = \frac{(n-1)!! + 1}{2}, \quad |\mathcal{OD}_n| = \frac{(n-1)!! - 1}{2}.$$

Тем самым в обозначениях работы [8] получены суммы коэффициентов $T_{n,k}$ для чётных и нечётных k при фиксированном n :

$$|\mathcal{ED}_n| = \sum_{i=0} T_{n,2i}, \quad |\mathcal{OD}_n| = \sum_{i=0} T_{n,2i+1}.$$

Если вместо n поочерёдно подставить чётные числа, то получим последовательности, представленные в табл. 1.

Таблица 1

m	1	2	3	4	5	6	7	8	...
$ \mathcal{ED}_{2m} $	1	2	8	53	473	5198	67568	1013513	...
$ \mathcal{OD}_{2m} $	0	1	7	52	472	5197	67567	1013512	...

Отметим, что в [23] эти последовательности имеют номера A193651 и A249349, но там не указана приведённая здесь интерпретация.

2.3. Беспетлевые хордовые диаграммы. Рассмотрим теперь хордовые диаграммы, в которых нет рёбер, соединяющих соседние вершины (иногда их называют беспетлевыми [2]). Класс таких диаграмм на $n = 2m$ вершинах задаётся симметричной $(0, 1)$ -матрицей A порядка n с нулями на главной диагонали, на примыкающих к ней диагоналях, а также на позициях $(1, n)$ и $(n, 1)$. Такая матрица также является матрицей смежности полного графа на n вершинах без гамильтонова цикла $1 - 2 - 3 - \dots - n - 1$. Например, при $n = 6$ матрица A будет иметь следующий вид:

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

Утверждение 2. Разность между числом чётных и нечётных беспетлевых хордовых диаграмм на $2m$, $m \geq 2$, вершинах равна 2, если $m \equiv 0 \pmod{3}$, и -1 в противном случае.

ДОКАЗАТЕЛЬСТВО. Рассмотрим кососимметричную матрицу A' порядка $2m$, которая получается из матрицы A умножением всех элементов ниже главной диагонали на -1 . Пусть $\pi_m = \text{Pf } A'$. Разница между числом чётных и нечётных беспетлевых хордовых диаграмм на $2m$ вершинах равна π_m . Чтобы вычислить π_m , введём вспомогательную матрицу A'' порядка $2m$, которая отличается от матрицы A' тем, что $a''_{1,2m} = a''_{2m,1} = 1$. Положим $\rho_m = \text{Pf } A''$. Учитывая правило (3) разложения пфаффиана по первой строке, нетрудно видеть, что пфаффианы матриц A' и A'' связаны друг с другом следующим соотношением:

$$\pi_m = \rho_m - \rho_{m-1}, \quad m \geq 2. \quad (8)$$

Рассмотрим матрицу A'' порядка $2m$. Вычтем в ней из первой строки вторую строку и из первого столбца второй столбец, а затем из второй строки третью строку и из второго столбца третий столбец. В силу свойства (5) пфаффиан матрицы при этом не изменится. Раскладывая затем пфаффиан получившейся матрицы по первой строке, получаем рекуррентное соотношение $\rho_m = -\rho_{m-1} - \rho_{m-2}$, $m \geq 2$. Тогда с учётом (8) для $m \geq 4$ получаем

$$\begin{aligned} \pi_m &= \rho_m - \rho_{m-1} = -\rho_{m-1} - \rho_{m-2} - (-\rho_{m-2} - \rho_{m-3}) = \\ &= -(\rho_{m-1} - \rho_{m-2}) - (\rho_{m-2} - \rho_{m-3}) = -\pi_{m-1} - \pi_{m-2}. \end{aligned}$$

Нетрудно непосредственно вычислить, что $\pi_2 = -1$, $\pi_3 = 2$. Утверждение 2 доказано.

Замечание. Если представить матрицу A' порядка $2m$ в виде суммы двух кососимметричных матриц, в одной из которых все элементы выше главной диагонали равны 1, и применить формулу (6), выражающую пфаффиан суммы матриц через сумму произведений пфаффианов подматриц, то при $m \geq 2$ получим

$$\text{Pf } A' = \pi_m = \sum_{k=0}^m (-1)^{m-k} (C_{m+k}^{m-k} + C_{m+k-1}^{m-k-1}). \quad (9)$$

Общее число беспетлевых хордовых диаграмм на $2m$, $m \geq 2$, вершинах равно гафниану матрицы A порядка $2m$. Чтобы вычислить данную величину, можно опять представить матрицу A в виде суммы двух симметричных матриц, одна из которых полностью состоит из единиц за исключением главной диагонали, и воспользоваться формулой (7). В результате имеем

$$\text{Hf } A = \sum_{k=0}^m (-1)^{m-k} (C_{m+k}^{m-k} + C_{m+k-1}^{m-k-1}) (2k-1)!!. \quad (10)$$

Подставляя (9), (10) в (2), получаем, что общие числа чётных и нечётных беспетлевых хордовых диаграмм на $2m$, $m \geq 2$, вершинах можно выразить следующими суммами:

$$|\mathcal{ED}_A| = \sum_{k=0}^m (-1)^{m-k} (C_{m+k}^{m-k} + C_{m+k-1}^{m-k-1}) \frac{(2k-1)!! + 1}{2},$$

$$|\mathcal{OD}_A| = \sum_{k=0}^m (-1)^{m-k} (C_{m+k}^{m-k} + C_{m+k-1}^{m-k-1}) \frac{(2k-1)!! - 1}{2}.$$

Случай $m = 1$ стоит особняком: здесь очевидно, что $|\mathcal{ED}_A| = |\mathcal{OD}_A| = 0$. Отметим, что данные формулы можно также получить с помощью метода включений-исключений, используя результаты п. 2.2. Подставляя поочерёдно в эти формулы значения m , получаем целочисленные последовательности, представленные в табл. 2.

Таблица 2

m	1	2	3	4	5	6	7	8	...
$ \mathcal{ED}_A $	0	0	3	15	146	1663	22 094	336 735	...
$ \mathcal{OD}_A $	0	1	1	16	147	1663	22 095	336 736	...

На рис. 4 представлены соответствующие диаграммы для $m = 3$. Первые три имеют чётное число пересечений рёбер, последняя — нечётное.

Рассмотрим далее два «пограничных» случая, когда число чётных и нечётных хордовых диаграмм в графе совпадает и когда в графе есть только диаграммы одного типа.

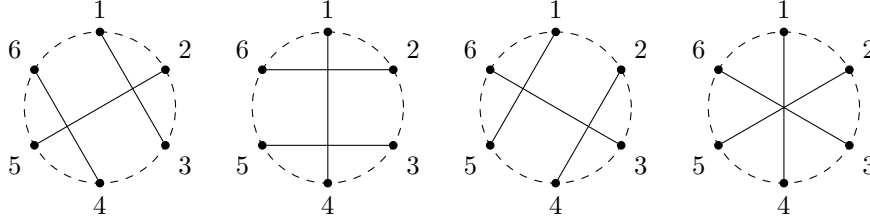


Рис. 4. Беспетлевые хордовые диаграммы на шести вершинах

2.4. Критерий равенства числа чётных и нечётных хордовых диаграмм. В силу (2) и (4) получаем следующий простой критерий равенства числа чётных и нечётных хордовых диаграмм в рассматриваемом классе хордовых диаграмм (в графе).

Утверждение 3. Класс хордовых диаграмм, задаваемый матрицей A (граф с матрицей смежности A), содержит одинаковое число чётных и нечётных хордовых диаграмм тогда и только тогда, когда $\det A' = 0$.

Напомним, что *беспорядком* (англ. derangement) называется перестановка, при которой ни один элемент не остаётся на своём месте (перестановка без неподвижных точек). В качестве диаграммного аналога таких перестановок рассмотрим хордовые диаграммы, в которых нет рёбер вида $(2i - 1, 2i)$. Назовём такие диаграммы *хордовыми беспорядками*. Множество таких диаграмм на $n = 2m$ вершинах задаётся симметричной $(0, 1)$ -матрицей A порядка n с нулевыми элементами на главной диагонали и на позициях $(2i - 1, 2i)$, $i = 1, 2, \dots, m$. Данная матрица является также матрицей смежности полного графа на n вершинах без совершенного паросочетания $\{(1, 2), (3, 4), \dots, (n - 1, n)\}$. Например, при $n = 6$ такая матрица имеет вид

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

Очевидно, что при любом n определитель соответствующей кососимметричной матрицы A' равен нулю. Отсюда в силу утверждения 3 следует, что для фиксированного n число чётных и нечётных хордовых беспорядков совпадает. Для того чтобы вычислить общее число хордовых беспорядков для фиксированного n , можно представить матрицу A в виде суммы двух симметричных матриц, одна из которых за исключением главной диагонали полностью состоит из единиц, и воспользоваться

формулой (7), выражающей гафниан суммы матриц через сумму произведений гафнианов подматриц, или же применить метод включений-исключений. В результате получим

$$|\mathcal{D}_A| = \sum_{k=0}^m (-1)^{m-k} C_m^k (2k-1)!!,$$

откуда в силу вышесказанного

$$|\mathcal{ED}_A| = |\mathcal{OD}_A| = \frac{1}{2} \sum_{k=0}^m (-1)^{m-k} C_m^k (2k-1)!!.$$

Подставляя поочерёдно вместо m натуральные числа, получаем последовательность, представленную в табл. 3.

Таблица 3

m	1	2	3	4	5	6	7	8	...
$ \mathcal{ED}_A = \mathcal{OD}_A $	0	1	4	30	272	3020	39 504	595 336	...

Эта последовательность представлена в [23] под номером A179540, но имеет весьма скудное описание без приведённой здесь интерпретации.

Пример 2. Рассмотрим хордовые беспорядки на шести вершинах. Всего таких диаграмм 8, из них 4 с чётным числом пересечений хорд и 4 — с нечётным (рис. 5).

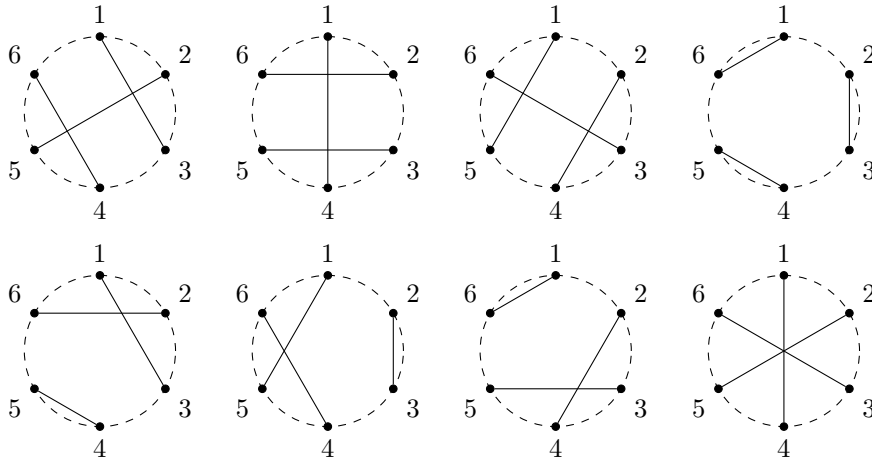


Рис. 5. Хордовые беспорядки на шести вершинах

2.5. Хордовые диаграммы пфаффовых графов. В 1960-х гг. Кастелейн разработал метод, позволяющий вычислять число совершенных паросочетаний (хордовых диаграмм) графа с помощью определителя его матрицы смежности. Суть метода заключается в том, чтобы построить на графе так называемую *пфаффову ориентацию*. При такой ориентации все слагаемые пфаффиана матрицы смежности графа имеют одинаковый знак и модуль пфаффиана матрицы смежности, или в силу свойства (4) квадратный корень её определителя, равны общему числу совершенных паросочетаний графа. Граф, на котором можно построить пфаффову ориентацию, называется *пфаффовым*. Известно [20, с. 411], что если граф не содержит подразбиения полного двудольного графа $K_{3,3}$, то он пфаффов. В частности, в эту категорию попадают планарные графы. Для того чтобы ориентация связного планарного графа была пфаффовой, достаточно, чтобы граница каждой внутренней грани соответствующего плоского графа содержала нечётное число рёбер, ориентированных по часовой стрелке.

Очевидно, что общее число хордовых диаграмм графа не зависит от нумерации его вершин, но число его чётных и нечётных хордовых диаграмм, вообще говоря, от нумерации вершин зависит. В данном контексте следующее утверждение характеризует один из специальных случаев нумерации вершин пфаффовых графов.

Утверждение 4. Пусть на графе задана пфаффова ориентация так, что у него нет ориентированных циклов. Тогда вершины графа можно занумеровать таким образом, чтобы все его хордовые диаграммы имели одинаковую чётность.

ДОКАЗАТЕЛЬСТВО. Занумеруем вершины графа так, чтобы каждое ребро было направлено от вершины с меньшим номером к вершине с большим. Покажем, что это всегда можно сделать. Так как в графе нет ориентированных циклов, в нём есть хотя бы один сток. Присвоим этой вершине-стоку наибольший номер. Далее удалим из графа эту вершину и смежные с ней рёбра. Получим новый граф, в котором так же нет ориентированных циклов и есть сток. Найденному стоку присвоим следующий по убыванию номер. Продолжая аналогично, занумеруем требуемым образом все вершины.

При такой нумерации матрица смежности данного графа выше главной диагонали будет содержать неотрицательные элементы, равные 0 или 1. Как известно, пфаффиан такой матрицы равен разности чисел чётных и нечётных хордовых диаграмм. С другой стороны, поскольку по условию заданная ориентация пфаффова, пфаффиан такой матрицы по модулю равен общему числу хордовых диаграмм графа. Утверждение 4 доказано.

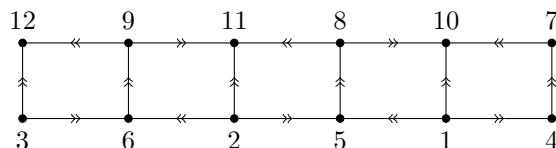


Рис. 6. Прямоугольная решётка с пфаффовской ориентацией без ориентированных циклов

Пример 3. Рассмотрим прямоугольную решётку на рис. 6. Нетрудно видеть, что на ней задана пфаффовская ориентация. Так как все вертикальные рёбра ориентированы вверх, у неё нет ориентированных циклов. Следовательно, вершины можно занумеровать так, чтобы каждое ребро было направлено от вершины с меньшим номером к вершине с большим. Одна из таких нумераций представлена на рис. 6. Легко проверить непосредственно, что, например, хордовая диаграмма, соответствующая множеству вертикальных рёбер, чётная. Отсюда в силу утверждения 4 следует, что все хордовые диаграммы данной решётки будут чётными.

Финансирование работы

Исследование выполнено в рамках плановой гос. бюджетной темы НИР Физико-математического института ФИЦ Коми НЦ УрО РАН «Математические проблемы теории стохастических и детерминированных систем, включая системы большой размерности» (номер гос. регистрации 122040600066–5). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Acan H.** An enumerative-probabilistic study of chord diagrams: PhD Thes. Columbus: Ohio State Univ., 2013. 144 p.
2. **Краско Е. С., Лабутин И. Н., Омельченко А. В.** Перечисление помеченных и непомеченных гамильтоновых циклов в полных k -дольных графах // Зап. науч. сем. ПОМИ. 2019. Т. 488. С. 119–142.
3. **Nakamigawa T.** The expansion of a chord diagram and the Genocchi numbers // Ars Math. Contemp. 2020. V. 18. P. 381–391.
4. **Sullivan E.** Linear chord diagrams with long chords // Electron. J. Comb. 2017. V. 24, No. 4. Paper ID P4.20. 8 p.
5. **Cameron N. T., Killpatrick K.** Statistics on linear chord diagrams // Discrete Math. Theor. Comput. Sci. 2020. V. 21, No. 2. Paper ID 11. 10 p.
6. **Acan H.** On a uniformly random chord diagram and its intersection graph // Discrete Math. 2017. V. 340, No. 8. P. 1967–1985.

7. **Touchard J.** Sur un problème de configurations et sur les fractions continues // *Canad. J. Math.* 1952. V. 4. P. 2–25. [French].
8. **Riordan J.** The distribution of crossing of chords joining pairs of $2n$ points on a circle // *Math. Comput.* 1975. V. 29, No. 129. P. 215–222.
9. **Courtiell J., Yeats K., Zeilberger N.** Connected chord diagrams and bridgeless maps // *Electron. J. Comb.* 2019. V. 26, No. 4. Paper ID P4.37. 56 p.
10. **Mahmoud A. A., Yeats K.** Connected chord diagrams and the combinatorics of asymptotic expansions // *J. Integer Seq.* 2022. V. 25, No. 7. Paper ID 22.7.5. 22 p.
11. **Shevelev V. S.** Combinatorial minors for matrix functions and their applications // *Zesz. Nauk. Politech. Śląsk. Ser. Mat. Stosow.* 2014. No. 4. P. 5–16.
12. **Шевелев В. С.** Некоторые вопросы теории перечисления перестановок с ограниченными позициями // *Итоги науки и техники. Сер. Теория вероятностей. Мат. статистика. Теор. кибернетика. Т. 30. М.: ВИНТИ, 1992. С. 113–177.*
13. **Минк Х.** Перманенты. М.: Мир, 1982. 211 с.
14. **Stembridge J. R.** Nonintersecting paths, Pfaffians, and plane partitions // *Adv. Math.* 1990. V. 83. P. 96–131.
15. **Barvinok A.** Combinatorics and complexity of partition functions. Cham: Springer, 2016. 304 p. (Algorithms Comb.; V. 30).
16. **Fishel S., Grojnowski I.** Canonical bases for the Brauer centralizer algebra // *Math. Res. Lett.* 1995. V. 2. P. 15–26.
17. **Barcelo H., Ram A.** Combinatorial representation theory // *New perspectives in algebraic combinatorics.* New York: Camb. Univ. Press, 1999. P. 23–90. (Math. Sci. Res. Inst. Publ.; V. 38).
18. **Shalile A.** On the center of the Brauer algebra // *Algebr. Represent. Theory.* 2013. V. 16. P. 65–100.
19. **Schwartz M.** Efficiently computing the permanent and Hafnian of some banded Toeplitz matrices // *Linear Algebra Appl.* 2009. V. 430. P. 1364–1374.
20. **Ловас Л., Пламмер М.** Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии. М.: Мир, 1998. 656 с.
21. **Gabiati G., Maffioli F.** On the computation of Pfaffians // *Discrete Appl. Math.* 1994. V. 51. P. 269–275.
22. **Wimmer M.** Algorithm 923: Efficient numerical computation of the Pfaffian for dense and banded skew-symmetric matrices // *ACM Trans. Math. Softw.* 2012. V. 38, No. 4. Paper ID 30. 17 p. DOI: 10.1145/2331130.2331138.
23. The on-line encyclopedia of integer sequences. Highland Park, NJ: OEIS Found., 2024. URL: oeis.org (accessed Mar. 3, 2024).

Ефимов Дмитрий Борисович

Статья поступила
3 апреля 2023 г.
После доработки —
11 октября 2023 г.
Принята к публикации
22 декабря 2023 г.

ENUMERATION OF EVEN AND ODD CHORD DIAGRAMS

D. B. Efimov

Institute of Physics and Mathematics FRC Komi SC UrB RAS,
24 Kommunisticheskaya Street, 167982 Syktyvkar, Russia
E-mail: defimov@ipm.komisc.ru

Abstract. A general method for the enumeration of various classes of chord diagrams with even and odd numbers of chord intersections is considered. The method is based on the calculation of the Pfaffian and Hafnian of the constraint matrix characterizing a class of diagrams. Tab. 3, illustr. 6, bibliogr. 23.

Keywords: chord diagram, Pfaffian, Hafnian.

References

1. **H. Acan**, An enumerative-probabilistic study of chord diagrams, *PhD Thesis* (Ohio State Univ., Columbus, 2013).
2. **E. S. Krasko, I. N. Labutin, and A. V. Omelchenko**, Enumeration of labelled and unlabelled Hamiltonian cycles in complete k -partite graphs, *Zap. Nauchn. Semin. POMI*, **488**, 119–142 (2019) [Russian].
3. **T. Nakamigawa**, The expansion of a chord diagram and the Genocchi numbers, *Ars Math. Contemp.* **18**, 381–391 (2020).
4. **E. Sullivan**, Linear chord diagrams with long chords, *Electron. J. Comb.* **24** (4), Paper ID P4.20 (2017).
5. **N. T. Cameron and K. Killpatrick**, Statistics on linear chord diagrams, *Discrete Math. Theor. Comput. Sci.* **21** (2), Paper ID 11 (2020).
6. **H. Acan**, On a uniformly random chord diagram and its intersection graph, *Discrete Math.* **340** (8), 1967–1985 (2017).
7. **J. Touchard**, Sur un problème de configurations et sur les fractions continues, *Canad. J. Math.* **4**, 2–25 (1952) [French].
8. **J. Riordan**, The distribution of crossing of chords joining pairs of $2n$ points on a circle, *Math. Comput.* **29** (129), 215–222 (1975).
9. **J. Courtiel, K. Yeats, and N. Zeilberger**, Connected chord diagrams and bridgeless maps, *Electron. J. Comb.* **26** (4), Paper ID P4.37 (2019).

English transl.: *Journal of Applied and Industrial Mathematics* **18** (2), 216–226 (2024), DOI: 10.1134/S1990478924020042.

10. **A. A. Mahmoud** and **K. Yeats**, Connected chord diagrams and the combinatorics of asymptotic expansions, *J. Integer Seq.* **25** (7), Paper ID 22.7.5 (2022).
11. **V. S. Shevelev**, Combinatorial minors for matrix functions and their applications, *Zesz. Nauk. Politech. Śląsk. Ser. Mat. Stosow.* No. 4, 5–16 (2014).
12. **V. S. Shevelev**, Some problems of the theory of enumerating the permutations with restricted positions, in *Itogi Nauki Tekhn., Teoriya Veroyatn. Mat. Stat. Teor. Kibern.*, Vol. 30 (VINITI, Moscow, 1992), pp. 113–177 [Russian] [*J. Sov. Math.* **61** (4), 2272–2317 (1992)].
13. **H. Minc**, *Permanents* (Addison-Wesley, Reading, MA, 1978; Mir, Moscow, 1982 [Russian]).
14. **J. R. Stembridge**, Nonintersecting paths, Pfaffians, and plane partitions, *Adv. Math.* **83**, 96–131 (1990).
15. **A. Barvinok**, *Combinatorics and Complexity of Partition Functions* (Springer, Cham, 2016) (Algorithms Comb., Vol. 30).
16. **S. Fishel** and **I. Grojnowski**, Canonical bases for the Brauer centralizer algebra, *Math. Res. Lett.* **2**, 15–26 (1995).
17. **H. Barcelo** and **A. Ram**, Combinatorial representation theory, in *New Perspectives in Algebraic Combinatorics* (Camb. Univ. Press, New York, 1999), pp. 23–90 (Math. Sci. Res. Inst. Publ., Vol. 38).
18. **A. Shalile**, On the center of the Brauer algebra, *Algebr. Represent. Theory* **16**, 65–100 (2013).
19. **M. Schwartz**, Efficiently computing the permanent and Hafnian of some banded Toeplitz matrices, *Linear Algebra Appl.* **430**, 1364–1374 (2009).
20. **L. Lovász** and **M. D. Plummer**, *Matching Theory* (North-Holland, Amsterdam, 1986; Mir, Moscow, 1998 [Russian]).
21. **G. Gabiati** and **F. Maffioli**, On the computation of Pfaffians, *Discrete Appl. Math.* **51**, 269–275 (1994).
22. **M. Wimmer**, Algorithm 923: Efficient numerical computation of the Pfaffian for dense and banded skew-symmetric matrices, *ACM Trans. Math. Softw.* **38** (4), Paper ID 30 (2012), DOI: 10.1145/2331130.2331138.
23. *The On-Line Encyclopedia of Integer Sequences* (OEIS Found., Highland Park, NJ, 2024), URL: oeis.org (accessed Mar. 3, 2024).

Dmitry B. Efimov

Received April 3, 2023

Revised October 11, 2023

Accepted December 22, 2023

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ДЛЯ ОЦЕНКИ ДИНАМИКИ ЧИСЛЕННОСТИ РЫБ

В. И. Зоркальцев^а, А. С. Князев^б

Байкальский гос. университет,
ул. Ленина, 11, 664003 Иркутск, Россия
E-mail: ^а vizork@mail.ru, ^б 010193@bgu.ru

Аннотация. Цель представляемой разработки состояла в создании инструмента для сопоставления алгоритмов оценки динамики численности отдельных видов рыбы озера Байкал на основе экспериментальных отловов. Для каждого экземпляра случайно отобранной рыбы по специальной технологии определяется её возраст. Из полученных данных о численности рыбы разных возрастов в выборке оцениваются параметры заданных законов возрастного распределения. Это служит основанием для формирования представления о динамике смертности и изменения популяции рыбы данного вида в предыдущие годы. Для оценки параметров законов распределения могут использоваться разные алгоритмы, приводящие порой к существенно разным результатам.

Обсуждаемая методика сопоставления алгоритмов оценки параметров основывается на многократных вычислительных экспериментах с использованием метода Монте-Карло для имитации случайных выборок рыб. По предлагаемой методике анализируются несколько алгоритмов оценки параметров усечённого экспоненциального закона распределения при различных объёмах выборок. В качестве примера рассмотрена задача оценки динамики смертности голомянки, основной по биомассе рыбы озера Байкал. Ил. 4, библиогр. 13.

Ключевые слова: оценка параметров смертности рыб, озеро Байкал, голомянка, усечённый экспоненциальный закон распределения, метод статистических испытаний.

Введение

В данной статье излагается методика сравнительного анализа алгоритмов обработки данных возрастной структуры случайных выборок рыб озера Байкал для определения показателей динамики их смертности

по возрастам. Используя разные алгоритмы, можно получать из одних и тех же исходных данных существенно разные результаты. Возникает проблема, какой из алгоритмов следует выбрать? С этой проблемой тесно связаны и другие вопросы. В том числе: как влияет объём используемой выборки на результаты? Какой объём выборки можно считать минимально достаточным для получения надёжных результатов? В данной статье методику сравнительного анализа алгоритмов будем рассматривать на примерах алгоритмов оценки распределения возрастного состава рыб по усечённому экспоненциальному закону.

По традиции теоретическое обоснование алгоритма оценки параметров закона распределения случайной величины осуществляется на основе исследования свойств этого алгоритма при возрастании объёма выборки до бесконечности. При этом используются такие характеристики получаемых оценок, как состоятельность, асимптотическая несмещённость, асимптотическая эффективность, асимптотическая нормальность [1–3]. На практике оценка параметров закона распределения производится на базе ограниченной выборки. Вместе с тем, свойства алгоритмов, выявляемые при возрастании выборки до бесконечности, малопригодны для сравнения этих алгоритмов при оценке параметров на базе ограниченных выборок. О том, что способы оценки параметров, свойственных большим выборкам, не подходят в реальных ситуациях с ограниченным объёмом выборки, писал ещё Гаусс [4].

Статья подготовлена на основе доклада [5], представленного авторами на 2-м Международном семинаре «Вычислительные технологии и прикладная математика», прошедшем 12–16 июня 2023 г. в Благовещенске. Авторы надеются, что представленная в данной статье технология сравнительного анализа методов оценки параметров законов распределения случайных величин на базе вычислительных экспериментов будет полезна не только для исследования и выбора методов оценки динамики смертности рыб озера Байкал, но и в других случаях.

1. Исходные положения

Излагаемая методика базируется на многократной имитации методом статистических испытаний (методом Монте-Карло) заданного объёма случайных реализаций по рассматриваемому закону распределения. Для каждой имитации осуществляется оценка параметров исследуемым алгоритмом. Тем самым получается множество расчётных реализаций параметров, которое можно рассматривать как набор случайных чисел. По их отклонениям от истинного значения параметров можно определять различные статистические характеристики результатов использования алгоритма оценки параметров. Выбор наилучшего алгоритма, исследования влияния объёмов используемых выборок могут осуществляться

путём сопоставления рассчитываемых характеристик для разных алгоритмов, для разных объёмов выборок.

В данной статье излагаются результаты разработки и апробации данной методики на примерах алгоритмов оценки параметров усечённого экспоненциального закона. Согласно этому закону вероятность случайно пойманной рыбе иметь возраст t выражается зависимостью

$$P_t = AR^t, \quad t = 1, \dots, T, \quad (1)$$

где T — заданный максимальный возраст дожития,

$$R = \exp(-\Lambda) \quad (2)$$

— условная вероятность дожития до возраста $t + 1$ для рыбы, дожившей до возраста $t = 1, \dots, T - 1$. Здесь Λ — положительный параметр, который принято называть применительно к динамике жизни биологических организмов коэффициентом смертности. Из условия, что сумма вероятностей P_t , $t = 1, \dots, T$, равна единице, следует, что

$$A = \frac{1 - R}{1 - R^{T+1}}. \quad (3)$$

Отметим, что при фиксированном T все величины определяются однозначно, если задать показатель $R \in (0, 1)$ или параметр $\Lambda > 0$.

Оценки указанных параметров будем обозначать теми же, только строчными буквами. В этой статье производится сопоставление по четырём возможным методам, каждый из которых применяется в трёх вариантах. Итого будет рассмотрено двенадцать алгоритмов оценки параметров усечённого экспоненциального закона распределения.

Методика использует m случайных выборок рыб одного и того же объёма N с заданным законом распределения случайной величины возраста рыб в выборке. Обозначим через N_{ti} численность рыб возраста t в выборке $i \in \{1, \dots, m\}$,

$$\sum_{t=1}^T N_{ti} = N, \quad i = 1, \dots, m. \quad (4)$$

В проведённых ниже расчётах используется $m = 50\,000$ выборок, что является достаточным для получения устойчивых результатов. В первых двух методах сначала оценивается условная вероятность r дожития до следующего года для рыбы, дожившей до года $t = 1, \dots, T - 1$. Затем на основе (1)–(3) определяются оценки λ, a, p_t . В третьем и четвертом методах сначала вычисляется коэффициент смертности λ , а затем, исходя из (1)–(3), определяются значения r, a, p_t , $t = 1, \dots, T$.

В рамках каждого из рассматриваемых методов алгоритмы отличаются используемыми в расчётах положительными весовыми коэффициентами h_t . Рассматриваются следующие весовые коэффициенты:

1) неизменные во времени

$$h_t = \frac{1}{T-1}, \quad t = 1, \dots, T, \quad (5)$$

2) пропорциональные оценкам вероятности p_t

$$h_t = p_t / \sum_{\tau=1}^{T-1} p_{\tau}, \quad t = 1, \dots, T, \quad (6)$$

3) пропорциональные обратным значениям оценок дисперсии случайных величин p_t

$$h_t = \frac{1}{p_t(1-p_t)} / \sum_{\tau=1}^{T-1} \frac{1}{p_{\tau}(1-p_{\tau})}, \quad t = 1, \dots, T. \quad (7)$$

Во всех трёх вариантах сумма первых $T-1$ весовых коэффициентов равна единице. В правилах (6), (7) весовые коэффициенты зависят от величин p_t , которые предстоит определить с использованием этих же весовых коэффициентов. Это достигается применением итеративной процедуры, включающей на каждой итерации оценку вероятностей p_t (при некоторых заданных в начале итерации весовых коэффициентах), а затем пересчёт весовых коэффициентов по формулам (6) или (7) с использованием полученных оценок p_t . Этот процесс повторяется, пока оценки вероятностей p_t на соседних итерациях не совпадут с заданной точностью. В качестве начального приближения используются одинаковые весовые коэффициенты, определяемые по формуле (4).

Приведём расчётные формулы методов оценки параметров. В этих описаниях индекс — номер выборки i опущен. Используются относительные величины

$$n_t = \frac{N_t}{N}, \quad t = 1, \dots, T,$$

$$q_t = \frac{N_{t+1}}{N_t}, \quad t = 1, \dots, T-1.$$

1. Оценка условной вероятности дожития до следующего года в виде взвешенного среднего арифметического от соотношений количеств рыб соседних возрастов в выборке равна

$$r = \sum_{t=1}^{T-1} h_t q_t. \quad (8)$$

2. Оценка условной вероятности дожития до следующего года в виде взвешенного среднего геометрического от соотношений количеств рыб

соседних возрастов в выборке равна

$$r = \prod_{t=1}^{T-1} q_t^{h_t}. \quad (9)$$

В обоих методах на основе найденного по формулам (8) или (9) значения r вычисляются

$$\lambda = -\ln(r), \quad a = \frac{1-r}{1-r^{T+1}},$$

$$p_t = ar^t, \quad t = 1, \dots, T.$$

3. Оценка коэффициента смертности с использованием метода наименьших квадратов в логарифмической шкале. Сначала определяются значения переменных α , λ в результате безусловной минимизации квадратичной выпуклой функции от двух переменных

$$\varphi(\alpha, \lambda) = \sum_{t=1}^T h_t (\ln n_t - \alpha - t\lambda)^2. \quad (10)$$

Затем, исходя из (1)–(3), определяются значения

$$r = \exp(-\lambda), \quad a = \frac{1-r}{1-r^{T+1}},$$

$$p_t = ar^t, \quad t = 1, \dots, T. \quad (11)$$

4. Оценка методом наименьших квадратов в исходной шкале. Необходимо найти минимум функции одной переменной

$$f(\lambda) = \sum_{t=1}^T h_t (n_t - a(\lambda) \exp(-\lambda t))^2, \quad (12)$$

$$a(\lambda) = \frac{1-r(\lambda)}{1-r(\lambda)^{T+1}}, \quad r(\lambda) = \exp(-\lambda). \quad (13)$$

Остальные параметры вычисляются по формулам (11). Функция f минимизируется методом золотого сечения [6].

Можно считать, что изложенные методы представлены в последовательности от наиболее простых к более сложным в отношении восприятия и вычислений. В созданном программно-вычислительном комплексе предусмотрена возможность добавления других методов.

Результаты оценки параметров для каждого варианта случайных выборок $i = 1, \dots, m$ используются для вычислений обобщённых характеристик данного метода. В реализованном к настоящему времени программно-вычислительном комплексе рассчитываются следующие характеристики.

1. Математическое ожидание коэффициента смертности, абсолютное и относительное смещение математического ожидания от истинного значения коэффициента смертности:

$$\mathbb{M}\lambda = \frac{1}{m} \sum_{i=1}^m \lambda_i, \quad \Delta = \mathbb{M}\lambda - \Lambda, \quad \delta = \frac{\Delta}{\Lambda}. \quad (14)$$

2. Абсолютные и относительные средние квадратические отклонения оценок коэффициента смертности от математического ожидания этих оценок и от истинного значения этих оценок:

$$D = \left(\frac{1}{m} \sum_{i=1}^m (\lambda_i - \mathbb{M}\lambda)^2 \right)^{1/2}, \quad \delta D = \frac{D}{\mathbb{M}\lambda}, \quad (15)$$

$$Q = \left(\frac{1}{m} \sum_{i=1}^m (\lambda_i - \Lambda)^2 \right)^{1/2}, \quad \delta Q = \frac{Q}{\Lambda}.$$

3. Вероятности (частоты) попадания значений λ_i в заданные интервалы (например в 20% или 10%) от математического ожидания $\mathbb{M}\lambda$ и от истинного значения Λ .

4. Медиана оценок коэффициента смертности: значение λ , при котором половина оценок λ_i будет меньше этой величины. Абсолютные и относительные отклонения медианы от истинного значения Λ .

Рассматриваемая методика позволяет использовать и другие показатели качества алгоритмов оценки параметров. Например, приведённые показатели можно дополнить аналогичными характеристиками для условной вероятности r дожития до следующего года рыб, доживших до данного возраста.

2. Обсуждение

Можно выделить три источника погрешностей в оценках динамики смертности рыб на основе данных экспериментальных отловов.

1. Неадекватность рассматриваемой модели при сопоставлении с действительностью. В данном случае — возможное несоответствие реального распределения возраста рыб усечённому экспоненциальному закону. Об экспоненциальном распределении можно говорить, если есть веские основания полагать, что ежегодно в водоёме появляется примерно одно и то же число рыб-сеголеток исследуемого вида, доля выживающих за год рыб примерно одинаковая для каждого возраста, и эта доля не изменяется во времени.

Для указанных условий стационарности в Байкале наиболее подходящей является рыба голомянка. Имеются два вида — большая и малая голомянка, суммарно составляющих по имеющимся оценкам около

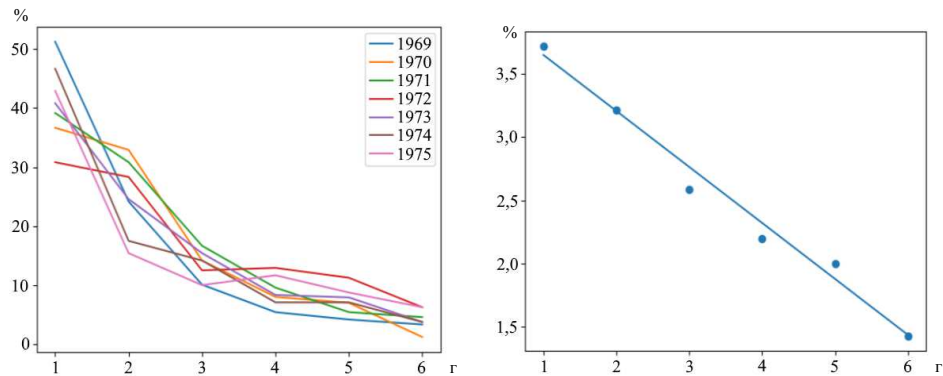


Рис. 1. Возрастной состав (%) большой голомянки в отловах 1969–75 гг. [7] и линейная аппроксимация усреднённых данных в логарифмической шкале

70% рыб Байкала [7]. Голомянка — живородящая рыба (вынашивающая внутри себя личинки из икры), не собирающаяся в стаи — распределённая по всей территории и по всей глубине озера. Оба вида голомянок всю свою жизнь проводят непосредственно в акватории озера, поэтому не испытывают особого влияния колебаний приточности воды в реках, впадающих в Байкал. Конечно, влияние на численность голомянки могут оказывать изменения кормовой базы (биомассы отдельных видов зоопланктона) и, вероятно, некоторые отмечаемые изменения температуры верхних слоёв воды озера Байкал. К факторам возможной нестационарности условий существования голомянки следует отнести и колебание численности байкальской нерпы, для которой голомянка является источником питания. Представленные на рис. 1 данные Старикова [7] о возрастной структуре большой голомянки в отловах 1969–75 гг. можно рассматривать как подтверждение возможности использования усечённого экспоненциального закона распределения для описания структуры возрастного состава.

Ещё более оправданным выглядит такое допущение при рассмотрении усреднённых за весь указанный семилетний период данных о возрастной структуре большой голомянки. Как видно из рис. 1, усреднённая за семь лет возрастная структура в логарифмической шкале хорошо аппроксимируется линейной зависимостью. В представленных ниже результатах расчётов будем рассматривать усечённый экспоненциальный закон с параметрами $T = 6$ и $\Lambda = 0,44$. Согласно рис. 1 это примерно соответствует ожидаемым значениям параметров усечённого экспоненциального закона распределения для большой голомянки.

Помимо экспоненциального могут рассматриваться и другие законы распределения случайной величины. В том числе, более общие, которые

учитывают возможные зависимости коэффициента Λ от возраста. К ним относится распределение Вейсбула [1], в котором коэффициент Λ задаётся в виде линейной функции от возраста.

2. Погрешности экспериментов, непредставительность используемых выборок. Причиной этого могут быть случайные и систематические отклонения в возрастной структуре выборки, влияние на используемые в расчётах исходные данные времени, места и технологии проведения отловов. Для промысловых рыб (в частности для байкальского омуля [8]) отмечается сокращение удельного веса рыб младших возрастов, поскольку эти рыбы из-за меньших размеров чаще проходят сквозь ячейки рыболовецких сетей, а также из-за мотивации рыболовов к вылову крупных рыб.

Погрешности возможны и в определении возраста отдельных экземпляров рыб, что оценивается на основе специальной трудоёмкой технологии анализа рыбной чешуи, отолитов и костей рыб [9]. Одним из планируемых направлений развития излагаемой в этой статье методики оценки параметров является включение в неё методов, учитывающих возможные погрешности в самих исходных данных в том числе в оценках возраста отдельных экземпляров рыб.

3. Погрешности алгоритмов расчёта. На одних и тех же исходных данных при оценке параметров одного и того же закона распределения случайной величины можно получать существенно разные результаты разными алгоритмами оценки параметров. Излагаемая здесь методика исследования алгоритмов нацелена в основном на уменьшение роли этих погрешностей.

3. Результаты расчётов

На рис. 2–4 представлены результаты расчётов для трёх из приведённых выше характеристик качества алгоритмов оценки параметров в зависимости от объёма выборки N . Рассматриваются следующие характеристики: относительное смещение δ математического ожидания оценки коэффициента смертности от его истинного значения (14) (рис. 2); относительное среднеквадратическое отклонение δD оценки коэффициента смертности от её математического ожидания (15) (рис. 3); частота попадания оценки коэффициента смертности в 10%-й интервал от его истинного значения (рис. 4). На каждом рисунке отражены расчёты для всех четырёх изложенных методов с тремя видами весовых коэффициентов: а) метод взвешенного среднего арифметического (8); б) метод взвешенного среднего геометрического (9); в) метод наименьших квадратов в логарифмической шкале (10)–(11); г) метод наименьших квадратов в исходной шкале (12)–(13). Способ взвешивания обозначен цветом: зелёный — равные неизменные веса (5); жёлтый — веса, пропорциональные оценкам

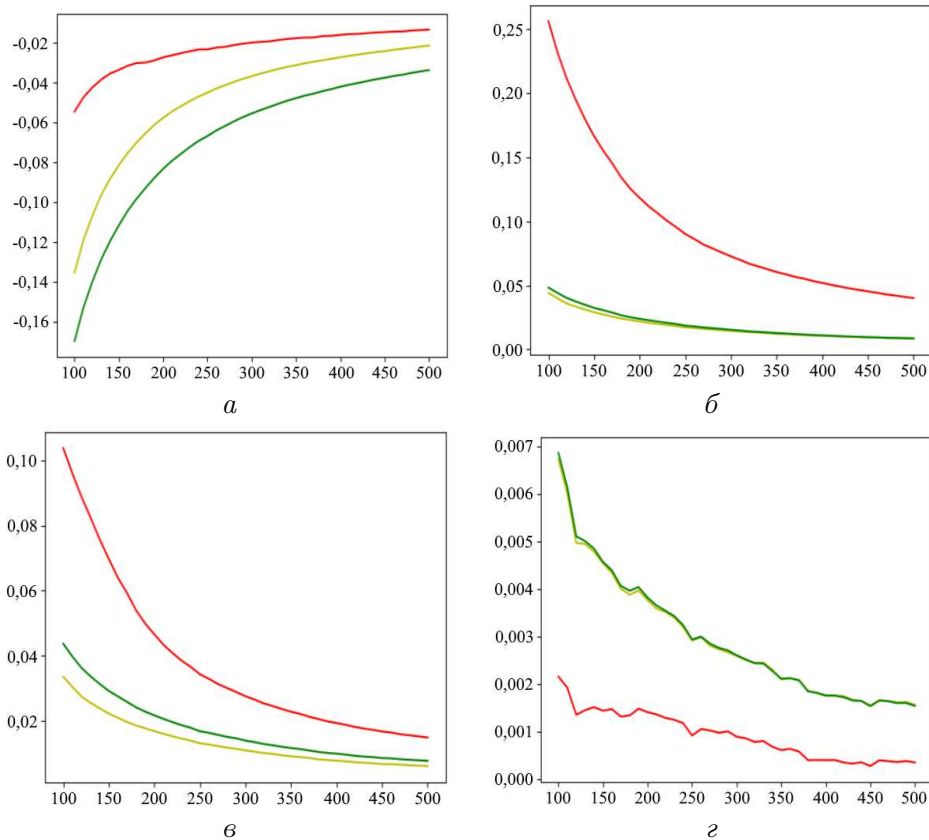


Рис. 2. Относительное смещение δ математического ожидания оценки коэффициента смертности от его истинного значения

вероятности (6); красный — веса, пропорциональные обратным значениям оценок дисперсии (7). В представленных расчётах объём выборки варьируется в диапазоне 100–500 экземпляров рыб.

Из представленных расчётов видно, что выбор метода и правила задания весовых коэффициентов и объёмов используемых выборок имеет очень существенное значение. Все методы имеют значительно худшие характеристики при малых объёмах используемых выборок, скажем, при $N \leq 300$, чем при больших значениях N . Этот факт важен для выработки рекомендации о минимально необходимом числе рыб в выборках.

Качество методов по всем рассматриваемым характеристикам возрастает от более простых к более сложным. Если для метода среднего геометрического (второго из описанных выше) смещение математического ожидания λ достигает 15–25% при выборке в 100–200 экземпляров, то для четвёртого метода смещение имеет малозначительную величину

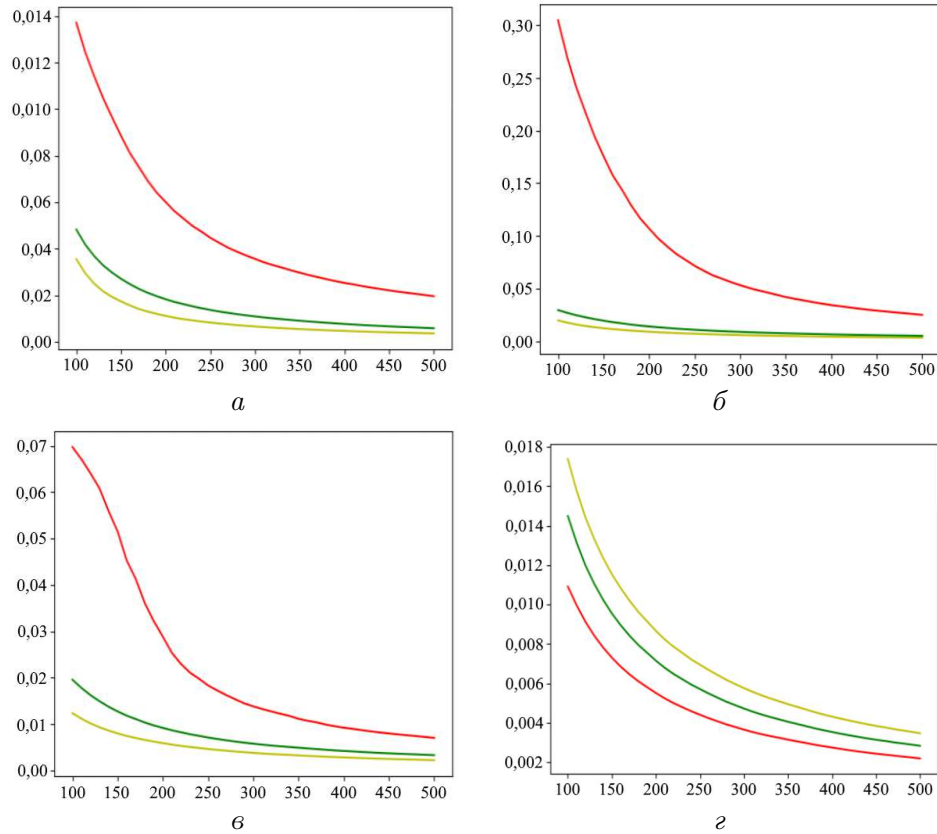


Рис. 3. Относительное среднеквадратическое отклонение δD оценки коэффициента смертности от её математического ожидания

в 0,7–0,4% при таких же объёмах выборки. Малая величина смещения для метода (12), (13), выражаемая тысячными долями, объясняет, почему в этом случае построенная расчётная зависимость относительного смещения λ не имеет монотонного и гладкого характера. Чтобы достичь строгой монотонности и гладкости графика, надо многократно увеличить число выборок m .

Для первых трёх методов явно лучшим оказался второй способ взвешивания (6), а не третий (7). Проблема выбора весов при оценке параметров усечённого экспоненциального метода распределения возраста большой и малой голомянки третьим из изложенных здесь методов обсуждалась в статье [10]. Обычно в литературе по методам математической статистики рекомендуется использовать веса, пропорциональные обратным значениям точности измерений показателей, под которой понимается оценка дисперсии измеряемой величины. Это соответствует третьему

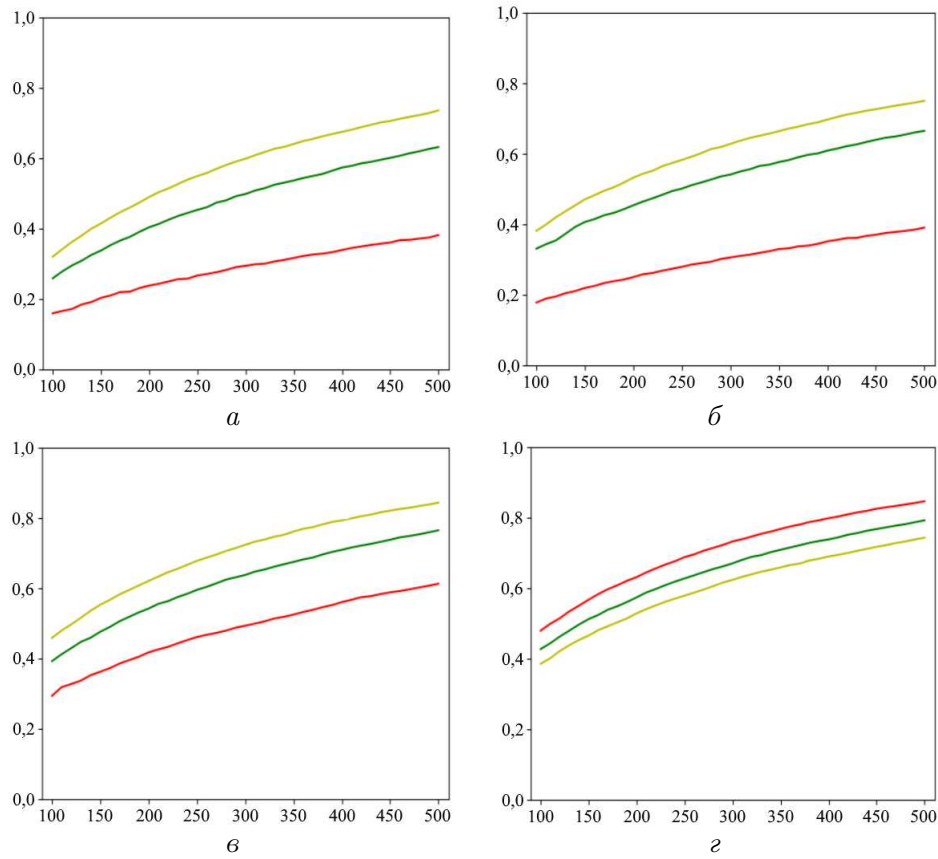


Рис. 4. Вероятность попадания оценки коэффициента смертности в 10%-й интервал от его истинного значения

способу взвешивания, выражаемому формулой (7). В таком случае веса h_t должны возрасть с увеличением t . В то же время имелись весо-
мые аргументы, изложенные в [10], по использованию весов h_t , убывающих с увеличением t , а именно весовых коэффициентов (6). Стремление строго доказать преимущество коэффициентов (6) привело к реализации первоначального варианта излагаемой здесь методики [10]. В [10] представлены результаты исследования первых трёх из рассматриваемых здесь четырёх методов для усечённого экспоненциального распределения при $T = 6$ и $\Lambda = 0,65$. Это примерно соответствует параметрам малой голомянки [7]. Соотношения исследовавшихся показателей оказались примерно такими же, как и представленные в настоящей статье для данных, примерно адекватных для большой голомянки.

Оценка динамики смертности (по различным причинам) отдельных видов организмов озера Байкал необходима для построения автономных

моделей функционирования этих организмов в Байкале, что является одним из этапов создания модели экосистемы озера с учётом взаимодействий разных видов между собой [12, 13].

Наилучшие характеристики из приведённых здесь имеет четвёртый метод, базирующийся на минимизации функции (12) и расчётных формулах (13). Причём для этого метода лучшим оказался третий вариант взвешивания, а увеличение объёма используемой выборки до 300–500 экземпляров позволяет увеличить вероятность попадания оценки коэффициента смертности в 10%-й интервал от истинного значения до 70–80%, что не достижимо другими методами из рассмотренных.

Можно отметить, что и по другим характеристикам, не приведённым здесь из-за ограниченного объёма статьи, также отмечается существенное превосходство четвёртого из рассматриваемых методов при небольшом преимуществе третьего способа взвешивания. Конечно, необходимо более подробное исследование соотношений расчётных характеристик алгоритмов оценки параметров, устойчивости этих соотношений к варьированию исходного значения Λ и теоретическое объяснение получаемых результатов.

4. Некоторые выводы

1. Представленные результаты расчётов иллюстрируют работоспособность развиваемой методики исследования алгоритмов оценки параметров законов распределения на базе вычислительных экспериментов, важность учёта объёма используемых выборок, важность выбора правила задания весовых коэффициентов. Можно утверждать, что анализ алгоритмов оценки параметров по рассматриваемой здесь технологии должен быть необходимым этапом при сопоставлениях и обосновании методов оценки параметров законов распределения случайных величин на базе ограниченных выборок.

2. Расчёты иллюстрируют преимущество второго способа взвешивания для рассмотренных здесь первых трёх методов оценки параметров и преимущество четвёртого из рассмотренных методов оценки параметров на этот раз с третьим способом взвешивания. Можно отметить, что такие соотношения преимуществ алгоритмов имеют место и по другим рассчитанным характеристикам, не приведённым здесь из-за ограниченности объёма статьи.

3. Данная методика может использоваться не только для усечённого экспоненциального, но и для других законов распределения случайных величин. Может быть расширен состав рассчитываемых характеристик алгоритмов оценки параметров, расширен набор методов оценки параметров (целесообразно, в частности, включение в этот набор метод максимального правдоподобия).

В дальнейшем планируется осуществить развитие данной методики в том числе в направлении учёта возможных ошибок в определении возраста рыб, использования других законов распределения и выявления возможных возмущений, приводящих к увеличению или сокращению численности рыб в отдельные годы прошлых лет. Требуется развитие технологии в направлении исследования устойчивости получаемых характеристик алгоритмов в окрестности ожидаемых значений параметров законов распределения.

Финансирование работы

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

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Кобзарь А. И. Прикладная математическая статистика. М.: Физматлит, 2012. 816 с.
2. Трофимова Е. А., Кисляк Н. В., Гилёв Д. В. Теория вероятностей и математическая статистика. Екатеринбург: Изд-во Урал. ун-та, 2018. 160 с.
3. Козлов М. В., Прохоров А. К. Введение в математическую статистику. М.: МГУ, 1987. 264 с.
4. Гаусс К. Ф. Избранные геодезические сочинения. Т. 1. М.: Геодезиздат, 1957. 152 с.
5. Зоркальцев В. И., Князев А. С. Сравнительный анализ методов измерения динамики смертности рыб озера Байкал на основе вычислительных экспериментов // Вычислительные технологии и прикладная математика. Мат. II Междунар. семин. (Благовещенск, Россия, 12–16 июня 2023 г.). Благовещенск: Изд-во Амур. гос. ун-та, 2023. С. 91–94.
6. Зоркальцев В. И. Элементы оптимизации. Иркутск: ИСЭМ СО РАН, 2014. 100 с.
7. Стариков Г. В. Голомянки Байкала. Новосибирск: Наука, 1977. 93 с.
8. Соколов А. В., Петекрфельд В. А., Бобков А. И. [и др.]. Омуль байкальский // Материалы, обосновывающие общие допустимые уловы водных биологических ресурсов в озере Байкал на 2018 год. М.: ВНИРО, 2017. С. 5–24.
9. Кафанова В. В. Методы определения возраста и роста рыб. Томск: Изд-во Томск. ун-та, 1984. 56 с.
10. Бычков И. В., Зоркальцев В. И., Казазаева А. В. Весовые коэффициенты в методе взвешенных наименьших квадратов // Сиб. журн. вычисл. математики. 2015. Т. 18, № 3. С. 275–288.

11. Бычков И. В., Зоркальцев В. И., Мокрый И. В. Сопоставление методов оценки параметров усечённого экспоненциального закона распределения на основе вычислительных экспериментов // Вычисл. технологии. 2018. Т. 23, № 5. С. 3–20.
12. Аношко П. Н., Афанасьева Э. Л., Дзюба Е. В. [и др.]. Формирование системы моделей трофических взаимоотношений в экосистеме пелагиали оз. Байкал. Иркутск, 1998. 21 с. (Препр. / ИСЭМ СО РАН; № 11).
13. Зоркальцев В. И., Мокрый И. В., Казазаева А. В. Моделирование пелагического сообщества экосистемы озера Байкал // Вычисл. технологии. 2011. Т. 16, № 1. С. 48–60.

Зоркальцев Валерий Иванович
Князев Александр Сергеевич

Статья поступила
6 сентября 2023 г.
После доработки —
13 ноября 2023 г.
Принята к публикации
22 декабря 2023 г.

COMPARATIVE ANALYSIS OF ALGORITHMS
FOR ESTIMATING FISH POPULATION DYNAMICSV. I. Zorkaltsev^a and A. S. Knyazev^bBaikal State University,
11 Lenin Street, 664003 Irkutsk, Russia
E-mail: ^avizork@mail.ru, ^b010193@bgu.ru

Abstract. The aim of the present work is to create a tool for comparing algorithms for estimating the dynamics of the abundance of individual fish species in Lake Baikal based on experimental catching. For each instance of a randomly selected fish, its age is determined using a special technology. From the obtained data on the numbers of fish of different ages in the sample, the parameters of the given laws of age distribution are estimated. This serves as a basis for the formation of ideas about the dynamics of mortality and changes in the abundance of fish of this species in previous years. Various algorithms can be used to estimate the parameters of distribution laws, sometimes leading to considerably different results.

The discussed technique for comparing parameter estimation algorithms is based on multiple computational experiments using the Monte Carlo method to simulate random samples of fish. The proposed method analyzes several algorithms for estimating the parameters of a truncated exponential distribution law for different sample sizes. As an example, the problem of estimating the mortality dynamics of the Baikal oilfish (Comephorus), the major biomass fish of Lake Baikal, is considered. Illustr. 4, bibliogr. 13.

Keywords: estimation of fish mortality parameters, Lake Baikal, comephorus, truncated exponential distribution law, method of statistical testing.

References

1. A. I. Kobzar', *Applied Mathematical Statistics* (Fizmatlit, Moscow, 2012) [Russian].

English transl.: *Journal of Applied and Industrial Mathematics* **18** (2), 361–370 (2024), DOI: 10.1134/S1990478924020170.

2. **E. A. Trofimova, N. V. Kislyak, and D. V. Gilyov**, *Probability Theory and Mathematical Statistics* (Izd. Ural. Univ, Yekaterinburg, 2018) [Russian].
3. **M. V. Kozlov and A. K. Prokhorov**, *Introduction to Mathematical Statistics* (MGU, Moscow, 1987) [Russian].
4. **K. F. Gauss**, *Selected Geodesic Works*, Vol. 1 (Geodezizdat, Moscow, 1957) [Russian].
5. **V. I. Zorkaltsev and A. S. Knyazev**, Comparative analysis of methods for assessing the dynamics of fish mortality in Lake Baikal based on computational experiments, in *Computation Technologies and Applied Mathematics* (Proc. 2nd Int. Semin., Blagoveshchensk, Russia, June 12–16, 2023) (Izd. Amur. Gos. Univ., Blagoveshchensk, 2023), pp. 91–94 [Russian].
6. **V. I. Zorkaltsev**, *Elements of Optimization Theory* (ISEhM SO RAN, Irkutsk, 2014) [Russian].
7. **G. V. Starikov**, *Golomyankas of The Baikal* (Nauka, Novosibirsk, 1977) [Russian].
8. **A. V. Sokolov, V. A. Petekrfeld, A. I. Bobkov**, [et al.], The Baikal omul, in *Proceedings Justifying the Total Allowable Catches of Aquatic Biological Resources in Lake Baikal for 2018* (VNIRO, Moscow, 2017), pp. 5–24 [Russian].
9. **V. V. Kafanova**, *Methods for Determining the Age and Growth of Fish* (Izd. Tomsk. Univ., Tomsk, 1984) [Russian].
10. **I. V. Bychkov, V. I. Zorkaltsev, and A. V. Kazazaeva**, Weight coefficients in the weighted least squares method, *Sib. Zh. Vychisl. Mat.* **18** (3), 275–288 (2015) [Russian] [*Numer. Anal. Appl.* **18** (3), 223–234 (2015)].
11. **I. V. Bychkov, V. I. Zorkaltsev, and I. V. Mokryi**, Estimates of parameters for the truncated exponential law of distribution on the basis of exponential data, *Vychisl. Tekhnol.* **23** (5), 3–20 (2018) [Russian].
12. **P. N. Anoshko, Eh. L. Afanasyeva, E. V. Dzyuba**, [et al.], Formation of a system of models for trophic relationships in the pelagic ecosystem of Lake Baikal. Irkutsk, 1998. (Prepr., ISEhM SO RAN, No. 11) [Russian].
13. **V. I. Zorkaltsev, I. V. Mokryi, and A. V. Kazazaeva**, Modeling the pelagic community of Lake Baikal ecosystem, *Vychisl. Tekhnol.* **16** (1), 48–60 (2011) [Russian].

Valery I. Zorkaltsev
Aleksandr S. Knyazev

Received September 6, 2023
Revised November 13, 2023
Accepted December 22, 2023

СПОСОБ ОПТИМАЛЬНОГО ИЗМЕРЕНИЯ ЗНАЧЕНИЙ НЕКОТОРЫХ ПАРАМЕТРОВ FAULT-АТАК НА КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ

Ю. А. Зуев^а, П. Г. Ключарёв^б

Московский гос. технический университет им. Н. Э. Баумана,
ул. 2-я Бауманская, 5, 105005 Москва, Россия

E-mail: ^а79851965730@yandex.ru, ^бpk.iu8@yandex.ru

Аннотация. Работа посвящена построению оптимального по времени способа измерения максимально допустимого (критического) значения напряжения питания (а также других параметров) устройства, на котором выполняется криптографический алгоритм. Знание этих величин необходимо для успешного проведения индуцирования ошибок в рамках проведения fault-атак. Способ построен на основе метода динамического программирования. Библиогр. 11.

Ключевые слова: fault-атака, динамическое программирование.

Введение

Один из важнейших классов атак по побочным каналам на практические реализации криптоалгоритмов — это так называемые fault-атаки или атаки по ошибкам вычислений [1–7]. Они основаны на индуцировании ошибок в работе аппаратных устройств, на которых выполняются криптографические алгоритмы. Эти ошибки приводят к искажениям информации, при анализе которых в ряде случаев можно восстановить секретный ключ. Индуцирование ошибок производится различными способами, среди которых можно назвать, в частности, воздействие магнитными и электромагнитными полями, высокой температурой, лазерным лучом и т. д. Пожалуй, наиболее распространённым является индуцирование ошибок с помощью изменения напряжения питания. Существенное повышение или понижение напряжения либо изменение напряжения по сложному закону часто способны индуцировать необходимые для проведения атаки ошибки. Во время практической реализации такой атаки необходимо знать максимальное напряжение, не приводящее к выходу устройства из строя. Назовём это напряжение критическим. Измерение критического напряжения может сводиться к постепенному увеличению

напряжения питания устройства с периодической проверкой его работоспособности. Не следует однако забывать, что при достижении критического напряжения устройство может выйти из строя далеко не мгновенно, поэтому процесс измерения критического напряжения может занять длительное время. Вместе с тем, если имеется несколько идентичных образцов устройства, то процесс измерения критического напряжения можно оптимизировать по времени.

В связи с изложенным рассмотрим следующую задачу. Пусть проводится измерение критического напряжения следующим образом. Рассматривается монотонно возрастающая конечная последовательность напряжений питания: $u_1, u_2, \dots, u_n$. Если в процессе измерения мы готовы испортить только один образец устройства, то последовательно подавая на него напряжения $u_1, u_2, \dots, u_i$ и зафиксировав выход устройства из строя при напряжении u_i , прекращаем испытания, установив тем самым, что критическое напряжение равно u_{i-1} .

При наличии одного образца — это единственный план проведения измерения критического напряжения. Его осуществление может потребовать n испытаний. Однако если имеется несколько идентичных опытных образцов, то максимальное число требуемых испытаний может быть уменьшено. Простейшим является случай, когда число опытных образцов потенциально не ограничено. При этом условии оптимальная стратегия состоит в выборе напряжения $u_{\lceil n/2 \rceil}$ с последующим переходом к большему или меньшему напряжению в зависимости от результата испытания и дальнейшими аналогичными дихотомиями. Такая стратегия включает $\lfloor \log_2 n \rfloor + 1$ испытаний, а для её осуществления потребуется $\lfloor \log_2 n \rfloor + 1$ опытных образцов.

Интерес представляет исследование промежуточных случаев, когда при ограниченном числе образцов k требуется найти $\min \max$ (число испытаний), где $\min$ берётся по всем возможным стратегиям проведения испытаний, а $\max$ означает выбор для стратегии самого трудного для неё случая. Обозначим этот $\min \max$, зависящий от k и n , через $l_k(n)$. Наш предварительный анализ показывает, что $l_1(n) = n$ и $l_k(n) = \lfloor \log_2 n \rfloor + 1$, если $k \geq \lfloor \log_2 n \rfloor + 1$. Заметим также, что если каждое испытание занимает одну единицу времени, то $l_k(n)$ представляет собой полное время испытаний, и задача может быть сформулирована как минимизация времени испытаний.

С чисто математической точки зрения это задача оптимального определения монотонной функции, заданной на цепи высоты n (т. е. линейно упорядоченном множестве из n элементов) и принимающей два значения: 0 или 1. Своеобразие задачи заключается в том, что в процессе распознавания значение 1 (выход образца из строя) гарантированно не должно появиться более k раз.

В работе даётся эффективный алгоритм типа динамического программирования для получения оптимального плана испытаний для любых значений k и n . При $k = 2$ указана точная формула для $l_2(n)$, и найдена асимптотика для $l_k(n)$ при фиксированном k и $n \rightarrow \infty$. Практически все полученные результаты фактически основаны на том очевидном свойстве функции $l_k(n)$, что она является монотонно неубывающей функцией от n и монотонно невозрастающей функцией от k .

При построении оптимального плана будем опираться на принцип динамического программирования Ричарда Беллмана, согласно которому «оптимальное поведение обладает тем свойством, что, каковы бы ни были первоначальное состояние и решение в начальный момент, последующие решения должны составлять оптимальное поведение относительно состояния, получающегося в результате первого решения» [8]. Этот принцип широко применяется в различных областях [9–11]. Также будем использовать принцип балансировки трудоёмкости на двух ветвях испытания, возникающих при выходе и невыходе из строя образца, который можно описать следующим образом.

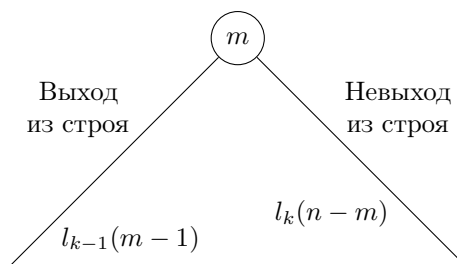


Рис. 1. Две ветви испытания

Как показано на рис. 1, если исследуется диапазон напряжений питания $u_1, u_2, \dots, u_n$ и на испытываемый образец подаётся напряжение u_m , то в случае выхода образца из строя диапазон подлежащих дальнейшему исследованию напряжений становится равным $u_1, u_2, \dots, u_{m-1}$, а число образцов уменьшается на единицу. Если же при подаче напряжения u_m образец не выходит из строя, то диапазон подлежащих дальнейшему исследованию напряжений становится равным $u_{m+1}, u_{m+2}, \dots, u_n$, а число доступных для использования образцов остаётся неизменным. Тем самым в первом случае на выходе имеем $l_{k-1}(m-1)$, а во втором — $l_k(n-m)$.

При этом в оптимальном алгоритме шаг m в последовательности напряжений $u_1, u_2, \dots, u_n$ должен быть выбран таким образом, чтобы минимизировать величину $\max\{l_{k-1}(m-1), l_k(n-m)\}$. Так как величина $l_j(i)$ при фиксированном j монотонно не убывает и с увеличением i на единицу может возрасти на единицу или остаться прежней, номер

напряжения m , при котором $l_{k-1}(m-1) = l_k(n-m)$, является оптимальным для выбора. В этом и состоит принцип балансировки трудоёмкости на двух ветвях испытания.

Здесь, однако, следует заметить, что точный баланс между ветвями не всегда может быть достигнут. Если для некоторого m имеет место $l_k(n-m) - l_{k-1}(m-1) = 1$ и при увеличении m на единицу $l_{k-1}(m-1)$ возрастает на единицу, а $l_k(n-m)$ на единицу убывает, то оказывается, что $l_k(n-m) - l_{k-1}(m-1) = -1$, и точного баланса добиться невозможно. В этом случае в оптимальном алгоритме может быть выбрано любое из двух значений: либо m , либо $m+1$.

1. Алгоритм построения оптимального плана

Для $l_k(n)$ имеем следующие очевидные граничные условия:

$$\begin{aligned} l_1(i) &= i, & i &= 1, 2, \dots, n; \\ l_i(1) &= 1, & i &= 1, 2, \dots, k. \end{aligned} \quad (1)$$

В основе алгоритма (см. алгоритм 1) лежит следующее рекуррентное соотношение:

$$l_k(n) = \min_{1 \leq m \leq \lceil n/2 \rceil} \max\{l_{k-1}(m-1), l_k(n-m)\} + 1, \quad (2)$$

где m — номер выбираемого при испытании напряжения u_m , $l_{k-1}(m-1)$ относится к ветви, связанной с выходом из строя образца, а $l_k(n-m)$ — с невыходом из строя. Диапазон проверяемых напряжений заканчивается на номере $\lceil n/2 \rceil$ ввиду того, что функция $l_k(n)$ при фиксированном k не убывает по n , а при фиксированном n не возрастает по k . Здесь и работает принцип балансировки между двумя ветвями.

Алгоритм 1. Построение оптимального плана

```

1: function OPTIMALPLAN( $n, k$ )
2:   for  $i = 1$  to  $n$  do
3:      $l_1(i) = i$ 
4:   for  $i = 1$  to  $k$  do
5:      $l_i(1) = 1$ 
6:   for  $i = 2$  to  $k$  do
7:     for  $j = 2$  to  $n$  do
8:        $l_i(j) = \min_{1 \leq m \leq \lceil j/2 \rceil} \max\{l_{i-1}(m-1), l_i(j-m)\} + 1$ 

```

С помощью равенства (2) и граничных условий (1) построчно заполняем $(k \times n)$ -таблицу, последовательно вычисляя $l_i(j)$ для $i = 2, 3, \dots, k$, $j = 2, 3, \dots, n$. В каждую клетку (i, j) таблицы наряду со значением $l_i(j)$

заносится номер m напряжения питания u_m , которое следует использовать. Тем самым задаётся оптимальный план проведения исследования. Для любого числа используемых образцов $1 \leq i \leq k$ и длины $1 \leq j \leq n$ диапазона напряжений, которые могут возникнуть в ходе измерения, определено напряжение u_m , которое следует использовать, переходя затем к бóльшим или меньшим напряжениям в зависимости от результата испытания. Трудоемкость алгоритма равна $O(kn^2)$ операций, так как заполнение каждой клетки таблицы требует $O(n)$ сравнений.

2. Определение $l_2(n)$

Примечательно, что в случае двух образцов оказывается возможным задать оптимальный план простой формулой, не прибегая к использованию приведённого выше алгоритма. Основная идея получения такой формулы также опирается на сбалансированность в оптимальном плане числа испытаний на двух ветвях алгоритма, получаемых при выходе и невыходе из строя опытного образца (рис. 2).

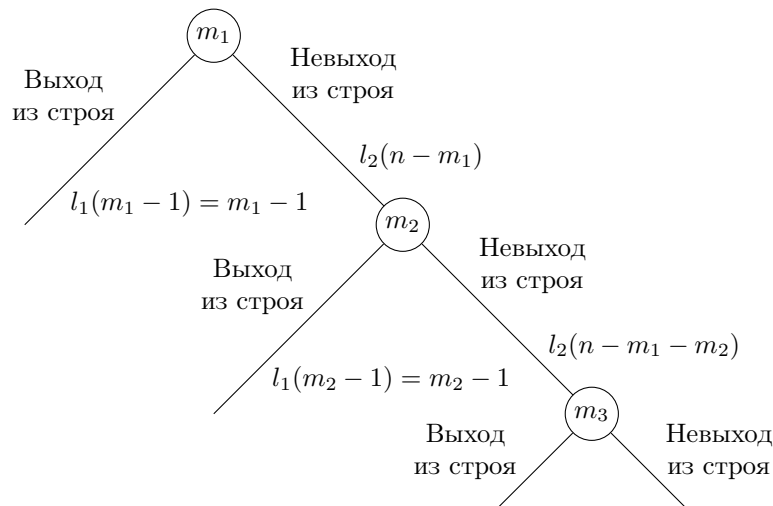


Рис. 2. Сбалансированность числа испытаний на ветвях алгоритма в случае двух образцов

Если для первого испытания выбрано напряжение с номером m_1 , то при выходе образца из строя исследование будет продолжено со вторым образцом на напряжениях с номерами от 1 до $m_1 - 1$, которое может потребовать проведения $l_1(m_1 - 1) = m_1 - 1$ испытаний, а полное число испытаний окажется равным m_1 . Если же образец не выйдет из строя, то исследование продолжится с двумя образцами на $n - m_1$ напряжениях

в диапазоне от u_{m_1+1} до u_n . При этом если для второго эксперимента выбирается напряжение с номером m_2 из нового диапазона, то в случае разрушения образца исследование будет проводиться со вторым образцом на напряжениях с номерами от 1 до $m_2 - 1$, которое может потребовать проведения $l_1(m_2 - 1) = m_2 - 1$ испытаний. Согласно принципу сбалансированности имеем $l_1(m_1 - 1) + 1 = l_1(m_2 - 1) + 2$, откуда $m_2 = m_1 - 1$.

Аналогично получаем, что $m_3 = m_2 - 1$, $m_4 = m_3 - 1$ и т. д., т. е. на каждом последующем испытании шаг уменьшается на единицу. Тем самым последовательность шагов по номерам напряжений питания выглядит следующим образом: $m_1, m_1 - 1, m_1 - 2, \dots, 1$. Если очередной образец не выйдет из строя, то мы дойдём до нуля, и диапазон всех возможных напряжений будет исчерпан. Это позволяет записать для m_1 уравнение

$$\sum_{i=1}^{m_1} i = n. \quad (3)$$

Решением уравнения (3) является $m_1 = \sqrt{2n + 1/4} - 1/2$. Это число целое, если и только если число n треугольное, т. е. $n = \sum_{i=1}^k i = k(k + 1)/2$, так что $\sqrt{2n + 1/4} - 1/2 = k$. В этом случае при всех дальнейших ветвлениях число остающихся для исследования напряжений остаётся треугольным, и при каждом испытании при наличии двух образцов сохраняется точный баланс между двумя ветвями алгоритма. Если $n = k(k + 1)/2 + 1$, то уже на первом шаге алгоритма точного баланса между ветвями достигнуть не удастся, и трудоёмкость одной из ветвей вырастет на единицу, поэтому $l_2(k(k + 1)/2 + 1) = k + 1$.

В общем случае при $k(k + 1)/2 < n < (k + 1)(k + 2)$ из монотонности функции $l_2(n)$ следует, что равенство $l_2(n) = k + 1 = \lceil \sqrt{2n + 1/4} - 1/2 \rceil$ сохраняется, поэтому число испытаний равно $l_2(n) = \lceil \sqrt{2n + 1/4} - 1/2 \rceil$. Такой же номер напряжения следует выбирать при первом испытании, действуя аналогично при последующих испытаниях в случае невыхода образца из строя. Таким образом, доказано

Утверждение 1. В случае двух образцов и n напряжений в последовательности оптимальное число испытаний равно $\lceil \sqrt{2n + 1/4} - 1/2 \rceil$, и такой же шаг нужно делать по диапазону номеров исследуемых напряжений при каждом испытании с двумя образцами, когда текущий диапазон исследуемых напряжений питания состоит из n элементов.

3. Нахождение асимптотики $l_k(n)$ при $n \rightarrow \infty$

Точные формулы для $l_k(n)$, по-видимому, были бы слишком сложными. Однако, качественное представление о зависимости числа испытаний

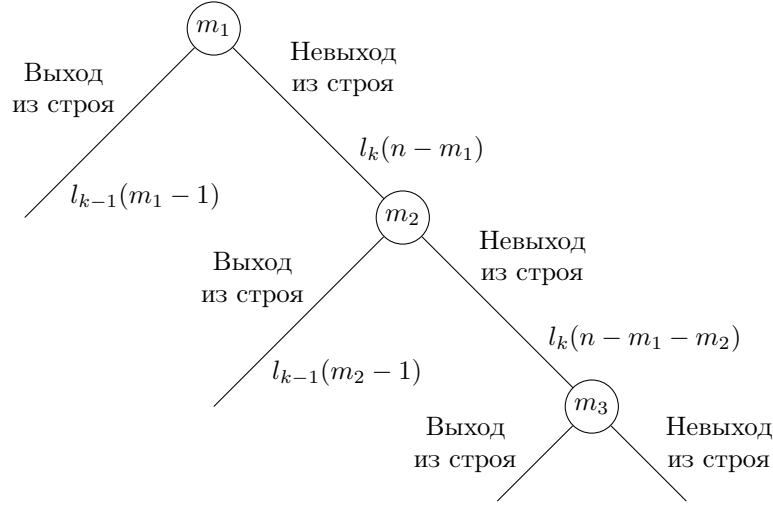


Рис. 3. Сбалансированность числа испытаний на ветвях алгоритма в общем случае

от числа имеющихся образцов можно получить, ограничившись асимптотическими оценками. Докажем, что число необходимых испытаний $l_k(n)$ при фиксированном числе k опытных образцов и числе возможных различных напряжений питания $n \rightarrow \infty$ асимптотически равно

$$l_k(n) \sim \sqrt[k]{k!n}, \quad n \rightarrow \infty. \quad (4)$$

Доказательство проведём индукцией по числу k опытных образцов. При $k = 1, 2$ утверждение, очевидно, справедливо. Пусть оно справедливо для k опытных образцов, т. е. имеет место (4). Докажем его справедливость для $k + 1$ опытных образцов, т. е. покажем, что

$$l_{k+1}(n) \sim \sqrt[k+1]{(k+1)!n}, \quad n \rightarrow \infty.$$

Как и прежде, будем опираться на равенство числа испытаний в случаях выхода из строя и невыхода из строя образца (рис. 3).

Рассмотрим два возможных пути по дереву испытаний на рис. 3. В первом случае на первом же шаге при подаче напряжения u_{m_1} образец выходит из строя. Во втором случае образец выдерживает напряжение u_{m_1} и выходит из строя на втором шаге при подаче напряжения $u_{m_1+m_2}$. Тогда, как следует из принципа сбалансированности, при оптимальном выборе m_1 и m_2 должно выполняться равенство

$$l_k(m_1 - 1) + 1 = l_k(m_2 - 1) + 2,$$

т. е.

$$l_k(m_1 - 1) - l_k(m_2 - 1) = 1.$$

Аналогичное соотношение справедливо и для всех последующих выборов напряжения

$$l_k(m_i - 1) - l_k(m_{i+1} - 1) = 1.$$

(В асимптотическом анализе возможными нарушениями баланса пренебрегаем.)

Последним является $l_{k+1}(n)$ -е испытание, в котором $m_{l_{k+1}(n)} = 1$, следовательно, $l(m_{l_{k+1}(n)} - 1) = 0$. Отсюда получаем

$$\begin{aligned} l_k(m_{l_{k+1}(n)} - 1) &= 0; \\ l_k(m_{l_{k+1}(n)-1} - 1) &= 1; \\ l_k(m_{l_{k+1}(n)-2} - 1) &= 2; \\ &\dots\dots\dots \\ l_k(m_1 - 1) &= l_{k+1}(n) - 1. \end{aligned} \quad (5)$$

Согласно (4) имеем $l_k(m_i - 1) \sim \sqrt[k]{k!(m_i - 1)}$, значит,

$$m_i \sim m_i - 1 \sim \frac{l_k^k(m_i - 1)}{k!}, \quad m_i \rightarrow \infty. \quad (6)$$

Далее имеем очевидное соотношение

$$\sum_{i=1}^{l_{k+1}(n)} m_i = n. \quad (7)$$

Из (5)–(7) выводим

$$\begin{aligned} n = \sum_{i=1}^{l_{k+1}(n)} m_i &\sim \sum_{i=1}^{l_{k+1}(n)} \frac{l_k^k(m_i - 1)}{k!} \sim \frac{1}{k!} \sum_{i=1}^{l_{k+1}(n)-1} i^k \\ &\sim \frac{1}{k!} \int_0^{l_{k+1}(n)} x^k dx = \frac{l_{k+1}^{k+1}(n)}{(k+1)!}, \quad n \rightarrow \infty. \end{aligned} \quad (8)$$

Из (8) получаем требуемое асимптотическое соотношение

$$l_{k+1}(n) \sim \sqrt[k+1]{(k+1)!n}, \quad n \rightarrow \infty.$$

Утверждение 2. При фиксированном числе k изначально имеющихся образцов и числе различных напряжений питания $n \rightarrow \infty$ оптимальное число испытаний асимптотически равно $\sqrt[k]{k!n}$, и асимптотически такой же шаг нужно делать по диапазону исследуемых напряжений, содержащему n элементов.

Заключение

В работе представлен эффективный алгоритм типа динамического программирования для составления оптимального плана испытаний при определении критического напряжения питания, необходимого в процессе планирования fault-атак. В случае двух имеющихся образцов получены точные формулы для проведения оптимального по времени измерения, делающие в этом случае ненужным использование алгоритма. В общем случае полученные асимптотические формулы дают качественное представления о том, как увеличение числа образцов уменьшает число необходимых испытаний. В основе всех полученных результатов лежит последовательно используемый принцип балансировки.

Полученные результаты могут применяться также в задачах поиска максимально допустимых значений других величин, использующихся в задачах индуцирования ошибок для проведения fault-атак, например, температуры, напряжённости электромагнитного поля и др. Кроме того, они могут оказаться востребованными и в иных, отличных от криптоанализа, сферах, когда требуется найти силу разрушающего воздействия на изделие, для которого имеются несколько образцов.

Финансирование работы

Исследование выполнено в рамках стратегического проекта DeepAnalytics программы «Приоритет 2030». Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Baksi A., Bhasin S., Breier J., Jap D., Saha D.** A survey on fault attacks on symmetric key cryptosystems // ACM Comput. Surveys. 2022. V. 55, No. 4. Paper ID 86. 34 p.
2. **Breier J., Jap D.** A survey of the state-of-the-art fault attacks // Proc. 14th Int. Symp. Integrated Circuits (Singapore, Dec. 10–12, 2014). Piscataway: IEEE, 2014. P. 152–155.
3. **Giraud C., Thiebeauld H.** A survey on fault attacks // Smart card research and advanced applications VI. IFIP 18th World Computer Congress. TC8/WG8.8 & TC11/WG11.2 6th Int. Conf. (Toulouse, France, Aug. 22–27, 2004) New York: Kluwer Acad. Publ., 2004. P. 159–176.
4. **Kim C. H., Quisquater J.-J.** Faults, injection methods, and fault attacks // IEEE Des. Test Comput. 2007. V. 24, No. 6. P. 544–545.
5. **Gangolli A., Mahmoud Q. H., Azim A.** A systematic review of fault injection attacks on IoT systems // Electronics. 2022. V. 11, No. 13. Paper ID 2023. 24 p.

6. **Piscitelli R., Bhasin S., Regazzoni F.** Fault attacks, injection techniques and tools for simulation // Hardware security and trust. Design and deployment of integrated circuits in a threatened environment. Cham: Springer, 2017. P. 27–47.
7. **Shepherd C., Markantonakis K., van Heijningen N., Aboulkassimi D., Gaine C., Heckmann T., Naccache D.** Physical fault injection and side-channel attacks on mobile devices: A comprehensive analysis // Comput. Secur. 2021. V. 111. Paper ID 102471. 31 p.
8. **Беллман Р.** Динамическое программирование. М.: Изд-во иностр. лит-ры, 1960. 400 с.
9. **Dreyfus S.** The art and theory of dynamic programming. New York: Acad. Press, 1977. 284 p. (Math. Sci. Eng.; V. 130).
10. **Lew A., Mauch H.** Dynamic programming: A computational tool. Heidelberg: Springer, 2010. 379 p.
11. **Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.** Introduction to algorithms. Cambridge, MA: MIT Press, 2022. 1312 p.

Зуев Юрий Анатольевич
Ключарёв Пётр Георгиевич

Статья поступила
4 июня 2023 г.
После доработки —
9 октября 2023 г.
Принята к публикации
22 декабря 2023 г.

A METHOD FOR OPTIMAL VALUE MEASUREMENT
OF SOME PARAMETERS OF FAULT ATTACKS
ON CRYPTOGRAPHIC ALGORITHMSYu. A. Zuev^a and P. G. Klyucharev^bBauman Moscow State Technical University,
5 Vtoraya Baumanskaya Street, 105005 Moscow, Russia
E-mail: ^a79851965730@yandex.ru, ^bpk.iu8@yandex.ru

Abstract. This paper is devoted to the construction of a time-optimal method for measuring the maximum permissible (critical) value of the supply voltage (as well as other parameters) of the device on which the cryptographic algorithm is performed. Knowledge of these values is necessary for successful conduct of error injection, as part of a fault attack. The method is based on dynamic programming. Bibliogr. 11.

Keywords: fault-attack, dynamic programming.

References

1. **A. Baksi, S. Bhasin, J. Breier, D. Jap, and D. Saha**, A survey on fault attacks on symmetric key cryptosystems, *ACM Comput. Surveys* **55** (4), Paper ID 86 (2022).
2. **J. Breier and D. Jap**, A survey of the state-of-the-art fault attacks, in *Proc. 14th Int. Symp. Integrated Circuits, Singapore, Dec. 10–12, 2014* (IEEE, Piscataway, 2014), pp. 152–155.
3. **C. Giraud and H. Thiebeault**, A survey on fault attacks, in *Smart Card Research and Advanced Applications VI* (IFIP 18th World Computer Congress. TC8/WG8.8 & TC11/WG11.2 6th Int. Conf., Toulouse, France, Aug. 22–27, 2004) (Kluwer Acad. Publ., New York, 2004), pp. 159–176.
4. **C. H. Kim and J.-J. Quisquater**, Faults, injection methods, and fault attacks, *IEEE Des. Test Comput.* **24** (6), 544–545 (2007).
5. **A. Gangolli, Q. H. Mahmoud, and A. Azim**, A systematic review of fault injection attacks on IoT systems, *Electronics* **11** (13), Paper ID 2023 (2022).

6. **R. Piscitelli, S. Bhasin, and F. Regazzoni**, Fault attacks, injection techniques and tools for simulation, in *Hardware Security and Trust. Design and Deployment of Integrated Circuits in a Threatened Environment* (Springer, Cham, 2017), pp. 27–47.
7. **C. Shepherd, K. Markantonakis, N. van Heijningen, D. Aboulkassimi, C. Gaine, T. Heckmann, and D. Naccache**, Physical fault injection and side-channel attacks on mobile devices: A comprehensive analysis, *Comput. Secur.* **111**, Paper ID 102471 (2021).
8. **R. Bellman**, *Dynamic Programming* (Princeton Univ. Press, Princeton, NJ, 1957; Izd. Inostr. Lit., Moscow, 1960 [Russian]).
9. **S. Dreyfus**, *The Art and Theory of Dynamic Programming* (Acad. Press, New York, 1977) (Math. Sci. Eng., Vol. 130).
10. **A. Lew and H. Mauch**, *Dynamic Programming: A Computational Tool* (Springer, Heidelberg, 2010).
11. **T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein**, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 2022).

Yury A. Zuev
Petr G. Klyucharev

Received June 4, 2023
Revised October 9, 2023
Accepted December 22, 2023

РАЗНОСТИ ПО МОДУЛЮ 2^n ДЛЯ ARX-ПРЕОБРАЗОВАНИЙ,
ВЕРОЯТНОСТЬ КОТОРЫХ БОЛЬШЕ $1/4$ А. С. Мокроусов^а, Н. А. Коломеец^бНовосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, РоссияE-mail: ^аsettingx@mail.ru, ^бkolomeec@math.nsc.ru

Аннотация. Рассматриваются разностные характеристики по модулю 2^n для преобразований $x \oplus y$ и $(x \oplus y) \lll r$, где $x, y \in \mathbb{Z}_2^n$ и $1 \leq r < n$. Эти характеристики применяются при разностном криптоанализе шифров архитектуры ARX, использующих в качестве операций только сложение по модулю 2^n , побитовое исключающее «или» (XOR, $\oplus$) и циклический сдвиг битов на r позиций ($\lll r$). Получена полная характеристика разностей, вероятность которых больше $1/4$. Возможными значениями вероятности при этом условии являются $1/3 + 4^{2-i}/6$ для обоих преобразований, где $i \in \{1, \dots, n\}$. Описаны разности, на которых достигается каждое из значений, и подсчитано их число. Найдено общее число разностей с приведёнными вероятностями: $48n - 68$ для $x \oplus y$ и $24n - 30$ для $(x \oplus y) \lll r$, где $n \geq 2$. Также дано сравнение разностных характеристик в контексте рассматриваемого ограничения на вероятность. Табл. 6, библиогр. 23.

Ключевые слова: ARX-схема, разностная характеристика, сложение по модулю, XOR, циклический сдвиг битов.

Введение

Одним из подходов к построению криптографических примитивов являются ARX-схемы, комбинирующие сложение по модулю 2^n ($\boxplus$), побитовое исключающее «или» (XOR, $\oplus$) и циклический сдвиг битов в векторе $x \in \mathbb{Z}_2^n$ на r позиций в сторону старших разрядов ($x \lll r$). Приведём как примеры FEAL [1], Skein [2], Speck [3], Salsa20 [4], ChaCha [5], CHAM [6, 7], Chaskey [8]. Важно, чтобы шифр был стойким к разностному криптоанализу [9], основа которого — изучение преобразования разностей алгоритмом шифрования. Разности могут рассматриваться, например, относительно $\oplus$, $\boxplus$, какой-нибудь другой операции или вообще быть смешанными (см. [10–14]). Будем рассматривать разность по модулю 2^n ,

т. е. относительно $\boxplus$. Соответствующая разностная характеристика adr^f для функции вида $f: (\mathbb{Z}_2^n)^k \rightarrow \mathbb{Z}_2^n$ определяется формулой

$$\begin{aligned} \text{adr}^f(\alpha_1, \dots, \alpha_k \rightarrow \alpha_{k+1}) &= \\ &= \Pr[f(x_1 \boxplus \alpha_1, \dots, x_k \boxplus \alpha_k) = \alpha_{k+1} \boxplus f(x_1, \dots, x_k)], \end{aligned}$$

где $x_1, \dots, x_k$ независимы и равномерно распределены на $\mathbb{Z}_2^n$. Будем называть $\alpha_1, \dots, \alpha_{k+1} \in \mathbb{Z}_2^n$ *набором разностей*, а значение характеристики $\text{adr}^f(\alpha_1, \dots, \alpha_k \rightarrow \alpha_{k+1})$ — его *вероятностью* относительно f . Как правило, для построения атак используются цепочки разностей, в которых каждый следующий элемент максимизирует значение разностной характеристики для соответствующей базовой операции в схеме шифра. Отметим, что использование характеристик для композиций может повысить достоверность метода, так как универсальный метод их вычисления по базовым операциям неточен.

В настоящей работе достигается полная классификация наборов разностей для характеристик $\text{adr}^\oplus$ и adr^{XR} , вероятность которых больше $\frac{1}{4}$. Это разностные характеристики для преобразований $x \oplus y$ и $(x \oplus y) \lll r$ соответственно, где $x, y \in \mathbb{Z}_2^n$ и $1 \leq r < n$. Отметим, что $\text{adr}^{\lll}$ исследовалась, например, в [15, 16], $\text{adr}^\oplus$ — в [17, 18], а adr^{XR} — в [19, 20]. Известны эффективные способы их вычисления, а в некоторых случаях — и их максимумов при определённых ограничениях. Ниже доказано, что значениями $\text{adr}^\oplus$ и adr^{XR} , превышающими $\frac{1}{4}$, являются вероятности $p_i = \frac{1}{3} + \frac{4^{2-i}}{6}$, где $i \in \{1, \dots, n\}$. Оказалось, что все наборы разностей, на которых данные значения достигаются, можно свести к одному набору для $\text{adr}^\oplus$ и трём наборам для adr^{XR} , используя известные преобразования разностей, сохраняющие значения характеристик. Для каждой из перечисленных вероятностей конструктивно подсчитано число наборов разностей, а также их число в целом: $48n - 68$ для $\text{adr}^\oplus$ и $24n - 30$ для adr^{XR} при $n \geq 2$. Полученные результаты показывают, что несмотря на некоторое сходство, например спектр значений, характеристика adr^{XR} устроена сложнее даже с учётом рассматриваемого ограничения. Наличие циклического сдвига вне зависимости от значения параметра r уменьшает число наборов разностей с высокой вероятностью почти в два раза.

Работа имеет следующую структуру. Базовые определения и утверждения даны в разд. 1. В разд. 2 приведена теорема 1, согласно которой $\{\text{adr}^\oplus(\alpha, \beta \rightarrow \gamma) \mid \alpha, \beta, \gamma \in \mathbb{Z}_2^n\} \cap (\frac{1}{4}, 1] = \{p_1, \dots, p_n\}$. Более того, если $\text{adr}^\oplus(\alpha, \beta \rightarrow \gamma) = p_i$, то тройку-набор разностей (α, β, γ) можно привести к тройке $(2^{n-i}, 2^{n-i}, 0)$ преобразованиями из утверждения 1. Их можно назвать симметриями аргументов: для $\text{adr}^\oplus$ это перестановки разностей в тройке, инверсия наиболее значимых битов у любых двух

элементов тройки и замена любых разностей обратными им по модулю 2^n . В разд. 3 описаны все пары, которые приводятся к данному виду (теорема 2 и табл. 1 и 2), а также подсчитано их число (следствия 2 и 3). Разд. 4 содержит дополнительные определения и утверждения, необходимые для работы с adr^{XR} , а в разд. 5 доказаны некоторые вспомогательные леммы. Наконец, в разд. 6 приводится теорема 3 о том, что спектры значений $\text{adr}^\oplus$ и adr^{XR} , превышающих $\frac{1}{4}$, совпадают. Более того, она показывает, что если $\text{adr}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) = p_i$, то (α, β, γ) преобразованиями из утверждения 7 приводится к одной из трёх троек. Отметим, что среди них присутствует тройка $(2^{n-i}, 2^{n-i}, 0)$ и набор, отличный от неё как минимум при $i = 2$ (утверждение 8). Тем не менее, из $3n$ битов их двоичного представления только два единичных, что, впрочем, меняется при применении симметрий аргументов. Следствие 4 и табл. 6 описывают все наборы разностей, приводимые к данным тройкам. Число наборов для каждого p_i и их общее число найдены в следствиях 5 и 6.

1. Определения

1.1. Двоичные векторы. Пусть $\mathbb{Z}_2^n$ — n -мерное линейное пространство двоичных векторов. Поскольку ARX-схемы комбинируют операции $\oplus$ и $\boxplus$, будем ассоциировать с вектором $x = (x_1, \dots, x_n) \in \mathbb{Z}_2^n$ целое число

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

т. е. биты слева соответствуют старшим разрядам числа, а справа — младшим. Под $x + y$, где $x, y \in \mathbb{Z}_2^n$, будем подразумевать сложение соответствующих x и y целых чисел по модулю 2^n , приводя остатки к множеству $\{0, \dots, 2^n - 1\}$. Аналогично $-x$ является противоположным к ассоциированному с x числу относительно сложения по модулю 2^n . Будем использовать как векторную, так и целочисленную нотацию. Например, вектор $(1, 0, \dots, 0)$ и число 2^{n-1} рассматриваем как один и тот же элемент $\mathbb{Z}_2^n$.

Будем говорить, что $x \preceq y$, если $x_i \leq y_i$ для всех $i \in \{1, \dots, n\}$. Обозначим через $(x, z) = (x_1, \dots, x_n, z_1, \dots, z_m)$ конкатенацию вектора $x \in \mathbb{Z}_2^n$ с вектором $z \in \mathbb{Z}_2^m$, разряды x в ней старшие. В случае $z \in \mathbb{Z}_2$ будем записывать её как xz , что в целочисленной нотации эквивалентно $xz = 2x + z$. Вес Хэмминга $\text{wt}(x)$ вектора x — число его ненулевых координат. Положим $\bar{x} = (x_1 \oplus 1, x_2 \oplus 1, \dots, x_n \oplus 1)$, а для $a \in \mathbb{Z}_2$ определим также $x^{[a]}$ следующим образом:

$$x^{[a]} = \begin{cases} x, & \text{если } a = 0, \\ \bar{x}, & \text{если } a = 1. \end{cases}$$

1.2. Разностные характеристики и восьмеричные слова. Рассматриваемые ниже разностные характеристики относительно сложения по модулю 2^n для преобразований $x \oplus y$ и $(x \oplus y) \lll r$, где $1 \leq r < n$, определяются формулами

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

Для удобства будем представлять тройки $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ в виде восьмеричного слова $\omega(\alpha, \beta, \gamma) \in \mathbb{O}^n$, определив

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

Введём и обратные преобразования: $\alpha(\omega) = \alpha$, $\beta(\omega) = \beta$ и $\gamma(\omega) = \gamma$, если $\omega = \omega(\alpha, \beta, \gamma)$. Аналогично двоичным векторам множество $\mathbb{O}$ будем рассматривать одновременно и как $\mathbb{Z}_2^3$, и как $\{0, \dots, 7\}$. В качестве аргументов разностных характеристик будем использовать как тройки векторов, так и слова: например $\text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) = \text{adp}^\oplus(\omega(\alpha, \beta, \gamma))$.

Записывать слова $\omega \in \mathbb{O}^n$ будем коротко: например 030^{n-2} вместо $(0, 3, 0, \dots, 0)$. В некоторых случаях запись может содержать *шаблонные символы* $\mathbf{e}$, $\mathbf{d}$ и $\mathbf{*}$, т. е. являться *шаблоном*. Например $\omega = \mathbf{*}0^{n-1}$ означает, что ω_1 может быть любым, а $\omega_2 = \omega_3 = \dots = \omega_n = 0$. Аналогично при $\omega = \mathbf{e}0^{n-1}$ требуем, чтобы $\omega_1 \in \{0, 3, 5, 6\}$, т. е. чётность $\text{wt}(\omega_1) = \text{wt}(\alpha_1, \beta_1, \gamma_1)$. И наоборот, $\omega_1 \in \{1, 2, 4, 7\}$ при $\omega = \mathbf{d}0^{n-1}$. Как и в примерах, будем использовать шаблонные символы только в старшем разряде. Отметим, что подобные сокращения не будут использоваться для двоичных векторов.

1.3. Преобразования слов и троек. Определим следующие преобразования на тройках $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$.

- $\mathcal{T}_{\alpha\beta}: (\alpha, \beta, \gamma) \mapsto (\beta, \alpha, \gamma)$. Аналогично задаются $\mathcal{T}_{\alpha\gamma}$ и $\mathcal{T}_{\beta\gamma}$, будем обозначать их через $\mathcal{T}_*$.

- $\mathcal{I}_{\alpha\beta}: (\alpha, \beta, \gamma) \mapsto (\alpha \oplus 2^{n-1}, \beta \oplus 2^{n-1}, \gamma)$. Аналогично задаются $\mathcal{I}_{\alpha\gamma}$ и $\mathcal{I}_{\beta\gamma}$, будем обозначать их через $\mathcal{I}_*$.

- $\mathcal{S}_\alpha: (\alpha, \beta, \gamma) \mapsto (-\alpha, \beta, \gamma)$. Аналогично задаются $\mathcal{S}_\beta$, $\mathcal{S}_\gamma$ и их композиции $\mathcal{S}_{\alpha\beta}$, $\mathcal{S}_{\alpha\gamma}$, $\mathcal{S}_{\beta\gamma}$ и $\mathcal{S}_{\alpha\beta\gamma}$. Будем обозначать их через $\mathcal{S}_*$.

Множество всех композиций преобразований вида $\mathcal{T}_*$, $\mathcal{I}_*$, $\mathcal{S}_*$ обозначим через $\mathcal{E}$, а через id — тождественное преобразование. Для $\mathcal{A} \subseteq \mathcal{E}$

$$\mathcal{A}(\alpha, \beta, \gamma) = \{\xi(\alpha, \beta, \gamma) \mid \xi \in \mathcal{A}\}.$$

Поскольку все «именованные» преобразования являются инволюциями, то $\mathcal{E}$ — группа. Отметим, что элементы этой группы сохраняют значение

$\text{adp}^\oplus$. Будем применять описанные выше преобразования как к тройкам (α, β, γ) , так и к соответствующим им словам $\omega(\alpha, \beta, \gamma) \in \mathbb{O}^n$.

1.4. Необходимые теоремы для работы с $\text{adp}^\oplus$. Для подсчёта $\text{adp}^\oplus$ будем использовать три факта. Во-первых, очевидное в случае $\alpha, \beta, \gamma \in \mathbb{Z}_2$ утверждение:

$$\text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) = \begin{cases} 1, & \text{если } \text{wt}(\alpha, \beta, \gamma) \text{ чётный,} \\ 0 & \text{иначе.} \end{cases} \quad (1)$$

Во-вторых, преобразования, описанные в [21, утверждения 1–3].

Утверждение 1 [21]. Для любых $\xi \in \mathcal{E}$, $\omega \in \mathbb{O}^n$ справедливо равенство $\text{adp}^\oplus(\omega) = \text{adp}^\oplus(\xi(\omega))$.

В-третьих, рекуррентные формулы, доказанные в [21, теорема 3]. Приведём соответствующее утверждение в более компактном виде, используя введённые ранее обозначения.

Утверждение 2 [21]. Пусть $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$ и $e \in \mathbb{Z}_2^3$. Тогда при чётном $\text{wt}(e)$ справедливо

$$\text{adp}^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) = \frac{1}{2^{\text{wt}(e)}} \sum_{q \in \mathbb{Z}_2^3, q \preceq e} \text{adp}^\oplus(\alpha^{[q_1]}, \beta^{[q_2]} \rightarrow \gamma^{[q_3]}).$$

Если $\text{wt}(e)$ нечётный, то $\text{adp}^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) = 0$.

Использовать его будем следующим образом. Во-первых,

$$\text{adp}^\oplus(\alpha 0, \beta 0 \rightarrow \gamma 0) = \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma). \quad (2)$$

Во-вторых, $e_1 \oplus e_2 \oplus e_3 = 1$ равносильно нечётности $\text{wt}(e)$ и влечёт

$$\text{adp}^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) = 0. \quad (3)$$

В третьих,

$$\begin{aligned} \text{adp}^\oplus(\alpha 0, \beta 1 \rightarrow \gamma 1) &= \frac{1}{4} \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) + \frac{1}{4} \text{adp}^\oplus(\alpha, \bar{\beta} \rightarrow \bar{\gamma}) + \\ &+ \frac{1}{4} \text{adp}^\oplus(\alpha, \bar{\beta} \rightarrow \gamma) + \frac{1}{4} \text{adp}^\oplus(\alpha, \beta \rightarrow \bar{\gamma}). \end{aligned} \quad (4)$$

Более того, как минимум два из четырёх слагаемых выше нулевые. Действительно,

$$\alpha_n \oplus \beta_n \oplus \gamma_n = \alpha_n \oplus \bar{\beta}_n \oplus \bar{\gamma}_n = \alpha_n \oplus \bar{\beta}_n \oplus \gamma_n \oplus 1 = \alpha_n \oplus \beta_n \oplus \bar{\gamma}_n \oplus 1,$$

что по предыдущему пункту зануляет либо два первых слагаемых, либо два последних. Также в силу утверждения 1 все оставшиеся случаи $\text{wt}(e) = 2$ можно получить из описанного выше перестановкой аргументов.

2. Значения $\text{adr}^\oplus$, превышающие $1/4$

В этом разделе рассматриваются наборы разностей $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ и их вероятности $\text{adr}^\oplus(\alpha, \beta \rightarrow \gamma)$ такие, что $\text{adr}^\oplus(\alpha, \beta \rightarrow \gamma) > \frac{1}{4}$. Начнём с определения последовательности, элементы которой и будут искомыми вероятностями.

Определение 1. Определим бесконечную последовательность рациональных чисел $\{p_i\}$ следующим образом:

$$p_i = \frac{1}{3} + \frac{4^{2-i}}{6}, \quad i \geq 1.$$

Сразу отметим её важные свойства.

Утверждение 3. Элементы последовательности $\{p_i\}$ обладают следующими свойствами:

- 1) $p_1 = 1, p_2 = \frac{1}{2}, p_3 = \frac{3}{8}, p_4 = \frac{11}{32}, p_5 = \frac{43}{128}$;
- 2) при $i \geq 2$ справедливо

$$p_i = \frac{1}{4} + \dots + \frac{1}{4^{i-1}} + \frac{1}{4^{i-1}};$$

- 3) $p_{i+1} = \frac{1}{4} + \frac{1}{4}p_i$ для всех $i \geq 1$;
- 4) последовательность p_i строго убывает и сходится к $\frac{1}{3}$, при этом $p_i > \frac{1}{3}$ для всех $i \geq 1$.

ДОКАЗАТЕЛЬСТВО. Значения p_1, p_2, p_3, p_4 и p_5 нетрудно вычислить по определению. Для доказательства п. 2 достаточно применить формулу суммы членов геометрической прогрессии. Действительно,

$$\frac{1}{4} + \dots + \frac{1}{4^{i-1}} + \frac{1}{4^{i-1}} = \frac{4}{3} \left(\frac{1}{4} - \frac{1}{4^i} \right) + \frac{1}{4^{i-1}} = \frac{1}{3} + \frac{3-1}{3 \cdot 4^{i-1}} = \frac{1}{3} + \frac{4^{2-i}}{6}.$$

П. 3 следует из второго, а п. 4 очевидно следует из определения. Утверждение 3 доказано.

Далее докажем несколько лемм, задающих ограничения на аргументы $\text{adr}^\oplus$, при которых значение этой характеристики больше $\frac{1}{4}$. Следующее утверждение можно найти в [17]. Приведём его с доказательством, демонстрируя удобство использования рекуррентных формул.

Лемма 1. Если $\omega \in \mathbb{O}^n$, то $\text{adr}^\oplus(\omega) = 1$ тогда и только тогда, когда $\omega = \mathbf{e}0^{n-1}$. Более того, не существует $\omega \in \mathbb{O}^n$ таких, что $\frac{1}{2} < \text{adr}^\oplus(\omega) < 1$.

ДОКАЗАТЕЛЬСТВО. Предположим, напротив, что $n > 1$, $\omega_n \neq 0$ и при этом $\text{adr}^\oplus(\omega) = 1$. Если $\text{wt}(\omega_n)$ нечётный, то $\text{adr}^\oplus(\omega) = 0$ в силу (3). Таким образом, остаётся только случай $\text{wt}(\omega_n) = 2$, и можно применить (4):

$$\text{adr}^\oplus(\omega) \leq \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 0 = \frac{1}{2};$$

противоречие. Значит, $n = 1$ или $\omega_n = 0$. Далее, применяя формулу (2), получаем, что только ω_1 может быть ненулевой. Равенство (1) завершает доказательство. Лемма 1 доказана.

Лемма 2. Пусть $1 \leq i \leq n$. Тогда $\text{adp}^\oplus(0^{i-1}30^{n-i}) = p_i$.

ДОКАЗАТЕЛЬСТВО. В силу (2) $\text{adp}^\oplus(0^{i-1}30^{n-i}) = \text{adp}^\oplus(0^{i-1}3)$. Далее воспользуемся индукцией по i .

БАЗА ИНДУКЦИИ $i = 1$: $\text{adp}^\oplus(3) = \text{adp}^\oplus(0) = 1 = p_1$ по (1).

ШАГ ИНДУКЦИИ: предположим, что утверждение справедливо для i , и докажем его для $i + 1$. Используя (4), получаем

$$\begin{aligned} \text{adp}^\oplus(0^i 3) &= \frac{1}{4} \text{adp}^\oplus(0^i) + \frac{1}{4} \text{adp}^\oplus(3^i) + \\ &\quad + \frac{1}{4} \text{adp}^\oplus(2^i) + \frac{1}{4} \text{adp}^\oplus(1^i) = \frac{1}{4} + \frac{1}{4} \text{adp}^\oplus(3^i), \end{aligned}$$

так как $\text{wt}(1) = \text{wt}(2) = 1$ (см. (3)). В то же время $\mathcal{S}_{\beta\gamma}(3^i) = 0^{i-1}3$. Таким образом, $\text{adp}^\oplus(3^i) = \text{adp}^\oplus(0^{i-1}3) = p_i$ по утверждению 1 и индукционному предположению. При этом утверждение 3 гарантирует, что $\frac{1}{4} + \frac{1}{4}p_i = p_{i+1}$. Лемма 2 доказана.

Лемма 3. Пусть $\omega \in \mathbb{O}^n$, $\omega \neq *0^{n-1}$ и $\text{adp}^\oplus(\omega) > 0$. Тогда существует $\xi \in \mathcal{E}$ такое, что $\xi(\omega) = 0u30^{n-m-2}$, где $u \in \mathbb{O}^m$ для некоторого $m \geq 0$.

ДОКАЗАТЕЛЬСТВО. Поскольку $\omega \neq *0^{n-1}$, хотя бы одно из α, β, γ не лежит в множестве $\{0, 2^{n-1}\}$. Без ограничения общности положим, что $\gamma \notin \{0, 2^{n-1}\}$.

Далее, используя преобразования $\mathcal{I}_*$, можно привести ω_1 к значению 0 или 1, не изменяя $\omega_2, \dots, \omega_n$. Если получим $\omega_1 = 1$, то дополнительное применение $\mathcal{S}_\gamma$ приведёт к $\omega_1 = 0$ в силу $\gamma \notin \{0, 2^{n-1}\}$.

Заметим, что $\gamma \notin \{0, 2^{n-1}\}$ гарантирует существование t такого, что $2 \leq t \leq n$, $\omega_t \neq 0$ и $\omega_{t+1} = \omega_{t+2} = \dots = \omega_n = 0$. По утверждению 2 нечётный $\text{wt}(\omega_t)$ означает, что $\text{adp}^\oplus(\omega) = 0$, поэтому остаётся только $\text{wt}(\omega_t) = 2$, т. е. $\omega_t \in \{3, 5, 6\}$. Применяя к ω преобразования $\mathcal{T}_*$, легко привести ω к виду $0u30^{n-t}$, где $u \in \mathbb{O}^{t-2}$. Лемма 3 доказана.

Лемма 4. Если $k \geq 0$ и $u \in \mathbb{O}^m$, то $\text{adp}^\oplus(0u30^k) > \frac{1}{4}$ тогда и только тогда, когда $u \in \{0^m, 3^m\}$.

ДОКАЗАТЕЛЬСТВО. Так как $\text{adp}^\oplus(0u30^k) = \text{adp}^\oplus(0u3)$ в силу (2), можно полагать, что $k = 0$. Пусть также $u = \omega(\alpha, \beta, \gamma)$ для некоторых $\alpha, \beta, \gamma \in \mathbb{Z}_2^m$. По рекуррентной формуле (4) получаем

$$\text{adp}^\oplus(0u3) = \frac{1}{4} \text{adp}^\oplus(0u) + \frac{1}{4} \text{adp}^\oplus(3u_1) + \frac{1}{4} \text{adp}^\oplus(2u_2) + \frac{1}{4} \text{adp}^\oplus(1u_3),$$

где $u_1 = \omega(\alpha, \bar{\beta}, \bar{\gamma})$, $u_2 = \omega(\alpha, \bar{\beta}, \gamma)$ и $u_3 = \omega(\alpha, \beta, \bar{\gamma})$. Рассмотрим четыре значения из формулы: $\text{adp}^\oplus(0u)$, $\text{adp}^\oplus(3u_1)$, $\text{adp}^\oplus(2u_2)$ и $\text{adp}^\oplus(1u_3)$. Как минимум два из них нулевые по (3) и (1), поэтому для $\text{adp}^\oplus(0u3) > \frac{1}{4}$ необходимо, чтобы хотя бы одно из оставшихся значений было равно 1. Действительно, в противном случае по лемме 1 имеем

$$\text{adp}^\oplus(0u3) \leq \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 0 = \frac{1}{4}.$$

Проверим все возможные варианты, пользуясь леммой 1:

- $\text{adp}^\oplus(0u) = 1$, тогда $u = 0^m$ и $3u_1 = 33^m$, т. е. $\text{adp}^\oplus(0u3) > \frac{1}{4}$;
- $\text{adp}^\oplus(3u_1) = 1$, тогда $u_1 = 0^m$, $u = 3^m$ и $0u = 03^m$, что влечёт $\text{adp}^\oplus(0u3) > \frac{1}{4}$;
- $\text{adp}^\oplus(1u_2) \neq 1$ и $\text{adp}^\oplus(2u_3) \neq 1$, поскольку $1, 2 \notin \{0, 3, 5, 6\}$.

Таким образом, подходит либо $u = 0^m$, либо $u = 3^m$. Лемма 4 доказана.

Лемма 5. Если $\omega \in \mathbb{O}^n$, то $\text{adp}^\oplus(\omega) > \frac{1}{4}$ тогда и только тогда, когда $\omega = \xi(0^{i-1}30^{n-i})$ для некоторых $i \in \{1, \dots, n\}$ и $\xi \in \mathcal{E}$.

ДОКАЗАТЕЛЬСТВО. Если $\omega = s0^{n-1}$, $s \in \mathbb{O}$, то $\text{adp}^\oplus(\omega) = 1$ при $s \in \{0, 3, 5, 6\}$ и $\text{adp}^\oplus(\omega) = 0$ иначе (см. лемму 1 и равенства (1), (2)), причём

$$\mathcal{I}_{\beta\gamma}(00^{n-1}) = \mathcal{I}_{\alpha\beta}(50^{n-1}) = \mathcal{I}_{\alpha\gamma}(60^{n-1}) = 30^{n-1},$$

что означает верность утверждения леммы для данного ω .

Теперь в силу леммы 3 можно полагать, что $\omega = 0u30^{n-m-2}$, где $u \in \mathbb{O}^m$ и $m \geq 0$. По лемме 4 неравенство $\text{adp}^\oplus(\omega) > \frac{1}{4}$ эквивалентно $u \in \{0^m, 3^m\}$, т. е. $\omega = 00^m30^{n-m-2}$ или $\omega = 03^m30^{n-m-2}$. Осталось заметить, что

$$00^m30^{n-m-2} \xrightarrow{\mathcal{I}_{\beta\gamma}} 30^m30^{n-m-2} \xrightarrow{\mathcal{S}_{\beta\gamma}} 03^m30^{n-m-2}.$$

Таким образом, $\omega = \xi(0^{m+1}30^{n-m-2})$ для некоторого $\xi \in \mathcal{E}$. Лемма 5 доказана.

Перейдём к теореме о значениях $\text{adp}^\oplus$, превышающих $\frac{1}{4}$.

Теорема 1. Пусть $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$ и $P = \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) > \frac{1}{4}$. Тогда $P \in \{p_1, \dots, p_n\}$. Более того, $P = p_i$, $1 \leq i \leq n$, тогда и только тогда, когда $(\alpha, \beta, \gamma) = \xi(0, 2^{n-i}, 2^{n-i})$ для некоторого $\xi \in \mathcal{E}$.

ДОКАЗАТЕЛЬСТВО. По лемме 5 любое $\omega \in \mathbb{O}^n$ с $\text{adp}^\oplus(\omega) > \frac{1}{4}$ можно привести к $0^{i-1}30^{n-i}$, $i \in \{1, \dots, n\}$, преобразованием из $\mathcal{E}$, что соответствует тройке $(0, 2^{n-i}, 2^{n-i})$. Осталось заметить, что $\text{adp}^\oplus(0^{i-1}30^{n-i}) = p_i$ (см. лемму 2) и применить утверждение 1. Теорема 1 доказана.

Замечание 1. Вместо набора $(0, 2^{n-i}, 2^{n-i})$ из теоремы 1 удобно использовать тройку $(2^{n-i}, 2^{n-i}, 0)$, полученную преобразованием $\mathcal{T}_{\alpha\gamma}$, так как adp^{XR} на ней будет также равна p_i (см. теорему 3).

Следствие 1. Не существует таких $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$, что

$$\frac{1}{4} < \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) \leq \frac{1}{3}.$$

ДОКАЗАТЕЛЬСТВО очевидно следует из теоремы 1 и утверждения 3. Следствие 1 доказано.

Известно [21], что $\max_{\beta, \gamma \in \mathbb{Z}_2^n} \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) = \text{adp}^\oplus(\alpha, \alpha \rightarrow 0)$. Таким образом, тройки $(2^{n-i}, 2^{n-i}, 0)$ максимизируют $\text{adp}^\oplus$ при фиксации первого или второго аргумента. Отметим, что задачу, связанную со значениями $\text{adp}^\oplus(\alpha, \alpha \rightarrow 0)$, можно найти в [22] (см. также первую олимпиаду [23]).

3. Разности, значение $\text{adp}^\oplus$ на которых больше $1/4$

Шаблоны восьмеричных слов являются наиболее удобным способом описания разностей для $\text{adp}^\oplus$ с вероятностью p_i .

Таблица 1

Шаблоны $\omega \in \mathbb{O}^n$ таких, что $\text{adp}^\oplus(\omega) = p_i$, $3 \leq i \leq n$

	id	$\mathcal{T}_{\alpha\beta}$	$\mathcal{T}_{\alpha\gamma}$
id	$\mathbf{e}0^{i-2}30^{n-i}$	$\mathbf{e}0^{i-2}50^{n-i}$	$\mathbf{e}0^{i-2}60^{n-i}$
$\mathcal{S}_\gamma$	$\mathbf{d}1^{i-2}30^{n-i}$	$\mathbf{d}1^{i-2}50^{n-i}$	$\mathbf{d}4^{i-2}60^{n-i}$
$\mathcal{S}_\beta$	$\mathbf{d}2^{i-2}30^{n-i}$	$\mathbf{d}4^{i-2}50^{n-i}$	$\mathbf{d}2^{i-2}60^{n-i}$
$\mathcal{S}_{\beta\gamma}$	$\mathbf{e}3^{i-1}0^{n-i}$	$\mathbf{e}5^{i-1}0^{n-i}$	$\mathbf{e}6^{i-1}0^{n-i}$

Таблица 2

Шаблоны $\omega \in \mathbb{O}^n$ таких, что $\text{adp}^\oplus(\omega) \in \{p_1, p_2\}$

$\text{adp}^\oplus(\omega)$	Шаблоны ω
p_1	$\mathbf{e}0^{n-1}$
p_2	$\mathbf{*}30^{n-2}, \mathbf{*}50^{n-2}, \mathbf{*}60^{n-2}$

Теорема 2. Если $\omega \in \mathbb{O}^n$ и $i \in \{1, \dots, n\}$, то $\text{adp}^\oplus(\omega) = p_i$ тогда и только тогда, когда ω удовлетворяет одному из шаблонов, представленных в табл. 1 при $i \geq 3$ или табл. 2 при $i \leq 2$.

ДОКАЗАТЕЛЬСТВО. Начнём с табл. 1, т. е. со случаев $i \geq 3$. Согласно теореме 1 любую тройку $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ преобразованиями из $\mathcal{E}$ можно привести к тройке $(0, 2^{n-i}, 2^{n-i})$, которая в восьмеричном виде представляется как $0^{i-1}30^{n-i}$.

Докажем, что слова, удовлетворяющие шаблонам из табл. 1, преобразованиями из $\mathcal{E}$ приводятся к слову $0^{i-1}30^{n-i}$. Применяя к нему $\mathcal{I}_*$,

получим любой символ $x \in \mathbb{O}$ с чётным $\text{wt}(x)$ в старшем разряде, т. е. слово $0^{i-1}30^{n-i}$ можно заменить шаблоном $\mathbf{e}0^{i-2}30^{n-i}$, и далее говорить о шаблонах.

Нетрудно видеть, что шаблон в строке, помеченной $\mathcal{S}$, и столбце, помеченном $\mathcal{T}$, равен $\mathcal{T}(\mathcal{S}(\mathbf{e}0^{i-2}30^{n-i}))$: достаточно заметить, что

$$-(2^{n-i} + c \cdot 2^{n-1}) = 2^n - 2^{n-i} - c \cdot 2^{n-1} = (c \oplus 1, 1, \dots, 1, \underbrace{0, \dots, 0}_{n-i}), \quad c \in \mathbb{Z}_2.$$

После преобразований $\mathcal{S}_\gamma$ или $\mathcal{S}_\beta$ старший символ $\mathbf{e}$ превратится в $\mathbf{d}$, так как инвертируем ровно одну позицию символа-элемента $\mathbb{Z}_2^3$. Кроме того, $\mathcal{T}_{\alpha\beta}(3) = 5$, $\mathcal{T}_{\alpha\gamma}(3) = 6$, $\mathcal{T}_{\alpha\beta}(1) = 1$, $\mathcal{T}_{\alpha\gamma}(1) = 4$, $\mathcal{T}_{\alpha\beta}(2) = 4$ и $\mathcal{T}_{\alpha\gamma}(2) = 2$.

Осталось доказать полноту множества описанных шаблонов. Действительно, применение $\mathcal{S}_*$ к любому шаблону не выводит за пределы столбца, применение $\mathcal{T}_*$ не выводит за пределы строки, а $\mathcal{I}_*$ уже учтено в самом шаблоне. Таким образом, множество описанных в таблице шаблонов замкнуто относительно $\mathcal{E}$. Очевидно также, что все описанные шаблоны порождают различные слова.

В заключение обратимся к табл. 2. Нетрудно видеть, что её элементы порождают замкнутые относительно $\mathcal{E}$ множества слов, среди которых есть 30^{n-1} и 030^{n-2} . Теорема 2 доказана.

Замечание 2. Табл. 1 можно использовать и для $i \in \{1, 2\}$, игнорируя шаблоны с 1^{i-2} и т. п. при отрицательных $i - 2$. Однако, в этих случаях шаблоны не обеспечат уникальности порождаемых слов.

Следствие 2. Если $C_i = |\{(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3 \mid \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) = p_i\}|$, $i \in \{1, \dots, n\}$, то $C_1 = 4$, $C_2 = 24$, $C_3 = C_4 = \dots = C_n = 48$.

ДОКАЗАТЕЛЬСТВО. Все шаблоны, описанные в табл. 1, при $i \geq 3$ порождают различные слова. Каждый шаблон порождает ровно 4 слова, таким образом, имеется $4 \cdot 4 \cdot 3 = 48$ различных слов, для которых $\text{adp}^\oplus$ равна p_i . Из табл. 2 очевидно, что $C_1 = 4$ и $C_2 = 8 \cdot 3 = 24$. Следствие 2 доказано.

Нетрудно подсчитать и общее число троек.

Следствие 3. При $n \geq 2$ имеется ровно $48n - 68$ троек $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ таких, что $\text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) > \frac{1}{4}$.

4. Необходимые утверждения для работы с adp^{XR}

Здесь приведём необходимые для работы с adp^{XR} понятия и утверждения из [20].

4.1. «Неполные» характеристики и симметрии их аргументов. Разобьём $\text{adp}^\oplus$ на неотрицательные «неполные» характеристики $\text{adp}_c^\oplus$ и $\text{adp}_{a,b}^\oplus$, $a, b, c \in \mathbb{Z}_2$:

$$\sum_{c \in \mathbb{Z}_2} \text{adp}_c^\oplus(\alpha, \beta \rightarrow \gamma) = \sum_{(a,b) \in \mathbb{Z}_2^2} \text{adp}_{a,b}^\oplus(\alpha, \beta \rightarrow \gamma) = \text{adp}^\oplus(\alpha, \beta \rightarrow \gamma). \quad (5)$$

Значения введённых величин при $\text{adp}^\oplus(\alpha, \beta \rightarrow \gamma) \neq 0$ для $\alpha, \beta, \gamma \in \mathbb{Z}_2$ (случай $n = 1$) даны в табл. 3, её можно найти в [20, табл. 1].

Таблица 3

Значения $\text{adp}_{a,b}^\oplus(\alpha, \beta \rightarrow \gamma)$ и $\text{adp}_c^\oplus(\alpha, \beta \rightarrow \gamma)$ для $\alpha, \beta, \gamma \in \mathbb{Z}_2$

$\alpha\beta\gamma$	$\omega(\alpha, \beta, \gamma)$	$\text{adp}_{0,0}^\oplus$	$\text{adp}_{0,1}^\oplus$	$\text{adp}_{1,0}^\oplus$	$\text{adp}_{1,1}^\oplus$	$\text{adp}_0^\oplus$	$\text{adp}_1^\oplus$
000	0	1	0	0	0	1	0
011	3	1/2	1/2	0	0	1/2	1/2
101	5	1/2	0	1/2	0	1/2	1/2
110	6	1/4	1/4	1/4	1/4	1	0

Для подсчёта значений $\text{adp}_{a,b}^\oplus$ и $\text{adp}_c^\oplus$ на аргументах из $\mathbb{Z}_2^{n+1}$ будем пользоваться рекуррентными формулами, аналогичными приведённым в утверждении 2 [20, теорема 3].

Утверждение 4 [20]. Если $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$, $e \in \mathbb{Z}_2^3$, $a, b, c \in \mathbb{Z}_2$, то при чётном $\text{wt}(e)$

$$\begin{aligned} \text{adp}_c^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) &= \frac{1}{2^{\text{wt}(e)}} \sum_{q \in \mathbb{Z}_2^3, q \preceq e} \text{adp}_{c \oplus q_3}^\oplus(\alpha^{[q_1]}, \beta^{[q_2]} \rightarrow \gamma^{[q_3]}), \\ \text{adp}_{a,b}^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) &= \frac{1}{2^{\text{wt}(e)}} \sum_{q \in \mathbb{Z}_2^3, q \preceq e} \text{adp}_{a \oplus q_1, b \oplus q_2}^\oplus(\alpha^{[q_1]}, \beta^{[q_2]} \rightarrow \gamma^{[q_3]}), \end{aligned}$$

при нечётном $\text{wt}(e)$

$$\text{adp}_c^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) = \text{adp}_{a,b}^\oplus(\alpha e_1, \beta e_2 \rightarrow \gamma e_3) = 0.$$

Аналогично $\text{adp}^\oplus$ подсчёт $\text{adp}_{a,b}^\oplus$ и $\text{adp}_c^\oplus$ можно упростить, используя симметрии аргументов [20, теорема 4].

Утверждение 5 [20]. Пусть $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$, $a, b, c \in \mathbb{Z}_2$ и $\omega = \omega(\alpha, \beta, \gamma)$. Тогда

- $\text{adp}_{a,b}^\oplus(\omega) = \text{adp}_{a,b}^\oplus(\mathcal{S}_\gamma(\omega))$,
- $\text{adp}_c^\oplus(\omega) = \text{adp}_c^\oplus(\mathcal{S}_\alpha(\omega)) = \text{adp}_c^\oplus(\mathcal{S}_\beta(\omega)) = \text{adp}_c^\oplus(\mathcal{S}_{\alpha\beta}(\omega))$,
- $\text{adp}_c^\oplus(\omega) = \text{adp}_c^\oplus(\mathcal{T}_{\alpha\beta}(\omega)) = \text{adp}_c^\oplus(\mathcal{I}_{\alpha\beta}(\omega))$.

Некоторые преобразования работают вместе с изменением параметров:

- $\text{adp}_{a,b}^{\oplus}(\omega) = \text{adp}_{b,a}^{\oplus}(\mathcal{T}_{\alpha\beta}(\omega))$,
- $\text{adp}_{a,b}^{\oplus}(\omega) = \text{adp}_{a,b}^{\oplus}(\mathcal{S}_{\alpha}(\omega))$ при $\alpha \neq 0$,
- $\text{adp}_{a,b}^{\oplus}(\omega) = \text{adp}_{a,b}^{\oplus}(\mathcal{S}_{\beta}(\omega))$ при $\beta \neq 0$,
- $\text{adp}_c^{\oplus}(\omega) = \text{adp}_c^{\oplus}(\mathcal{S}_{\gamma}(\omega))$ при $\gamma \neq 0$.

В контексте $\text{adp}_{a,b}^{\oplus}$ и $\text{adp}_c^{\oplus}$ будем обозначать их через $\mathcal{T}_{\alpha\beta}^!$, $\mathcal{S}_{\alpha}^!$, $\mathcal{S}_{\beta}^!$ и $\mathcal{S}_{\gamma}^!$ соответственно, подчёркивая, что они действуют на ω и тройке (a, b, c) .

4.2. Подсчёт adp^{XR} и симметрии её аргументов. Будем подсчитывать adp^{XR} способом, предложенным в [20, следствие 2], используя «неполные» характеристики.

Утверждение 6 [20]. Пусть $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, $a = \alpha_{n-r} \oplus \beta_{n-r} \oplus \gamma_{n-r}$, $a' = \alpha'_r \oplus \beta'_r \oplus \gamma'_r$, т. е. $a, a' \in \mathbb{Z}_2$. Тогда

$$\begin{aligned} \text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta) \xrightarrow{r} (\gamma, \gamma')) = \\ = \text{adp}_{a',0}^{\oplus}(\alpha, \beta \rightarrow \gamma^{[a]}) \text{adp}_a^{\oplus}(\alpha'^{[a']}, \beta' \rightarrow \gamma') + \\ + \text{adp}_{a',1}^{\oplus}(\alpha, \beta \rightarrow \gamma^{[a]}) \text{adp}_a^{\oplus}(\bar{\alpha}'^{[a']}, \bar{\beta}' \rightarrow \gamma'). \end{aligned}$$

Симметрий аргументов у adp^{XR} меньше, чем у $\text{adp}^{\oplus}$ [20, теорема 5]. Аналогично $\mathcal{E}$ обозначим через $\mathcal{E}'$ множество всевозможных композиций преобразований $\mathcal{T}_{\alpha\beta}$, $\mathcal{I}_{\alpha\beta}$ и $\mathcal{S}_{*}$. Для него верно

Утверждение 7 [20]. Если $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$, $\xi \in \mathcal{E}'$, $(\alpha', \beta', \gamma') = \xi(\alpha, \beta, \gamma)$, то $\text{adp}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) = \text{adp}^{\text{XR}}(\alpha', \beta' \xrightarrow{r} \gamma')$.

5. Вспомогательные леммы о значениях $\text{adp}_{a,b}^{\oplus}$ и $\text{adp}_c^{\oplus}$

Этот раздел содержит вспомогательные утверждения о «неполных» характеристиках. Пользуясь результатами разд. 2 и 3, далее будем работать только со словами, представленными в табл. 1. Начнём с подсчёта значений $\text{adp}_{a,b}^{\oplus}$ и $\text{adp}_c^{\oplus}$ с помощью рекуррентных формул.

Лемма 6. Если $a, b, c \in \mathbb{Z}_2$, $s \in \{0, 3, 5, 6\}$, $t \in \{3, 5, 6\}$, то

$$\begin{aligned} \text{adp}_{a,b}^{\oplus}(st^{k-1}0^{n-k}) &= \left(p_k - \frac{1}{4^{k-1}}\right) \text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s \oplus t) + \frac{1}{4^{k-1}} \text{adp}_{a,b}^{\oplus}(s), \\ \text{adp}_c^{\oplus}(st^{k-1}0^{n-k}) &= \left(p_k - \frac{1}{4^{k-1}}\right) \text{adp}_{c \oplus t_3}^{\oplus}(s \oplus t) + \frac{1}{4^{k-1}} \text{adp}_c^{\oplus}(s), \end{aligned}$$

определяющие значения $\text{adp}_{a,b}^{\oplus}$ и $\text{adp}_c^{\oplus}$ пары будем обозначать через

$$\begin{aligned} \rho_{a,b}(s, t) &= (\text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s \oplus t), \text{adp}_{a,b}^{\oplus}(s)), \\ \rho_c(s, t) &= (\text{adp}_{c \oplus t_3}^{\oplus}(s \oplus t), \text{adp}_c^{\oplus}(s)). \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Воспользуемся индукцией по k . Ясно, что при $k = 1$ равенства верны. Предположим, что для всех $k < n$ равенства верны. Докажем, что они верны и для $k + 1$. Пусть $s' = s \oplus t$. Согласно утверждению 4 (см. также пояснения к утверждению 2) имеем

$$\begin{aligned} \text{adp}_{a,b}^{\oplus}(st^k 0^{n-k-1}) &= \text{adp}_{a,b}^{\oplus}(st^k) = \frac{1}{4} \text{adp}_{a,b}^{\oplus}(st^{k-1}) + \\ &+ \frac{1}{4} \text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s' 0^{k-1}) = \frac{1}{4} \text{adp}_{a,b}^{\oplus}(st^{k-1}) + \frac{1}{4} \text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s'). \end{aligned}$$

Согласно утверждению 3 $p_k - \frac{1}{4^{k-1}} = \frac{1}{4} + \dots + \frac{1}{4^{k-1}}$. Воспользуемся индукционным предположением:

$$\begin{aligned} \frac{1}{4} \text{adp}_{a,b}^{\oplus}(st^{k-1}) &= \frac{1}{4} \left(p_k - \frac{1}{4^{k-1}} \right) \text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s') + \frac{1}{4^k} \text{adp}_{a,b}^{\oplus}(s) = \\ &= \left(\frac{1}{4^2} + \dots + \frac{1}{4^k} \right) \text{adp}_{a \oplus t_1, b \oplus t_2}^{\oplus}(s') + \frac{1}{4^k} \text{adp}_{a,b}^{\oplus}(s), \end{aligned}$$

при этом $\frac{1}{4} + \left(\frac{1}{4^2} + \dots + \frac{1}{4^k} \right) = p_{k+1} - \frac{1}{4^k}$, т. е. формула для $\text{adp}_{a,b}^{\oplus}$ доказана. Равенство для $\text{adp}_c^{\oplus}$ доказывается аналогично. Лемма 6 доказана.

Рассмотрим значения выражений из леммы 6, заменив пары $\rho_{a,b}$ и ρ_c неизвестными.

Лемма 7. Если $f_i(x, y) = \left(p_i - \frac{1}{4^{i-1}} \right) x + \frac{1}{4^{i-1}} y$, $i \geq 2$, и $x, y \in \mathbb{Q}$, то

- 1) $f_i(x, y) \leq \frac{1}{4}$ при $x, y \leq \frac{1}{2}$ для $i = 2$ и при $x \leq \frac{1}{2}$, $y \leq 1$ для $i \geq 3$;
- 2) $f_i(1, p_k) = p_{i+k-1}$, в частности,

$$f_i(1, 1) = p_i, \quad f_i\left(1, \frac{1}{2}\right) = p_{i+1}, \quad f_i\left(1, \frac{1}{4} + \frac{1}{4} p_k\right) = p_{i+k}.$$

- 3) $f_2\left(\frac{1}{2}, 1\right) = p_3$.

ДОКАЗАТЕЛЬСТВО. 1, 3) При $x, y \leq \frac{1}{2}$ справедливо $f_i(x, y) \leq \frac{1}{2} p_i$, что в силу $i \geq 2$ и утверждения 3 означает, что $f_i(x, y) \leq \frac{1}{4}$. Кроме того, $\frac{1}{2} \left(p_i - \frac{1}{4^{i-1}} \right) + \frac{1}{4^{i-1}} = \frac{1}{2} p_i + \frac{1}{2 \cdot 4^{i-1}}$, что равно $\frac{1}{6} + \frac{1}{12 \cdot 4^{i-2}} + \frac{1}{2 \cdot 4^{i-1}}$ по утверждению 3, или $\frac{1}{6} + \frac{5}{6 \cdot 4^{i-1}}$. Данное выражение равно p_3 при $i = 2$ и меньше $\frac{1}{4}$ при $i \geq 3$.

- 2) Имеем $f_i(1, p_k) = p_i - \frac{1}{4^{i-1}} + \frac{1}{4^{i-1}} p_k$, что в силу утверждения 3 равно

$$\frac{1}{4} + \dots + \frac{1}{4^{i-1}} + \frac{1}{4^i} + \dots + \frac{1}{4^{i+k-2}} + \frac{1}{4^{i+k-2}} = p_{i+k-1}.$$

Осталось заметить, что $1 = p_1$, $\frac{1}{2} = p_2$, $\frac{1}{4} + \frac{1}{4} p_k = p_{k+1}$. Лемма 7 доказана.

Опишем все слова, на которых значение $\text{adp}_c^\oplus$ больше $\frac{1}{4}$.

Лемма 8. *Справедливы равенства*

$$\begin{aligned}\text{adp}_0^\oplus(6^i 0^{m-i}) &= p_i, \quad 1 \leq i \leq m, \\ \text{adp}_1^\oplus(3^{i-1} 0^{m-i+1}) &= p_i, \quad 2 \leq i \leq m+1.\end{aligned}$$

Более того, если $\omega \in \mathbb{O}^m$ и $c \in \mathbb{Z}_2$ не приводятся к перечисленным словам композициями преобразований $\mathcal{T}_{\alpha\beta}$, $\mathcal{I}_{\alpha\beta}$, $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$ и $\mathcal{S}_\gamma^!$ (учитывая возможное изменение параметра c (см. утверждение 5)), то $\text{adp}_c^\oplus(\omega) \leq \frac{1}{4}$.

ДОКАЗАТЕЛЬСТВО. Если $\text{adp}_c^\oplus(\omega) > \frac{1}{4}$, то и $\text{adp}^\oplus(\omega) > \frac{1}{4}$. Это означает, что для доказательства леммы достаточно рассматривать только ω , удовлетворяющие одному из шаблонов, описанному в табл. 1 (или в табл. 2 при $m \in \{1, 2\}$ (см. теорему 2)). Более того, любое такое ω можно привести к виду $st^{i-1}0^{m-1}$, где $t \in \{3, 5, 6\}$, $s \in \{0, 3, 5, 6\}$ и $1 \leq i \leq m$. Действительно, каждый столбец табл. 1 содержит соответствующий данному слову шаблон. Все остальные элементы столбца получаются из заданного преобразованиями вида $\mathcal{S}_*$. Эти преобразования согласуются с указанными в условии, а также с утверждением 5 (при применении $\mathcal{S}_\gamma^!$ и ненулевом $\gamma(\omega)$ инвертируем c). Далее, в силу леммы 6 имеем

$$\text{adp}_c^\oplus(st^{i-1}0^{m-1}) = \left(p_i - \frac{1}{4^{i-1}}\right) \text{adp}_{c \oplus t_3}^\oplus(s \oplus t) + \frac{1}{4^{i-1}} \text{adp}_c^\oplus(s),$$

или $\text{adp}_c^\oplus(st^{i-1}0^{m-1}) = f_i(x, y)$ в обозначениях леммы 7, где $(x, y) = \rho_c(s, t)$ (см. лемму 6). Для начала рассмотрим случай $i \geq 2$.

Таблица 4

Значения $\rho_c(s, t)$, $s, t \in \{0, 3, 5, 6\}$, $t \neq 0$

st	03	05	06	33	35	36	53	55	56	63	65	66
ρ_0	$\frac{1}{2} \ 1$	$\frac{1}{2} \ 1$	$1 \ 1$	$0 \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$\frac{1}{2} \ 1$	$\frac{1}{2} \ 1$	$1 \ 1$
ρ_1	$\frac{1}{2} \ 0$	$\frac{1}{2} \ 0$	$0 \ 0$	$1 \ \frac{1}{2}$	$1 \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$1 \ \frac{1}{2}$	$1 \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$\frac{1}{2} \ 0$	$\frac{1}{2} \ 0$	$0 \ 0$
$\checkmark$	$\circ$	$\circ$	$*$	$\bullet$	$\bullet$		$\bullet$	$\bullet$		$\circ$	$\circ$	$*$

В табл. 4 с использованием табл. 3 подсчитаны все возможные значения (x, y) . Более того, одинаковые столбцы сгруппированы, группы отмечены $\circ$, $\bullet$ и $*$. Неотмеченные столбцы содержат пары, у которых $x, y \leq \frac{1}{2}$, т. е. для них $f_i(x, y) \leq \frac{1}{4}$ по лемме 7. При разборе случаев все такие пары также будем игнорировать (они присутствуют в каждом столбце).

СЛУЧАЙ 1. Группа $\circ$, т. е. $(x, y) = (\frac{1}{2}, 1)$, $c = 0$ и $st \in \{03, 05, 63, 65\}$. По лемме 7 $f_2(x, y) = p_3$ и $f_i(x, y) \leq \frac{1}{4}$ при $i \geq 3$. Таким образом, подходят

слова $03 = \mathcal{S}_\gamma^!(\mathcal{S}_\beta(33))$, $05 = \mathcal{T}_{\alpha\beta}(03)$, $63 = \mathcal{I}_{\alpha\beta}(03)$ и $65 = \mathcal{T}_{\alpha\beta}(63)$, т. е. все они приводятся к $\text{adp}_1^\oplus(3^i 0^{m-i}) = p_{i+1}$ при $i = 2$.

СЛУЧАЙ 2. Группа $\bullet$, т. е. $(x, y) = (1, \frac{1}{2})$, $c = 1$ и $st \in \{33, 35, 53, 55\}$. По лемме 7 $f_i(x, y) = p_{i+1}$. Все слова подходят и приводятся к виду $\text{adp}_c^\oplus(3^{j-1} 0^{m-j+1}) = p_j$ преобразованиями $\mathcal{T}_{\alpha\beta}$ и $\mathcal{I}_{\alpha\beta}$ для всех $j = i + 1$, т. е. при $3 \leq j \leq m + 1$.

СЛУЧАЙ 3. Группа $*$, т. е. $(x, y) = (1, 1)$, $c = 0$ и $st \in \{06, 66\}$. По лемме 7 $f_i(x, y) = p_i$. Все слова подходят и имеют вид $\text{adp}_0^\oplus(6^i 0^{m-i}) = p_i$, где $2 \leq i \leq m$, с точностью до преобразования $\mathcal{I}_{\alpha\beta}$.

Осталось рассмотреть случай $i = 1$, при котором $\text{adp}_c^\oplus(\omega) = \text{adp}_c^\oplus(s)$. Согласно табл. 3 при $c = 0$ подходят $s = 0$ и $s = 6$, что соответствует $\text{adp}_0^\oplus(6^i 0^{m-i}) = p_i$ при $i = 1$ с точностью до $\mathcal{I}_{\alpha\beta}$. При $c = 1$ подходят $s = 3$ и $s = 5$, что соответствует $\text{adp}_1^\oplus(3^{i-1} 0^{m-i+1}) = p_i$ при $i = 2$ с точностью до $\mathcal{T}_{\alpha\beta}$. Лемма 8 доказана.

Схожий результат нам потребуется и для $\text{adp}_{a,b}^\oplus$.

Лемма 9. Если $h_{a,b}^r(\omega) = \text{adp}_{a,b}^\oplus(\omega) + p_r \text{adp}_{\bar{a},\bar{b}}^\oplus(\omega)$, $a, b \in \mathbb{Z}_2$, $\omega \in \mathbb{O}^m$, то $h_{0,0}^r(0^m) = p_1$, $h_{1,1}^r(0^m) = p_r$ и для $1 \leq i \leq m$

$$h_{1,0}^r(5^i 0^{m-i}) = h_{1,0}^1(3^i 0^{m-i}) = h_{0,0}^1(6^i 0^{m-i}) = p_{i+1}, \quad h_{1,1}^r(6^i 0^{m-i}) = p_{i+r}.$$

Более того, если $\omega \in \mathbb{O}^m$ и $a, b \in \mathbb{Z}_2$ не приводят к перечисленным словам и параметрам композициями преобразований $\mathcal{T}_{\alpha\beta}^!$, $\mathcal{S}_\alpha^!$, $\mathcal{S}_\beta^!$ и $\mathcal{S}_\gamma$ (учитывая возможное изменение параметров a и b (см. утверждение 5)), то $h_{a,b}^r(\omega) \leq \frac{1}{4}$.

ДОКАЗАТЕЛЬСТВО. Как и в доказательстве леммы 8, рассматриваем $h_{a,b}^r(\omega) > \frac{1}{4}$, $a, b \in \mathbb{Z}_2$, так что $h_{a,b}^1(\omega) > \frac{1}{4}$, откуда $\text{adp}^\oplus(\omega) > \frac{1}{4}$. Следовательно, ω можно привести к виду $st^{i-1} 0^{m-1}$ композициями преобразований $\mathcal{S}_\alpha^!$, $\mathcal{S}_\beta^!$ и $\mathcal{S}_\gamma$, где $t \in \{3, 5, 6\}$, $s \in \{0, 3, 5, 6\}$ и $1 \leq i \leq m$ (см. теорему 2 и табл. 1). При этом в соответствии с преобразованиями $\mathcal{S}_\alpha^!$ и $\mathcal{S}_\beta^!$ параметры a и b у $h_{a,b}^r$ будут меняться так же, как и у соответствующих $\text{adp}_{a,b}^\oplus$. Таким образом, достаточно рассматривать значения $h_{a,b}^r(st^{i-1} 0^{m-1})$ при всех $a, b \in \mathbb{Z}_2$.

Воспользуемся леммой 6:

$$h_{a,b}^r(st^{i-1} 0^{m-1}) = \left(p_i - \frac{1}{4^{i-1}}\right) h_{a \oplus t_1, b \oplus t_2}^r(s \oplus t) + \frac{1}{4^{i-1}} h_{a,b}^r(s). \quad (6)$$

Чтобы применять лемму 7, рассмотрим случай $i \geq 2$, в котором используем обозначения

$$x = h_{a \oplus t_1, b \oplus t_2}^r(s \oplus t), \quad y = h_{a,b}^r(s),$$

тогда $f_i(x, y) = h_{a,b}^r(st^{i-1}0^{m-1})$. Напомним, что

$$\rho_{a,b}(s, t) = (\text{adp}_{a \oplus t_1, b \oplus t_2}^\oplus(s \oplus t), \text{adp}_{a,b}^\oplus(s)).$$

Таким образом, если $\rho_{a,b}(s, t) = (u, v)$ и $\rho_{\bar{a}, \bar{b}}(s, t) = (u', v')$, то

$$x = u + p_r u', \quad y = v + p_r v'.$$

Согласно табл. 3 $u, v, u', v' \in \{0, \frac{1}{4}, \frac{1}{2}, 1\}$. Более того, если $u \in \{\frac{1}{2}, 1\}$, то $u' = 0$. Данное утверждение справедливо и для пар (u', u) , (v, v') , (v', v) . Таким образом, если $u, u', v, v' < 1$, то $x, y \leq \frac{1}{2}$, что в силу леммы 7 даёт $h_{a,b}^r(st^{i-1}0^{m-1}) \leq \frac{1}{4}$. Далее будем рассматривать случаи, когда хотя бы одно из u, u', v и v' равно 1.

Таблица 5

Значения $\rho_{a,b}(s, t)$, $s, t \in \{0, 3, 5, 6\}$, $t \neq 0$

st	03	05	06	33	35	36	53	55	56	63	65	66
$\rho_{0,0}$	$\frac{1}{2} \ 1$	$\frac{1}{2} \ 1$	$\frac{1}{4} \ 1$	$0 \ \frac{1}{2}$	$\frac{1}{4} \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$\frac{1}{4} \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$0 \ \frac{1}{2}$	$0 \ \frac{1}{4}$	$0 \ \frac{1}{4}$	$0 \ \frac{1}{4}$
$\rho_{1,1}$	$0 \ 0$	$0 \ 0$	$\frac{1}{4} \ 0$	$0 \ 0$	$\frac{1}{4} \ 0$	$\frac{1}{2} \ 0$	$\frac{1}{4} \ 0$	$0 \ 0$	$\frac{1}{2} \ 0$	$\frac{1}{2} \ \frac{1}{4}$	$\frac{1}{2} \ \frac{1}{4}$	$1 \ \frac{1}{4}$
$1\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$									$\checkmark$
$\rho_{0,1}$	$\frac{1}{2} \ 0$	$0 \ 0$	$\frac{1}{4} \ 0$	$1 \ \frac{1}{2}$	$\frac{1}{4} \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$\frac{1}{4} \ 0$	$0 \ 0$	$0 \ 0$	$\frac{1}{2} \ \frac{1}{4}$	$0 \ \frac{1}{4}$	$0 \ \frac{1}{4}$
$\rho_{1,0}$	$0 \ 0$	$\frac{1}{2} \ 0$	$\frac{1}{4} \ 0$	$0 \ 0$	$\frac{1}{4} \ 0$	$0 \ 0$	$\frac{1}{4} \ \frac{1}{2}$	$1 \ \frac{1}{2}$	$\frac{1}{2} \ \frac{1}{2}$	$0 \ \frac{1}{4}$	$\frac{1}{2} \ \frac{1}{4}$	$0 \ \frac{1}{4}$
$1\checkmark$				$\checkmark$				$\checkmark$				

В табл. 5 с использованием табл. 3 подсчитаны все возможные значения (u, v) и (u', v') . Более того, в ней отмечены колонки, в которых хотя бы одно из u, u', v и v' равно 1. Далее по значениям пар из табл. 5 будем подсчитывать x, y и применять лемму 7, оставляя только значения $f_i(x, y) > \frac{1}{4}$.

СЛУЧАЙ 1: $(s, t) \in \{(0, 3), (0, 5)\}$, $a = b$, $(u, v), (u', v') \in \{(\frac{1}{2}, 1), (0, 0)\}$.

СЛУЧАЙ 1.1: $a = 0$. Тогда $x = \frac{1}{2}$, $y = 1$ и $f_i(x, y) = p_3$ только при $i = 2$, в остальных случаях она не превосходит $\frac{1}{4}$, т. е. подходят только слова $030^{m-2} = \mathcal{S}_\gamma(\mathcal{S}_\beta^!(\mathcal{T}_{\alpha\beta}^!(5^2 0^{m-2})))$ и $050^{m-2} = \mathcal{T}_{\alpha\beta}^!(030^{m-2})$, значение $h_{0,0}^r$ на которых равно p_3 .

СЛУЧАЙ 1.2: $a = 1$. Тогда $x = \frac{1}{2}p_r$, $y = p_r$. В этом случае условие $f_i(x, y) > \frac{1}{4}$ может быть выполнено только при $p_r > \frac{1}{2}$, или $r = 1$, что соответствует $x = \frac{1}{2}$, $y = 1$, как в случае 1.1, т. е. подходят только слова $030^{m-2} = \mathcal{S}_\gamma(\mathcal{S}_\beta^!(3^2 0^{m-2}))$ и $050^{m-2} = \mathcal{T}_{\alpha\beta}^!(030^{m-2})$, значение $h_{1,1}^1$ на которых равно p_3 .

СЛУЧАЙ 2: $(s, t) = (0, 6)$, $a = b$, $(u, v), (u', v') \in \{(\frac{1}{4}, 1), (\frac{1}{4}, 0)\}$.

СЛУЧАЙ 2.1: $a = 0$. Тогда $x = \frac{1}{4} + \frac{1}{4}p_r$, $y = 1$. Поскольку $x = p_{r+1}$ (см. утверждение 3), аналогично вышесказанному $f_i(x, y) = p_3$ только при $r = 1$ и $i = 2$, т. е. подходит только слово $060^{m-2} = \mathcal{S}_\alpha^!(\mathcal{S}_\beta^!(6^2 0^{m-2}))$, значение $h_{0,0}^1$ на котором равно p_3 .

СЛУЧАЙ 2.2: $a = 1$. Тогда $x = \frac{1}{4} + \frac{1}{4}p_r$, $y = p_r$. Аналогично случаю 2.1 $f_i(x, y) = p_3$ только при $r = 1$ и $i = 2$, т. е. подходит только слово $060^{m-2} = \mathcal{S}_\alpha^!(\mathcal{S}_\beta^!(6^2 0^{m-2}))$, значение $h_{1,1}^1$ на котором равно p_3 .

СЛУЧАЙ 3: $(s, t) = (6, 6)$, $a = b$, $(u, v), (u', v') \in \{(0, \frac{1}{4}), (1, \frac{1}{4})\}$.

СЛУЧАЙ 3.1: $a = 0$. Тогда $x = p_r$, $y = \frac{1}{4} + \frac{1}{4}p_r = p_{r+1}$. По лемме 7 $f_i(x, y) > \frac{1}{4}$ только при $r = 1$, причём $f_i(x, y) = p_{i+r}$, т. е. подходит только слово $6^i 0^{m-i}$, значение $h_{0,0}^1$ на котором равно p_{i+1} .

СЛУЧАЙ 3.2: $a = 1$. Тогда $x = 1$, $y = \frac{1}{4} + \frac{1}{4}p_r = p_{r+1}$. По лемме 7 $f_i(x, y) > \frac{1}{4}$, причём $f_i(x, y) = p_{i+r}$, т. е. подходит слово $6^i 0^{m-i}$, значение $h_{1,1}^r$ на котором равно p_{i+r} .

СЛУЧАЙ 4: $(s, t) = (3, 3)$, $a \neq b$, $(u, v), (u', v') \in \{(1, \frac{1}{2}), (0, 0)\}$.

СЛУЧАЙ 4.1: $a = 0$. Тогда $x = 1$, $y = \frac{1}{2}$. По лемме 7 $f_i(x, y) > \frac{1}{4}$, причём $f_i(x, y) = p_{i+1}$, т. е. подходит только слово $3^i 0^{m-i} = \mathcal{T}_{\alpha\beta}^!(5^i 0^{m-i})$, значение $h_{0,1}^r$ на котором равно p_{i+1} .

СЛУЧАЙ 4.2: $a = 1$. Тогда $x = p_r$, $y = \frac{1}{2}p_r$. По лемме 7 $f_i(x, y) > \frac{1}{4}$ только при $r = 1$, причём $f_i(x, y) = p_{i+1}$, т. е. подходит только слово $3^i 0^{m-i}$, значение $h_{1,0}^1$ на котором равно p_{i+1} .

СЛУЧАЙ 5: $(s, t) = (5, 5)$, $a \neq b$, $(u, v), (u', v') \in \{(0, 0), (1, \frac{1}{2})\}$. Этот случай симметричен предыдущему относительно перестановки a и b , т. е. подходят только слова $5^i 0^{m-i} = \mathcal{T}_{\alpha\beta}^!(3^i 0^{m-i})$, значение $h_{0,1}^1$ на котором равно p_{i+1} , и $5^i 0^{m-i}$, значение $h_{1,0}^r$ на котором также равно p_{i+1} .

Тем самым для $i \geq 2$ получили все перечисленные в условии слова, за исключением 0^m . Разберём случай $i = 1$, т. е. найдём значение $h_{a,b}^r(s 0^{m-1})$, которое равно $h_{a,b}^r(s) = \text{adp}_{a,b}^\oplus(s) + p_r \text{adp}_{\bar{a},\bar{b}}^\oplus(s)$ согласно (6). Для этого достаточно воспользоваться табл. 3.

При $s = 0$ имеем $h_{0,0}^r(0^m) = 1$, $h_{0,0}^r(0^m) = p_r$ и $h_{1,0}^r(0^m) = h_{0,1}^r(0^m) = 0$. При $s = 6$ при всех a, b получаем $h_{a,b}^r(6 0^{m-1}) = \frac{1}{4} + \frac{1}{4}p_r$, что равно p_{r+1} по утверждению 3. Однако $\mathcal{S}_\alpha^!(6 0^{m-1}) = \mathcal{S}_\beta^!(6 0^{m-1}) = 6^i 0^{m-1}$ при $i = 1$. Кроме того, $h_{0,0}^1(6^i 0^{m-i})$ лежит в этом случае при $i = 1$.

При $s = 3$ имеем $h_{0,0}^r(3 0^{m-1}) = h_{0,1}^r(3 0^{m-1}) = \frac{1}{2} = p_2$ и $h_{1,0}^r(3 0^{m-1}) = h_{1,1}^r(3 0^{m-1}) = \frac{1}{2}p_r$, что равно p_2 только при $r = 1$ и не превосходит $\frac{1}{4}$ при других r . Однако $\mathcal{T}_{\alpha\beta}^!(3 0^{m-1}) = 5 0^{m-1}$ и $\mathcal{S}_\beta^!(3 0^{m-1}) = 3 0^{m-1}$, т. е. случаи учтены в $h_{1,0}^r(5^i 0^{m-i}) = p_{i+1}$ при $i = 1$. Оставшаяся пара аналогичными рассуждениями приводится к $h_{1,0}^1(3^i 0^{m-i}) = p_{i+1}$ при $i = 1$. Случай $s = 5$

можно не рассматривать, поскольку он соответствует применению $\mathcal{T}_{\alpha\beta}^1$ к $s = 3$. Лемма 9 доказана.

6. Значения adp^{XR} , превышающие $1/4$

Перейдём к описанию всех слов, значение adp^{XR} на которых больше $\frac{1}{4}$. Отметим, что при заданном условии adp^{XR} принимает в точности те же значения, что и $\text{adp}^\oplus$.

Теорема 3. Пусть $\alpha, \beta, \gamma \in \mathbb{Z}_2^n$ и $P = \text{adp}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) > \frac{1}{4}$, где $1 \leq r < n$. Тогда $P \in \{p_1, \dots, p_n\}$. Более того, $P = p_i$, $1 \leq i \leq n$, тогда и только тогда, когда тройку (α, β, γ) преобразованиями из $\mathcal{E}'$ можно привести к одной из следующих:

- 1) $(2^{n-i}, 2^{n-i}, 0)$ при $1 \leq i \leq n$,
- 2) $(0, 2^{n-i+1}, 2^{r-i+1})$ при $2 \leq i \leq r+1$,
- 3) $(2^{n-r-i+1}, 0, 2^{n-i+1})$ при $2 \leq i \leq n-r+1$.

ДОКАЗАТЕЛЬСТВО. Для подсчёта adp^{XR} используем утверждение 6, поэтому переобозначим вход, разделим его на две части: пусть теперь $\alpha, \beta, \gamma \in \mathbb{Z}_2^{n-r}$, $\alpha', \beta', \gamma' \in \mathbb{Z}_2^r$, $a = \alpha_{n-r} \oplus \beta_{n-r} \oplus \gamma_{n-r}$ и $a' = \alpha'_r \oplus \beta'_r \oplus \gamma'_r$. Тогда в силу утверждения 6

$$\begin{aligned} \text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta) \xrightarrow{r} (\gamma, \gamma')) = \\ = \text{adp}_{a',0}^\oplus(\alpha, \beta \rightarrow \gamma^{[a]}) \text{adp}_a^\oplus(\alpha'^{[a']}, \beta' \rightarrow \gamma') + \\ + \text{adp}_{a',1}^\oplus(\alpha, \beta \rightarrow \gamma^{[a]}) \text{adp}_a^\oplus(\bar{\alpha}'^{[a']}, \bar{\beta}' \rightarrow \gamma'). \end{aligned}$$

Аргументами теперь являются (α', α) , (β', β) и (γ, γ') , введём $\omega'_{u,v} = \omega(\alpha'^{[u]}, \beta'^{[v]}, \gamma')$ для произвольных $u, v \in \mathbb{Z}_2$ и $\omega_a = \omega(\alpha, \beta, \gamma^{[a]})$, а также $\omega' = \omega'_{0,0}$ и $\omega = \omega_0$. Тогда аргументы из формулировки теоремы, на которых adp^{XR} равно p_i , можно описать следующими парами:

$$(\omega', \omega) = (6^i 0^{r-i}, 0^{n-r}), \quad 1 \leq i \leq r, \quad (7)$$

$$(\omega', \omega) = (6^r, 6^{i-r} 0^{n-i}), \quad r+1 \leq i \leq n, \quad (8)$$

$$(\omega', \omega) = (2^{i-2} 30^{r-i+1}, 0^{n-r}), \quad 2 \leq i \leq r+1, \quad (9)$$

$$(\omega', \omega) = (4^r, 5^{i-1} 0^{n-r-i+1}), \quad 2 \leq i \leq n-r+1. \quad (10)$$

Здесь из-за деления входа на две части тройка $(2^{n-i}, 2^{n-i}, 0)$ распадается на пары (7) и (8), для удобства она приведена к виду $(-2^{n-i}, -2^{n-i}, 0)$ преобразованием $\mathcal{S}_{\alpha\beta}$. Аналогично вторая тройка преобразованием $\mathcal{S}_\beta$ приведена к виду $(0, -2^{n-i+1}, 2^{r-i+1})$, а третья тройка преобразованием $\mathcal{S}_{\alpha\gamma}$ — к виду $(-2^{n-r-i+1}, 0, -2^{n-i+1})$, они соответствуют (9) и (10).

Напомним, что

$$-2^{n-i} = (\underbrace{1, \dots, 1}_i, \underbrace{0, \dots, 0}_{n-i}).$$

Из формулы для adp^{XR} получаем неравенство

$$\begin{aligned} \text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta)) \xrightarrow{r} (\gamma, \gamma') &\leq \\ &\leq (\text{adp}_{a',0}^{\oplus}(\omega_a) + \text{adp}_{a',1}^{\oplus}(\omega_a)) \max \{ \text{adp}_a^{\oplus}(\omega'_{a',0}), \text{adp}_a^{\oplus}(\omega'_{a',1}) \}. \end{aligned}$$

Таким образом, если оба множителя не превосходят $\frac{1}{2}$, то имеет неравенство $\text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta)) \xrightarrow{r} (\gamma, \gamma') \leq \frac{1}{4}$. Значит, как минимум один из них больше $\frac{1}{2} = p_2$, что в силу лемм 8 и 9 требует равенство p_1 , т. е. единице.

СЛУЧАЙ 1: $\text{adp}_{a',0}^{\oplus}(\omega_a) + \text{adp}_{a',1}^{\oplus}(\omega_a) = 1$. В этом случае $\text{adp}^{\oplus}(\omega_a) = 1$, т. е. $\omega_a = s0^{n-r-1}$ в силу теоремы 2, где $s \in \{0, 3, 5, 6\}$. Поскольку $\text{adp}_{a',0}^{\oplus}(s0^{n-r-1}) = \text{adp}_{a',0}^{\oplus}(s)$ и аналогично $\text{adp}_{a',1}^{\oplus}(s0^{n-r-1}) = \text{adp}_{a',1}^{\oplus}(s)$ в силу утверждения 4, можно воспользоваться табл. 3. Из неё следует, что $a' = 0$, $\text{adp}_{a',0}^{\oplus}(\omega_a) = 1$, $\text{adp}_{a',1}^{\oplus}(\omega_a) = 0$ и $\omega_a = 0^{n-r}$. Таким образом,

$$\text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta)) \xrightarrow{r} (\gamma, \gamma') = \text{adp}_a^{\oplus}(\alpha', \beta' \rightarrow \gamma') = \text{adp}_a^{\oplus}(\omega') > \frac{1}{4}.$$

Лемма 8 гарантирует, что $\text{adp}_a^{\oplus}(\omega') \in \{p_1, \dots, p_{r+1}\}$, и описывает следующие два подслучая.

СЛУЧАЙ 1.1: $\text{adp}_0^{\oplus}(6^i 0^{r-i}) = p_i$ при $1 \leq i \leq r$, т. е. $a = 0$, что влечёт $\omega = 0^{n-r}$ и соответствует (7).

СЛУЧАЙ 1.2: $\text{adp}_1^{\oplus}(3^{i-1} 0^{r-i+1}) = p_i$ при $2 \leq i \leq r+1$, т. е. $a = 1$, что влечёт $\omega = 1^{n-r}$. Заметим, что

$$\mathcal{S}_\gamma(\omega', \omega) = \mathcal{S}_\gamma(3^{i-1} 0^{r-i+1}, 1^{n-r}) = (2^{i-2} 3 0^{r-i+1}, 0^{n-r})$$

(см. формулу ниже). Это соответствует (9).

Также лемма 8 гарантирует, что достаточно рассматривать только эти подслучаи. Действительно, $\omega_a = 0^{n-r}$, значит, что $\alpha = \beta = 0$. Поскольку $-(\alpha', 0) = (-\alpha', 0)$ и аналогично для β , получаем

$$\mathcal{S}_\alpha(\omega', \omega_a) = (\mathcal{S}_\alpha(\omega'), \omega_a), \quad \mathcal{S}_\beta(\omega', \omega_a) = (\mathcal{S}_\beta(\omega'), \omega_a).$$

Также если $\gamma' \neq 0$, то справедливо $-(\gamma, \gamma') = (\overline{\gamma}, -\gamma')$, поэтому

$$\mathcal{S}_\gamma(\omega', \omega_a) = (\mathcal{S}_\gamma(\omega'), \omega_a) \quad \text{при } \gamma(\omega') \neq 0,$$

что полностью соответствует $\mathcal{S}_\gamma^!$ для ω' и a : $\text{adp}_a^{\oplus}(\omega') = \text{adp}_a^{\oplus}(\mathcal{S}_\gamma^!(\omega'))$ (см. утверждение 5). Если же $\gamma' = 0$, то $\mathcal{S}_\gamma^!(\omega') = \omega'$, т. е. ни a , ни ω' не изменяются. Таким образом, преобразования $\mathcal{T}_{\alpha\beta}$, $\mathcal{I}_{\alpha\beta}$, $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$ и $\mathcal{S}_\gamma^!$ на ω' полностью соответствуют преобразованиям $\mathcal{T}_{\alpha\beta}$, $\mathcal{I}_{\alpha\beta}$, $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$ и $\mathcal{S}_\gamma$ на паре (ω', ω) , которая является входом adp^{XR} .

СЛУЧАЙ 2: $\max \{ \text{adp}_a^\oplus(\omega'_{a',0}), \text{adp}_a^\oplus(\omega'_{a',1}) \} = 1$. Лемма 8 гарантирует, что либо $\omega'_{a',0}$, либо $\omega'_{a',1}$ приводится к 60^{r-1} , при этом $a = 0$. Заметим, что $\mathcal{I}_{\alpha\beta}(60^{r-1}) = 0^r$, а любое из преобразований $\mathcal{S}_*$ и $\mathcal{T}_{\alpha\beta}$ оставляет слова 0^r и 60^{r-1} на месте, т. е. одно из $\omega'_{a',0}$ и $\omega'_{a',1}$ принадлежит множеству $\{0^r, 60^{r-1}\}$. Более того, можно применить преобразования $\mathcal{I}_{\alpha\beta}$, так как эти разряды старшие для исходных аргументов adp^{XR} , поэтому можно рассматривать только случаи $\omega'_{a',0} = 0^r$ или $\omega'_{a',1} = 0^r$. Таким образом, данный случай сводится к рассмотрению значения

$$\text{adp}^{\text{XR}}((\alpha', \alpha), (\beta', \beta) \xrightarrow{r} (\gamma, \gamma')) = \text{adp}_{u,v}^\oplus(\omega) + p_r \text{adp}_{\bar{u},\bar{v}}^\oplus(\omega) = h_{u,v}^r(\omega) > \frac{1}{4}$$

в обозначениях леммы 9, где $u = \alpha'_1$, $v = \beta'_1$, т. е. $\alpha' = (u, \dots, u)$, $\beta' = (v, \dots, v)$ и $\gamma' = 0$, что влечёт $\omega'_{u,v} = 0^r$ и $\omega' = x^r$, где $x = 4u + 2v$. Лемма 9 также гарантирует, что $h_{u,v}^r(\omega) \in \{p_1, \dots, p_n\}$, и описывает следующие три подслучая.

СЛУЧАЙ 2.1. Во-первых, $h_{1,0}^r(5^i 0^{m-i}) = p_{i+1}$, так что получаем пару $\omega' = 4^r$ и $\omega = 5^i 0^{m-i}$, что соответствует (10) с учётом уменьшения i на 1.

Во-вторых, $h_{1,0}^1(3^i 0^{m-i}) = p_{i+1}$ при $r = 1$, так что получаем $\omega' = 4$ и $\omega = 3^i 0^{n-i-1}$. Однако эта пара приводится к предыдущей преобразованиями $\mathcal{T}_{\alpha\beta}$ и $\mathcal{I}_{\alpha\beta}$.

В-третьих, $h_{0,0}^1(6^i 0^{m-i}) = p_{i+1}$ также при $r = 1$, так что $\omega' = 0$ и $\omega = 6^i 0^{n-i-1}$. Эта пара приводится к виду (8) преобразованием $\mathcal{I}_{\alpha\beta}$.

СЛУЧАЙ 2.2: $h_{1,1}^r(6^i 0^{m-i}) = p_{i+r}$, что даёт $\omega' = 6^r$, $\omega = 6^i 0^{m-i}$ и соответствует (8).

СЛУЧАЙ 2.3: $\omega = 0^m$, что соответствует значению $\text{adp}^\oplus$, равному 1, которое рассмотрено в случае 1.

Аналогично случаю 1 и лемме 8 лемма 9 гарантирует, что достаточно рассмотреть только перечисленные подслучаи, поскольку

$$\mathcal{S}_\alpha(\omega'_{u,v}, \omega) = (\omega'_{u,v}, \mathcal{S}_\alpha^!(\omega)), \quad \mathcal{S}_\beta(\omega'_{u,v}, \omega) = (\omega'_{u,\bar{v}}, \mathcal{S}_\beta^!(\omega))$$

при $\alpha \neq 0$ и $\beta \neq 0$ соответственно, а также $\mathcal{S}_\gamma(\omega'_{u,v}, \omega) = (\omega'_{u,v}, \mathcal{S}_\gamma(\omega))$ из-за $\gamma' = 0$. Очевидно также, что $\mathcal{T}_{\alpha\beta}(\omega'_{u,v}, \omega) = (\omega'_{v,u}, \mathcal{T}_{\alpha\beta}^!(\omega))$. Таким образом, преобразования $\mathcal{T}_{\alpha\beta}^!$, $\mathcal{S}_\alpha^!$, $\mathcal{S}_\beta^!$ и $\mathcal{S}_\gamma$ на ω для $h_{u,v}^r$ (т. е. для $\text{adp}_{u,v}^\oplus$) полностью соответствуют преобразованиям $\mathcal{T}_{\alpha\beta}$, $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$ и $\mathcal{S}_\gamma$ на паре (ω', ω) , которая является входом adp^{XR} . Теорема 3 доказана.

Утверждение 8. *Тройки из условия теоремы 3 с учётом ограничений на i не приводятся друг к другу преобразованиями из $\mathcal{E}'$. Более того, они не приводятся друг к другу преобразованиями из $\mathcal{E}$, за исключением случая $n = 2r$, когда тройка из п. 2 приводится к тройке из п. 3.*

ДОКАЗАТЕЛЬСТВО. Начнём со второй части. Единственный ненулевой элемент первой тройки 2^{n-i} не приводится ни отрицанием, ни прибавлением 2^{n-1} к отличной степени двойки, в том числе и к 2^{n-i+1} . Таким образом, первую тройку нельзя привести преобразованиями $\mathcal{T}_*$, $\mathcal{I}_*$ и $\mathcal{S}_*$ ни ко второй, ни к третьей. Далее, у второй и третьей элементы 0 и 2^{n-i+1} совпадают, и оставшимися являются 2^{r-i+1} и $2^{n-r-i+1}$ соответственно, причём оба отличны от 2^{n-i+1} , единственного ненулевого элемента. Тем самым единственный вариант приведения троек $-r-i+1 = n-r-i+1$, т. е. $n = 2r$.

Осталось доказать, что вторая и третья тройки не приводятся друг к другу композициями преобразований $\mathcal{T}_{\alpha\beta}$, $\mathcal{I}_{\alpha\beta}$ и $\mathcal{S}_*$. Действительно, из них только $\mathcal{S}_\gamma$ может изменять γ , при этом 2^{r-i+1} и 2^{n-i+1} — различные степени двойки. Утверждение 8 доказано.

Отметим, что $i = 2$ удовлетворяет ограничениям на все тройки (при $n \geq 2$). Таким образом, значение $p_2 = \frac{1}{2}$ гарантированно достигается на трёх тройках, не приводимых друг к другу преобразованиями из $\mathcal{E}'$.

Таблица 6

Множества $\mathcal{T}$, $\mathcal{I}$, $\mathcal{S}$, порождающие $\mathcal{E}'(\alpha, \beta, \gamma)$

(α, β, γ)	$\mathcal{T}$	$\mathcal{I}$	$\mathcal{S}$
$(2^{n-i}, 2^{n-i}, 0)$	$\{\text{id}\}$	$\{\text{id}, \mathcal{I}_{\alpha, \beta}\}$	$\{\text{id}, \mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{S}_{\alpha\beta}\}$
$(0, 2^{n-i+1}, 2^{r-i+1})$	$\{\text{id}, \mathcal{T}_{\alpha, \beta}\}$	$\{\text{id}, \mathcal{I}_{\alpha, \beta}\}$	$\{\text{id}, \mathcal{S}_\beta, \mathcal{S}_\gamma, \mathcal{S}_{\beta\gamma}\}$
$(2^{n-r-i+1}, 0, 2^{n-i+1})$	$\{\text{id}, \mathcal{T}_{\alpha, \beta}\}$	$\{\text{id}, \mathcal{I}_{\alpha, \beta}\}$	$\{\text{id}, \mathcal{S}_\alpha, \mathcal{S}_\gamma, \mathcal{S}_{\alpha\gamma}\}$

Следствие 4. Пусть $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ — тройка из условия теоремы 3. Тогда множество троек $\mathcal{E}'(\alpha, \beta, \gamma)$ порождается композициями преобразований из множеств $\mathcal{T}$, $\mathcal{I}$, $\mathcal{S}$, описанных для каждого случая в табл. 6, при этом

$$|\mathcal{E}'(2^{n-i}, 2^{n-i}, 0)| = \begin{cases} 2 & \text{при } i = 1, \\ 4 & \text{при } i = 2, \\ 8 & \text{при } 2 < i \leq n, \end{cases}$$

$$|\mathcal{E}'(0, 2^{n-i+1}, 2^{r-i+1})| = \begin{cases} 4 & \text{при } i = 2, \\ 16 & \text{при } 2 < i \leq r + 1, \end{cases}$$

$$|\mathcal{E}'(2^{n-r-i+1}, 0, 2^{n-i+1})| = \begin{cases} 8 & \text{при } i = 2, \\ 16 & \text{при } 2 < i \leq n - r + 1. \end{cases}$$

ДОКАЗАТЕЛЬСТВО. Очевидно, что все тройки $\mathcal{E}'(\alpha, \beta, \gamma)$ получаются композициями преобразований из множеств $\{\text{id}, \mathcal{T}_{\alpha\beta}\}$, $\{\text{id}, \mathcal{I}_{\alpha\beta}\}$ и $\{\text{id}, \mathcal{S}_\alpha$,

$\mathcal{S}_\beta, \mathcal{S}_\gamma, \mathcal{S}_{\alpha\beta}, \mathcal{S}_{\alpha\gamma}, \mathcal{S}_{\beta\gamma}, \mathcal{S}_{\alpha\beta\gamma}\}$. Далее, один из элементов каждой тройки нулевой и может быть преобразован только в 2^{n-1} , который также самообратный, поэтому половина возможных вариантов ξ для каждой тройки можно отбросить. В первой тройке два ненулевых элемента равны, тем самым можно убрать $\mathcal{T}_{\alpha\beta}$.

Осталось заметить, что при $i > 2$ первая тройка не содержит ни 2^{n-1} , ни 2^{n-2} . Аналогично для второй и третьей тройки при $i > 3$. Это означает, что все значения вида $c2^{n-1} \pm 2^k$, где $c \in \mathbb{Z}_2$ и k из имеющихся в тройках, будут различны, т. е. имеем описанные в условии следствия мощности. Если $i = 3$, то у третьей тройки только третий элемент равен 2^{n-2} , от которого может быть взят обратный, а во второй тройке возможен случай $2^{n-1} - 2^{n-2} = 2^{n-2}$. Тогда и бывший нулевой элемент будет равен 2^{n-1} , поскольку инверсию старших битов делаем одновременно у двух первых элементов, т. е. и в этом случае тройки будут различными. Случай $i = 1$ для первой тройки и $i = 2$ очевидны. Следствие 4 доказано.

Следствие 5. Если $C_i = |\{(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3 \mid \text{adr}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) = p_i\}|$, $i \in \{1, \dots, n\}$, то

$$C_i = \begin{cases} 2 & \text{при } i = 1, \\ 16 & \text{при } i = 2, \\ 40 & \text{при } 3 \leq i \leq \min\{r+1, n-r+1\}, \\ 24 & \text{при } \min\{r+1, n-r+1\} < i \leq \max\{r+1, n-r+1\}, \\ 8 & \text{при } \max\{r+1, n-r+1\} < i \leq n. \end{cases}$$

В частности, $C_3 = C_4 = \dots = C_n = 24$ при $r = 1$ и $r = n - 1$.

ДОКАЗАТЕЛЬСТВО следует из утверждения 8 и следствия 4. Следствие 5 доказано.

Несложно подсчитать общее число рассматриваемых троек.

Следствие 6. При $n \geq 2$ имеется ровно $24n - 30$ троек $(\alpha, \beta, \gamma) \in (\mathbb{Z}_2^n)^3$ таких, что $\text{adr}^{\text{XR}}(\alpha, \beta \xrightarrow{r} \gamma) > \frac{1}{4}$.

Это число почти в два раза меньше аналогичного для $\text{adr}^\oplus$.

Заключение

В работе получена полная характеристизация разностей по модулю 2^n для преобразований $x \oplus y$ и $(x \oplus y) \lll r$, вероятность которых больше $\frac{1}{4}$. Несмотря на то, что с учётом указанного ограничения спектры значений обеих характеристик совпадают, множества-прообразы «усложняются» после добавления циклического сдвига. Также они содержат примерно

в два раза меньше элементов. Аналогично результатам из [20] минимальные различия получаются при циклическом сдвиге на одну позицию влево или вправо ($r = 1$ и $r = n - 1$ соответственно).

Обратим внимание, что полученные результаты легко применить к более широкому классу преобразований, например к $(x \lll r) \oplus y$ или $((x \boxplus y) \lll r) \oplus z$, поскольку их разностные характеристики прямо выражаются через значения adr^{XR} .

Финансирование работы

Исследование выполнено при поддержке Математического центра в Академгородке в рамках соглашения с Министерством науки и высшего образования Российской Федерации (соглашение № 075–15–2022–282).

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Shimizu A., Miyaguchi S. Fast data encipherment algorithm FEAL // Advances in cryptology — EUROCRYPT'87. Proc. Workshop Theory and Application of Cryptographic Techniques (Amsterdam, The Netherlands, Apr. 13–15, 1987). Heidelberg: Springer, 1988. P. 267–278. (Lect. Notes Comput. Sci.; V. 304). DOI: 10.1007/3-540-39118-5_24.
2. Ferguson N., Lucks S., Schneier B. [et al.]. The Skein hash function family. Santa Barbara, CA: Univ. Calif., 2008. 100 p. URL: www.schneier.com/wp-content/uploads/2015/01/skein.pdf (accessed Apr. 3, 2024).
3. 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 June 1, 2024).
4. Bernstein D. J. Salsa20 specification. Chicago: Univ. Ill. Chic., 2007. 9 p. URL: cr.yp.to/snuffle/spec.pdf (accessed June 1, 2024).
5. Bernstein D. J. ChaCha, a variant of Salsa20. Chicago: Univ. Ill. Chic., 2008. 6 p. URL: cr.yp.to/chacha/chacha-20080128.pdf (accessed June 1, 2024).
6. Koo B., Roh D., Kim H., Jung Y., Lee D., Kwon D. CHAM: A family of lightweight block ciphers for resource-constrained devices // Information security and cryptology — ICISC 2017. Rev. Sel. Pap. 20th Int. Conf. (Seoul, South Korea, Nov. 29–Dec. 1, 2017). Cham: Springer, 2017. P. 3–25. (Lect. Notes Comput. Sci.; V. 10779). DOI: 10.1007/978-3-319-78556-1_1.
7. Roh D., Koo B., Jung Y., Jeong I., Lee D., Kwon D., Kim W. 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.

8. **Mouha N., Mennink B., Herrewewe 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. Workshop (Montreal, Canada, Aug. 14–15, 2014). Cham: Springer, 2014. P. 306–323. (Lect. Notes Comput. Sci.; V. 8781).
9. **Biham E., Shamir A.** Differential cryptanalysis of DES-like cryptosystems // J. Cryptol. 1991. V. 4. P. 3–72.
10. **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.
11. **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). DOI: 10.1007/978-3-642-34961-4_15.
12. **Leurent G.** Construction of differential characteristics in ARX designs application to Skein // Advances in cryptology — CRYPTO 2013. Proc. 33rd Annu. Cryptology Conf. (Santa Barbara, CA, USA, Aug. 18–22, 2013). Pt. I. Heidelberg: Springer, 2013. P. 241–258. (Lect. Notes Comput. Sci.; V. 8042). DOI: 10.1007/978-3-642-40041-4_14.
13. **Малышев Ф. М.** Вероятностные характеристики разностных и линейных соотношений для неоднородной линейной среды // Мат. вопросы криптографии. 2019. Т. 10, № 1. С. 41–72.
14. **Малышев Ф. М.** Разностные характеристики основных операций ARX-шифров // Мат. вопросы криптографии. 2020. Т. 11, № 4. С. 97–105.
15. **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).
16. **Daum M. A.** Cryptanalysis of hash functions of the MD4-family: PhD Thes. Bochum: Ruhr-Univ. Bochum, 2005. 178 p.
17. **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).
18. **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).
19. **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).

20. 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).
21. 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.
22. Gorodilova A. A., Tokareva N. N., Agievich S. V. [et al.]. An overview of the Eighth International Olympiad in Cryptography «Non-Stop University Crypto» // Сиб. электрон. мат. изв. 2022. Т. 19, № 1. С. А.9–А.37.
23. 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.

Мокроусов Антон Сергеевич
Коломеец Николай Александрович

Статья поступила
3 мая 2023 г.
После доработки —
16 октября 2023 г.
Принята к публикации
22 декабря 2023 г.

ADDITIVE DIFFERENTIALS FOR ARX MAPPINGS WITH PROBABILITY EXCEEDING $1/4$

A. S. Mokrousov^a and N. A. Kolomeec^b

Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
E-mail: ^asettingx@mail.ru, ^bkolomeec@math.nsc.ru

Abstract. We consider the additive differential probabilities of functions $x \oplus y$ and $(x \oplus y) \lll r$, where $x, y \in \mathbb{Z}_2^n$ and $1 \leq r < n$. The probabilities are used for the differential cryptanalysis of ARX ciphers that operate only with addition modulo 2^n , bitwise XOR ($\oplus$) and bit rotations ($\lll r$). A complete characterization of differentials whose probability exceeds $1/4$ is obtained. All possible values of their probabilities are $1/3 + 4^{2-i}/6$ for $i \in \{1, \dots, n\}$. We describe differentials with each of these probabilities and calculate the number of these values. We also calculate the number of all considered differentials. It is $48n - 68$ for $x \oplus y$ and $24n - 30$ for $(x \oplus y) \lll r$, where $n \geq 2$. We compare differentials of both mappings under the given constraint. Tab. 6, bibliogr. 23.

Keywords: ARX scheme, differential probabilities, modulo addition, XOR, bit rotation.

References

1. A. Shimizu and S. Miyaguchi, Fast data encipherment algorithm FEAL, in *Advances in Cryptology — EUROCRYPT'87* (Proc. Workshop Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, Apr. 13–15, 1987) (Springer, Heidelberg, 1988), pp. 267–278 (Lect. Notes Comput. Sci., Vol. 304), DOI: 10.1007/3-540-39118-5_24.
2. N. Ferguson, S. Lucks, B. Schneier, [et al.], The Skein Hash Function Family (Univ. California, Santa Barbara, CA, 2008), URL: www.schneier.com/wp-content/uploads/2015/01/skein.pdf (accessed Apr. 3, 2024).

3. **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, Paper ID 2013/404), URL: eprint.iacr.org/2013/404 (accessed June 1, 2024).
4. **D. J. Bernstein**, Salsa20 specification (Univ. Ill. Chic., Chicago, 2007), URL: cr.yp.to/snuffle/spec.pdf (accessed June 1, 2024).
5. **D. J. Bernstein**, ChaCha, a variant of Salsa20 (Univ. Ill. Chic., Chicago, 2008), URL: <https://cr.yp.to/chacha/chacha-20080128.pdf> (accessed June 1, 2024).
6. **B. Koo, D. Roh, H. Kim, Y. Jung, D. Lee, and D. Kwon**, CHAM: A family of lightweight block ciphers for resource-constrained devices, in *Information Security and Cryptology — ICISC 2017* (Rev. Sel. Pap. 20th Int. Conf., Seoul, South Korea, Nov. 29 – Dec. 1, 2017) (Springer, Cham, 2017), pp. 3–25 (Lect. Notes Comput. Sci., Vol. 10779), DOI: 10.1007/978-3-319-78556-1_1.
7. **D. Roh, B. Koo, Y. Jung, I. Jeong, D. Lee, D. Kwon, and W. 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.
8. **N. Mouha, B. Mennink, A. Herrewewege, 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. Workshop, Montreal, Canada, Aug. 14–15, 2014) (Springer, Cham, 2014), pp. 306–323 (Lect. Notes Comput. Sci., Vol. 8781).
9. **E. Biham and A. Shamir**, Differential cryptanalysis of DES-like cryptosystems, *J. Cryptol.* **4**, 3–72 (1991).
10. **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.
11. **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), DOI: 10.1007/978-3-642-34961-4_15.
12. **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, CA, USA, Aug. 18–22, 2013), Pt. I (Springer, Heidelberg, 2013), pp. 241–258 (Lect. Notes Comput. Sci., Vol. 8042), DOI: 10.1007/978-3-642-40041-4_14.
13. **F. M. Malyshev**, Probabilistic characteristics of differential and linear relations for nonhomogeneous linear medium, *Mat. Vopr. Kriptogr.* **10** (1), 41–72 (2019) [Russian].
14. **F. M. Malyshev**, Differential characteristics of base operations in ARX-ciphers, *Mat. Vopr. Kriptogr.* **11** (4), 97–105 (2020) [Russian].

15. **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).
16. **M. A. Daum**, Cryptanalysis of hash functions of the MD4-family, *PhD Thesis* (Ruhr-Univ. Bochum, Bochum, 2005).
17. **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).
18. **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).
19. **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).
20. **N. A. Kolomeec, I. A. Sutormin, D. A. Bykov, M. A. Panferov, and T. A. Bonich**, On additive differential probabilities of the composition of bit-wise exclusive-or and a bit rotation (Cornell Univ., Ithaca, NY, 2023) (Cornell Univ. Libr. e-Print Archive, arXiv:2303.04097).
21. **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).
22. **A. A. Gorodilova, N. N. Tokareva, S. V. Agievich, [et al.]**, An overview of the Eighth International Olympiad in Cryptography “Non-Stop University Crypto”, *Sib. Elektron. Mat. Izv.* **19** (1), A.9–A.37 (2022).
23. **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).

Anton S. Mokrousov
Nikolay A. Kolomeec

Received May 3, 2023
Revised October 16, 2023
Accepted December 22, 2023

О СЛОЖНОСТИ ЗАДАЧИ ВЫБОРА КЛАСТЕРОВ БОЛЬШОГО РАЗМЕРА

А. В. Пяткин

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

Аннотация. Рассматривается задача выбора в данном множестве евклидовых векторов заданного числа кластеров с ограничением на максимальный разброс каждого кластера так, чтобы размер минимального из этих кластеров был максимальным. Под разбросом понимается сумма квадратов расстояний от элементов кластера до его центроида. Доказана NP-трудность этой задачи в случае, когда размерность пространства является частью входа. Библиогр. 15.

Ключевые слова: кластер, центроид, разброс, NP-трудность.

Введение

Объектом изучения настоящей статьи является задача, в которой для заданного множества векторов $\mathcal{U} \subseteq \mathbb{R}^d$ необходимо построить семейство попарно не пересекающихся подмножеств (кластеров) «похожих» векторов из $\mathcal{U}$ при ограничении на разброс внутри каждого кластера, при этом размер минимального из этих кластеров максимизируется. Целью работы является определение алгоритмической сложности данной задачи в случае, когда разброс определяется как сумма квадратов расстояний от элементов кластера до его центроида, а размерность пространства является частью входа.

Задачи кластеризации весьма актуальны для анализа и обработки данных, искусственного интеллекта, вычислительной геометрии, математической статистики и дискретной оптимизации [1–3]. Одним из наиболее известных примеров является классическая задача MSSC (minimum sum-square clustering), также иногда называемая k -means [4–6]. В ней требуется разбить данное множество векторов евклидова пространства на k непересекающихся кластеров, минимизируя суммарный разброс (сумму квадратов расстояний от элемента каждого кластера

до его центроида) всех кластеров. Известно, что эта задача NP-трудна [7] даже при $k = 2$. Задача также NP-трудна в случае, когда заданы размеры кластеров [8], даже если размеры всех кластеров равны 3. В [9] доказано, что задача разбиения на два кластера равного размера также NP-трудна.

Однако в случае, когда исходные данные содержат случайные выбросы, более актуальна не задача разбиения множества на кластеры, а задача выбора k непересекающихся подмножеств с заданными ограничениями на их размеры и разброс, причём число кластеров k в этом случае является не частью входа, а параметром задачи. В случае $k = 1$ задача NP-трудна [10]; при $k \geq 2$ её сложностной статус неизвестен.

В работе исследуется

Задача 1 (Поиск k кластеров). Даны множество $\mathcal{Y} = \{y_1, \dots, y_n\}$ векторов в пространстве $\mathbb{R}^d$ и число $A > 0$. Выделить k попарно не пересекающихся кластеров $\mathcal{C}_i \subseteq \mathcal{Y}$, $i = 1, \dots, k$, максимизируя $\min\{|\mathcal{C}_1|, \dots, |\mathcal{C}_k|\}$ при условии

$$f(\mathcal{C}_i) = \sum_{y \in \mathcal{C}_i} \|y - \bar{y}(\mathcal{C}_i)\|^2 \leq A, \quad i = 1, \dots, k,$$

где $\bar{y}(\mathcal{C}_i) = \frac{1}{|\mathcal{C}_i|} \sum_{y \in \mathcal{C}_i} y$ — центроид кластера $\mathcal{C}_i$.

Близкие в постановочном плане задачи, когда вместо центроида используются заданные точки пространства или точки исходного множества, рассматривались в работе [11], где была доказана их NP-трудность даже в одномерном случае (т. е. при $d = 1$). В [12] для задачи 1 был предложен приближённый полиномиальный алгоритм с оценкой точности $1/k$ для произвольного $k \geq 2$ и размерности пространства $d = 1$. Там же было отмечено, что сложностной статус задачи 1 неизвестен.

В настоящей статье доказывается, что задача 1 NP-трудна в случае, когда размерность пространства d является частью входа. В разд. 1 приводятся предварительные сведения, необходимые для доказательства основного результата в разд. 2. В заключении даются финальные комментарии.

1. Предварительные результаты

Пусть задан граф $G = (V, E)$ и целое число $k \geq 2$. Тогда k -раскраской графа G называется разбиение множества вершин V на k независимых подмножеств $V_1, \dots, V_k$, называемых *цветами*. Известно, что задача существования в графе k -раскраски NP-полна [13] при $k \geq 3$. Более того, эта задача остаётся NP-полной для r -однородных графов при $r > k$, поскольку в [14] было доказано, что проблема 3-раскраски 4-однородного плоского графа NP-полна. В *сбалансированной* k -раскраске требуется

найти разбиение на k независимых подмножеств с дополнительным условием, что $|V_i| - |V_j| \in \{-1, 0, 1\}$ для всех i, j , т. е. размеры всех независимых множеств должны быть примерно одинаковы. Очевидно, что эта задача тоже NP-трудна, так как задача k -раскраски графа легко сводится к сбалансированной версии добавлением нужного числа независимых вершин. Покажем, что сбалансированная k -раскраска NP-трудна в следующем частном случае.

Задача 2 (Сбалансированная k -раскраска однородного графа). Даны r -однородный граф $G = (V, E)$ на n вершинах и целое число $k \geq 3$, причём $n = kt$ для некоторого целого $t > 1$. Существует ли разбиение V на k независимых подмножеств $V_1, \dots, V_k$, мощность каждого из которых равна t ?

Лемма 1. Задача 2 NP-полна.

ДОКАЗАТЕЛЬСТВО. Сведём к задаче 2 проблему существования k -раскраски в r -однородном графе. Рассмотрим произвольный r -однородный t -вершинный граф $H = (V_0, E_0)$, который требуется раскрасить в k цветов. Возьмём в качестве G граф, полученный объединением k копий графа H . Очевидно, что G является r -однородным графом на kt вершинах. Покажем, что в нём есть сбалансированная k -раскраска тогда и только тогда, когда граф H k -раскрашиваем.

Очевидно, что ограничение сбалансированной k -раскраски графа G на любую из копий графа H задаёт его k -раскраску.

Предположим, что граф H можно раскрасить в k цветов. Тогда раскрасим первую копию в графе G в соответствии с этой раскраской, а в каждой последующей копии сделаем циклический сдвиг цветов на 1 (т. е. в i -й копии цвет j имеют те и только те вершины, которые в 1-й копии имеют цвет $j - i + 1$, где разность берётся по модулю k). Нетрудно видеть, что каждым цветом раскрашено ровно t вершин (например, цветом 1 в i -й копии раскрашены вершины, которые в графе H раскрашены цветом i) и каждый цвет образует независимое множество. Лемма 1 доказана.

Нам также потребуется следующее известное тождество (доказательство можно найти, например, в [15]).

Утверждение 1 [15]. Имеет место формула

$$\sum_{y \in C} \|y - \bar{y}(C)\|^2 = \frac{1}{2|C|} \sum_{y, z \in C} \|y - z\|^2.$$

2. Основной результат

Сформулируем задачу 1 в виде задачи верификации свойств.

Задача 3. Даны множество $\mathcal{Y} = \{y_1, \dots, y_n\}$ векторов в пространстве $\mathbb{R}^d$ и числа $A > 0$, $M > 1$. Существует ли k попарно не пересекающихся кластеров $\mathcal{C}_i \subseteq \mathcal{Y}$, $i = 1, \dots, k$, удовлетворяющих условиям $|\mathcal{C}_i| \geq M$ и $f(\mathcal{C}_i) \leq A$ для всех $i = 1, \dots, k$?

Основным результатом работы является

Теорема 1. Задача 3 NP-полна.

ДОКАЗАТЕЛЬСТВО. Построим сведение задачи 2 к задаче 3. Рассмотрим произвольный r -однородный граф $G = (V, E)$ на n вершинах и целое число $k \geq 3$, причём $n = kt$. Зафиксируем произвольную ориентацию рёбер графа G . Положим $d = |E|$ и для каждого $j = 1, \dots, d$ обозначим j -ю дугу графа G через e_j . Далее для всех $i = 1, \dots, n$ поставим в соответствие вершине v_i этого графа следующий d -мерный вектор y_i . Пусть j -я координата вектора y_i равна -1 , если дуга e_j исходит из вершины v_i ; равна 1 , если дуга e_j заходит в вершину v_i ; и равна 0 , если дуга e_j не инцидентна вершине v_i . Положим $M = t$ и $A = r(t - 1)$. Поскольку граф G r -однороден, то

$$\|y_i - y_j\|^2 = \begin{cases} 0, & \text{если } i = j, \\ 2r, & \text{если } v_i v_j \notin E, \\ 2r + 2, & \text{если } v_i v_j \in E. \end{cases}$$

Покажем, что k попарно не пересекающихся кластеров с требуемыми условиями существуют тогда и только тогда, когда G допускает сбалансированную k -раскраску.

Если сбалансированная k -раскраска существует, то каждым цветом раскрашено ровно t вершин. Положим $\mathcal{C}_i = \{y_j \mid v_j \text{ имеет цвет } i\}$. Тогда $|\mathcal{C}_i| = t = M$ и по утверждению 1 $f(\mathcal{C}_i) = 2r(t - 1)t/2t = r(t - 1) = A$ для всех $i = 1, \dots, k$, что и требуется.

Предположим, что существует требуемое разбиение в задаче 3. Поскольку $|\mathcal{Y}| = Mk$, имеем $|\mathcal{C}_i| = M$ для всех $i = 1, \dots, k$. Пусть подграф V_i , порождённый всеми вершинами v_j , для которых $y_j \in \mathcal{C}_i$, содержит k_i рёбер. Тогда по утверждению 1 имеем

$$f(\mathcal{C}_i) = \frac{2rM(M - 1) + 2k_i}{2M} = A + \frac{k_i}{M}.$$

Поскольку $f(\mathcal{C}_i) \leq A$, получаем $k_i = 0$, откуда вытекает, что V_i независимо, т. е. имеет место сбалансированная раскраска графа G в k цветов. Теорема 1 доказана.

Заключение

В работе доказана NP-полнота задачи выбора в некотором множестве векторов d -мерного евклидова пространства k кластеров большого размера с заданным ограничением на максимальный разброс в случае, когда размерность пространства является частью входа. Ранее сложностной статус этой задачи был неизвестен при $k \geq 2$. Интересной открытой проблемой остаётся выяснение сложности задачи в случае фиксированной размерности пространства.

Финансирование работы

Работа выполнена в рамках государственного задания Институт математики им. С. Л. Соболева СО РАН (проект № FWNF-2022-0019).

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Berkhin P.** A survey of clustering data mining techniques // Grouping multidimensional data: Recent advances in clustering. Heidelberg: Springer, 2006. P. 25–71. DOI: 10.1007/3-540-28349-8_2.
2. **Jain A. K., Dubes R. C.** Algorithms for clustering data. Englewood Cliffs, NJ: Prentice Hall, 1988. 320 p.
3. **Ghoreyshi S., Hosseinkhani J.** Developing a clustering model based on K -means algorithm in order to creating different policies for policyholders in insurance industry // Int. J. Adv. Comput. Sci. Inf. Technol. 2015. V. 4, No. 2. P. 46–53.
4. **Fisher W. D.** On grouping for maximum homogeneity // J. Am. Stat. Assoc. 1958. V. 53, No. 284. P. 789–798.
5. **MacQueen J.** Some methods for classification and analysis of multivariate observations // Proc. 5th Berkeley Symp. Mathematics, Statistics and Probability (Berkeley, USA, June 21–July 18, 1965; Dec. 27, 1965–Jan. 7, 1966). V. 1. Berkeley: Univ. Calif. Press, 1967. P. 281–297.
6. **Blömer J., Lammersen C., Schmidt M., Sohler C.** Theoretical analysis of the k -means algorithm — A survey // Algorithm engineering: Selected results and surveys. Cham: Springer, 2016. P. 81–116. (Lect. Notes Comput. Sci.; V. 9220). DOI: 10.1007/978-3-319-49487-6.
7. **Aloise D., Deshpande A., Hansen P., Popat P.** NP-hardness of Euclidean sum-of-squares clustering // Mach. Learn. 2009. V. 75, No. 2. P. 245–248.
8. **Pyatkin A. V., Aloise D., Mladenovic N.** NP-hardness of balanced minimum sum-of-squares clustering // Pattern Recognit. Lett. 2017. V. 97. P. 44–45.
9. **Bertoni A., Goldwurm M., Lin J., Saccà F.** Size constrained distance clustering: Separation properties and some complexity results // Fund. Inform. 2012. V. 115, No. 1. P. 125–139. DOI: 10.3233/FI-2012-620.

10. Кельманов А. В., Пяткин А. В. NP-полнота некоторых задач выбора подмножества векторов // Дискрет. анализ и исслед. операций. 2010. Т. 17, № 5. С. 37–45.
11. Кельманов А. В., Пяткин А. В., Хандеев В. И. О сложности некоторых максиминных задач кластеризации // Тр. Ин-та математики и механики. 2018. Т. 24, № 4. С. 189–198. DOI: 10.21538/0134-4889-2018-24-4-189-198.
12. Khandeev V. I., Neshchadim S. M. Constant-factor approximation algorithms for some maximin multi-clustering problems // Mathematical optimization theory and operations research. Proc. 22nd Int. Conf. (Yekaterinburg, Russia, July 2–8, 2023). Cham: Springer, 2023. P. 85–100. (Lect. Notes Comput. Sci.; V. 13930). DOI: 10.1007/978-3-031-35305-5_6.
13. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
14. Dailey D. P. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete // Discrete Math. 1980. V. 30, No. 3. P. 289–293. DOI: 10.1016/0012-365X(80)90236-8.
15. Кельманов А. В., Пяткин А. В. Об одном варианте задачи выбора подмножества векторов // Дискрет. анализ и исслед. операций. 2008. Т. 15, № 5. С. 20–34.

Пяткин Артём Валерьевич

Статья поступила

8 ноября 2023 г.

После доработки —

20 ноября 2023 г.

Принята к публикации

22 декабря 2023 г.

ON THE COMPLEXITY OF THE PROBLEM
OF CHOICE OF LARGE CLUSTERS

A. V. Pyatkin

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

Abstract. The paper considers the following problem. Given a set of Euclidean vectors, find several clusters with a restriction on the maximum scatter of each cluster so that the size of the minimum cluster would be maximum. Here the scatter is the sum of squared distances from the cluster elements to its centroid. The NP-hardness of this problem is proved in the case where the dimension of the space is a part of the input. Bibliogr. 15.

Keywords: cluster, centroid, scatter, NP-hardness.

References

1. **P. Berkhin**, A survey of clustering data mining techniques, in *Grouping Multi-dimensional Data: Recent Advances in Clustering* (Springer, Heidelberg, 2006), pp. 25–71, DOI: 10.1007/3-540-28349-8_2.
2. **A. K. Jain** and **R. C. Dubes**, *Algorithms for Clustering Data* (Prentice Hall, Englewood Cliffs, NJ, 1988).
3. **S. Ghoreyshi** and **J. Hosseinkhani**, Developing a clustering model based on K -means algorithm in order to creating different policies for policyholders in insurance industry, *Int. J. Adv. Comput. Sci. Inf. Technol.* **4** (2), 46–53 (2015).
4. **W. D. Fisher**, On grouping for maximum homogeneity, *J. Am. Stat. Assoc.* **53** (284), 789–798 (1958).
5. **J. MacQueen**, Some methods for classification and analysis of multivariate observations, in *Proc. 5th Berkeley Symp. Mathematics, Statistics and Probability, Berkeley, USA, June 21 – July 18, 1965; Dec. 27, 1965 – Jan. 7, 1966*, Vol. 1 (Univ. Calif. Press, Berkeley, 1967), pp. 281–297.

6. **J. Blömer, C. Lammersen, M. Schmidt, and C. Sohler**, Theoretical analysis of the k -means algorithm — A survey, in *Algorithm Engineering: Selected Results and Surveys* (Springer, Cham, 2016), pp. 81–116 (Lect. Notes Comput. Sci., Vol. 9220), DOI: 10.1007/978-3-319-49487-6_3.
7. **D. Aloise, A. Deshpande, P. Hansen, and P. Popat**, NP-hardness of Euclidean sum-of-squares clustering, *Mach. Learn.* **75** (2), 245–248 (2009).
8. **A. V. Pyatkin, D. Aloise, and N. Mladenovic**, NP-hardness of balanced minimum sum-of-squares clustering, *Pattern Recognit. Lett.* **97**, 44–45 (2017).
9. **A. Bertoni, M. Goldwurm, J. Lin, and F. Saccà**, Size constrained distance clustering: Separation properties and some complexity results, *Fund. Inform.* **115** (1), 125–139 (2012), DOI: 10.3233/FI-2012-620.
10. **A. V. Kel'manov and A. V. Pyatkin**, NP-completeness of some problems of choosing a vector subset, *Diskretn. Anal. Issled. Oper.* **17** (5), 37–45 (2010) [Russian] [*J. Appl. Ind. Math.* **5** (3), 352–357 (2011)].
11. **A. V. Kel'manov, A. V. Pyatkin, and V. I. Khandeev**, On the complexity of some max–min clustering problems, *Tr. Inst. Mat. Mekh.* **24** (4), 189–198 (2018), DOI: 10.21538/0134-4889-2018-24-4-189-198 [Russian] [*Proc. Steklov Inst. Math.* **309** (Suppl. 1), S65–S73 (2020), DOI: 10.1134/S0081543820040082].
12. **V. I. Khandeev and S. M. Neshchadim**, Constant-factor approximation algorithms for some maximin multi-clustering problems, in *Mathematical Optimization Theory and Operations Research* (Proc. 22nd Int. Conf., Yekaterinburg, Russia, July 2–8, 2023) (Springer, Cham, 2023), pp. 85–100 (Lect. Notes Comput. Sci., Vol. 13930), DOI: 10.1007/978-3-031-35305-5_6.
13. **M. R. Garey and D. S. Johnson**, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979; Mir, Moscow, 1982 [Russian]).
14. **D. P. Dailey**, Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete, *Discrete Math.* **30** (3), 289–293 (1980), DOI: 10.1016/0012-365X(80)90236-8.
15. **A. V. Kel'manov and A. V. Pyatkin**, On a version of the problem of choosing a vector subset, *Diskretn. Anal. Issled. Oper.* **15** (5), 20–34 (2008) [Russian] [*J. Appl. Ind. Math.* **3** (4), 447–455 (2009)].

Artem V. Pyatkin

Received November 8, 2023

Revised November 20, 2023

Accepted December 22, 2023

О СЛОЖНОСТИ МЕТОДА ПОСЛЕДОВАТЕЛЬНОГО ОПРОБОВАНИЯ

В. М. Фомичёв^{1,2}

¹ ООО «Код Безопасности»,

1-й Нагатинский пр-д, 10, стр. 1, 115230 Москва, Россия

² Институт проблем информатики ФИЦ «Информатика и управление» РАН,
ул. Вавилова, 44, корп. 2, 119333 Москва, Россия

E-mail: fomichev.2016@yandex.ru

Аннотация. Система m булевых уравнений может быть решена методом последовательного опробования с помощью m -шагового алгоритма, где на i -м шаге опробуются значения всех переменных, существенных для первых i уравнений, и ложные решения отбраковываются по правым частям уравнений, $i = 1, \dots, m$. Оценка сложности метода, зависящая от структуры множеств существенных переменных уравнений, достигает минимума при некоторых перестановках уравнений системы. Предложен алгоритм оптимальной перестановки уравнений, минимизирующей среднюю вычислительную сложность алгоритма в естественных вероятностных предположениях. В ряде случаев оптимальные перестановки определены неоднозначно, и их нахождение является вычислительно сложным. Метод последовательного опробования вырождается в полный перебор, если каждое уравнение системы зависит существенно от всех переменных. Приведён пример построения оптимальной перестановки. Табл. 2, ил. 1, билиогр. 11.

Ключевые слова: булева функция, существенная переменная, решётка подмножеств множества, цепь в решётке.

Введение

Решение системы булевых уравнений в общем случае относится к задачам высокой сложности [1]. Для решения системы с n неизвестными требуется до 2^n операций опробования значений неизвестных.

Сложность решения некоторых частных классов систем булевых уравнений оценивается как полиномиальная или субэкспоненциальная. Большое число работ посвящено решению линейных систем булевых уравнений, где мотивацией было обобщение и улучшение характеристик метода

Гаусса и других методов [2, 3]. Активно исследовались методы линеаризации, направленные на сведение решения нелинейной системы к некоторому числу линейных систем [4, 5]. Средняя сложность решения треугольных систем n уравнений от n переменных в естественной вероятностной модели близка к $2n$ [5]. Со сложностью порядка $\sqrt{2^n}$ методом согласования (англ. meet in the middle of attack) решается широкий класс систем, например системы уравнений, в которых АНФ каждой функции системы разлагается в сумму двух многочленов от непересекающихся подмножеств переменных; варианты этого метода описаны в [5–7].

Имеется также ряд исследований, претендующих на достаточно высокую степень общности в связи с классом решаемых систем уравнений. Их задача — выявить параметры систем уравнений, определяющие сложность решения, что позволяет конструктивно описать некоторые классы систем уравнений с оценкой сложности решения ниже тривиальной оценки 2^n [8–11].

Метод последовательного опробования применим к любой системе булевых уравнений, однако не для всех систем он эффективен, т. е. не для всех систем его сложность ниже сложности полного опробования вариантов решения системы. Для систем уравнений, в которых некоторые функции зависят не от всех переменных, показано, что с помощью перестановки уравнений системы можно достичь определённого приближения к треугольной системе уравнений и минимизации сложности метода последовательного опробования. Предложен алгоритм построения оптимальной перестановки уравнений на основе анализа множеств существенных переменных всех функций системы и приведён пример.

1. Постановка задачи

Требуется решить систему m булевых уравнений

$$f_j(x_1, \dots, x_n) = a_j, \quad j = 1, \dots, m, \quad (1)$$

где $a_j \in \{0, 1\}$, $f_j: \{0, 1\}^n \rightarrow \{0, 1\}$ — булевы функции, возможно существенно зависящие не от всех переменных, при этом каждая переменная существенна хотя бы для одной из функций в левой части системы. *Решением системы (1)* называется любой набор значений переменных $(x_1, \dots, x_n) \in \{0, 1\}^n$, при котором выполняется каждое уравнение системы. Остальные наборы называются *ложными решениями*. Отнесение набора к множеству ложных решений называют *отбраковкой*.

Пусть $Q_r = (q_1, \dots, q_r)$ — бесповторная упорядоченная выборка $r \in \{1, \dots, m\}$ чисел из множества $\{1, \dots, m\}$, а $\{Q_r\} = \{q_1, \dots, q_r\}$ — множество элементов выборки Q_r . Выборки Q_r и Q'_r назовём *односоставными*, если $\{Q_r\} = \{Q'_r\}$.

Перестановку $Q = Q_m = (q_1, \dots, q_m)$ чисел $1, \dots, m$ назовём *маркером* алгоритма решения системы (1). Маркер Q однозначно определяет перестановку уравнений системы (1).

Для решения системы (1) применим m -шаговый алгоритм последовательного опробования (а. п. о.) с маркером $Q = (q_1, \dots, q_m)$, который последовательно бракует ложные решения по уравнениям: сначала с номером q_1 , затем с номером q_2 и т. д., наконец с номером q_m .

Задача состоит в том, чтобы найти маркер для а. п. о. с наименьшей вычислительной сложностью; назовём его *оптимальным маркером* (о. м.) для системы (1).

2. Алгоритм последовательного опробования

Пусть $S(j)$ — множество номеров всех существенных переменных булевой функции $f_j(x_1, \dots, x_n)$ в левой части системы (1), $j = 1, \dots, m$. Для маркера $Q = (q_1, \dots, q_m)$ обозначим

$$\begin{aligned}\Delta_1(Q) &= S(q_1), \\ \Delta_r(Q) &= S(q_r) \setminus (S(q_1) \cup \dots \cup S(q_{r-1})), \quad 2 \leq r \leq m, \\ n_r(Q) &= |\Delta_r(Q)|, \quad r = 1, \dots, m.\end{aligned}$$

Для краткости не указываем зависимость множеств и величин от выборки Q (т. е. $\Delta_r(Q) = \Delta_r$ и $n_r(Q) = n_r$), если корректность рассуждений не нарушается.

Опишем а. п. о. $A(Q)$ с маркером Q и оценим его сложность, измеряемую числом операций отбраковки наборов значений некоторых переменных из $\{x_1, \dots, x_n\}$ по одному из уравнений системы (1). Например, если $S(j) = \{i_1, \dots, i_k\}$, $j \in \{1, \dots, m\}$, то игнорируя несущественные переменные, имеем $f_j(x_1, \dots, x_n) = f_j(x_{i_1}, \dots, x_{i_k})$. Тогда одна операция состоит в проверке для произвольного набора двоичных констант $(\beta_{i_1}, \dots, \beta_{i_k})$ выполнения равенства

$$f_j(\beta_{i_1}, \dots, \beta_{i_k}) = a_j. \quad (2)$$

Если (2) не верно, то набор $(\beta_{i_1}, \dots, \beta_{i_k})$ «забракован по j -му уравнению»; в противном случае набор «не забракован по j -му уравнению», и далее для этого набора или его пополнения некоторым числом булевых констант проверяется выполнимость другого уравнения.

Алгоритм $A(Q)$ (с маркером $Q = (q_1, \dots, q_m)$)

ШАГ 1. Опробуются все наборы значений n_1 переменных с номерами из $S(q_1)$, и ложные наборы бракуются по уравнению с номером q_1 .

ШАГ $r \geq 2$. Каждый не отбракованный на $(r-1)$ -м шаге набор

1) бракуется по уравнению с номером q_r , если $\Delta_r = \emptyset$;

2) пополняется 2^{n_r} наборами значений переменных с номерами из Δ_r , если $\Delta_r \neq \emptyset$, и пополненные наборы значений переменных бракуются по уравнению с номером q_r .

При $r < m$ для каждого неотбракованного набора значений переменных выполняется шаг $r + 1$. При $r = m$ каждый неотбракованный набор значений переменных $(x_1, \dots, x_n) \in \{0, 1\}^n$ является решением системы уравнений (1).

КОНЕЦ.

Оценим среднюю сложность $T(Q)$ алгоритма $A(Q)$ решения случайной системы (1) при следующих условиях:

1) правая часть $(a_1, \dots, a_m) \in \{0, 1\}^m$ системы (1) выбрана случайно и равномерно;

2) при любом $j \in \{1, \dots, m\}$ равенство (2) выполняется с вероятностью $\frac{1}{2}$ для любого двоичного набора $(\beta_{i_1}, \dots, \beta_{i_k})$;

3) уравнения системы (1) независимы: при случайном наборе значений переменных вероятность выполнения любого уравнения не зависит от выполнимости других уравнений при этом наборе значений;

4) наборы значений переменных независимы: вероятность выполнения любого уравнения при случайном наборе значений переменных не зависит от выполнимости этого уравнения при любом другом наборе.

В этих условиях среднее число решений системы (1) равно 2^{n-m} .

Пусть $T(Q)$ обозначает среднюю вычислительную сложность а. п. о. $A(Q)$, $T_r(Q)$ — среднее число операций при выполнении r -го шага а. п. о., а $M_r(Q)$ — среднее число неотбракованных наборов по завершении шага r алгоритма. Из описания алгоритма $A(Q)$ следует, что

$$\sum_{1 \leq r \leq m} n_r(Q) = n, \quad (3)$$

$$T_r(Q) = M_{r-1}(Q) \cdot 2^{n_r},$$

$$M_r(Q) = M_{r-1}(Q) \cdot 2^{n_r-1}, \quad 1 \leq r \leq m,$$

где $M_0(Q) = 1$. Так как $T(Q) = \sum_{1 \leq r \leq m} T_r(Q)$, получаем

$$T(Q) = \sum_{1 \leq r \leq m} M_{r-1}(Q) \cdot 2^{n_r}. \quad (4)$$

Отсюда средняя сложность а. п. о. $A(Q)$ зависит от разбиения (3) числа n на m неотрицательных целых слагаемых $n_1(Q), \dots, n_m(Q)$.

3. Оптимальный маркер

Пусть R_n^m — множество разбиений натурального числа n на m неотрицательных целых слагаемых. Определим расстояние $\rho(\bar{n}, \bar{n}')$ между

разбиениями $\bar{n} = (n_1, \dots, n_m)$, $\bar{n}' = (n'_1, \dots, n'_m) \in R_n^m$ равенством

$$\rho(\bar{n}, \bar{n}') = \sum_{1 \leq i \leq m} |n_i - n'_i|.$$

Разбиения $\bar{n}$ и $\bar{n}'$ назовём *соседними*, если $\rho(\bar{n}, \bar{n}') = 2$. Для соседних разбиений множество разностей $\{n_1 - n'_1, \dots, n_m - n'_m\}$ состоит из одной единицы, одной «минус единицы» и $m - 2$ нулей. Заметим, что множество R_n^m линейно упорядочено лексикографически.

Определим бинарное отношение $\leq$ на множестве R_n^m . Для $\bar{n}, \bar{n}' \in R_n^m$ положим $\bar{n} \leq \bar{n}'$ тогда и только тогда, когда имеется цепь разбиений $(\bar{n}^{(1)}, \dots, \bar{n}^{(t)})$, $t > 1$, такая, что $\bar{n}^{(1)} = \bar{n}$, $\bar{n}^{(t)} = \bar{n}'$, а $\bar{n}^{(j)}$ и $\bar{n}^{(j+1)}$ соседние, $j = 1, \dots, t - 1$, причём разбиение $\bar{n}^{(j)}$ лексикографически меньше разбиения $\bar{n}^{(j+1)}$.

В силу (3) маркер Q однозначно определяет разбиение $\bar{n}(Q) = (n_1(Q), \dots, n_m(Q)) \in R_n^m$. Для краткости пишем $\bar{n}(Q) = \bar{n}$, $\bar{n}(Q') = \bar{n}'$.

Теорема 1. Если $\bar{n} \leq \bar{n}'$ для маркеров Q и Q' , то $T(Q) < T(Q')$.

ДОКАЗАТЕЛЬСТВО. Если разбиения $\bar{n}, \bar{n}'$ соседние, то $\rho(\bar{n}, \bar{n}') = 2$, и для некоторых $1 \leq k < l \leq m$, $n_k \geq 1$ разбиения имеют вид

$$\bar{n} = (n_1, \dots, n_{k-1}, n_k, n_{k+1}, \dots, n_{l-1}, n_l, n_{l+1}, \dots, n_m),$$

$$\bar{n}' = (n_1, \dots, n_{k-1}, n_k + 1, n_{k+1}, \dots, n_{l-1}, n_l - 1, n_{l+1}, \dots, n_m).$$

Тогда для среднего числа неотбракованных наборов M_r после шага r а. п. о. верно равенство

$$M_r(Q) = \begin{cases} M_r(Q') & \text{при } r = 1, \dots, k - 1, l, \dots, m, \\ \frac{1}{2} M_r(Q') & \text{при } r = k, \dots, l - 1, \end{cases} \quad (5)$$

при этом в силу (4) и (5)

$$T(Q') - T(Q) = \sum_{k \leq r \leq l-1} \frac{M_r(Q')}{2} > 0,$$

т. е. для соседних наборов теорема верна.

Если $\bar{n}, \bar{n}'$ не соседние, то по определению отношения $\leq$ в R_n^m имеется цепь разбиений $(\bar{n}^{(1)}, \dots, \bar{n}^{(t)})$, $t > 1$, такая, что $\bar{n}^{(1)} = \bar{n}$, $\bar{n}^{(t)} = \bar{n}'$, причём разбиения $\bar{n}^{(j)}$, $\bar{n}^{(j+1)}$ соседние и $\bar{n}^{(j)}$ лексикографически меньше $\bar{n}^{(j+1)}$, $j = 1, \dots, t - 1$. Тогда в соответствии с доказанным получаем

$$T(Q) = T(\bar{n}^{(1)}) < T(\bar{n}^{(2)}) < \dots < T(\bar{n}^{(t)}) = T(Q').$$

Теорема 1 доказана.

Заметим, что множество $S^{(m)} = \{S(1), \dots, S(m)\}$, порождаемое системой уравнений (1), частично упорядочено (образует ч. у. м.) относительно теоретико-множественного включения.

Теорема 2. Если $Q = (q_1, \dots, q_m)$ — о. м., то $S(q_1)$ — минимальный элемент ч. у. м. $S^{(m)}$.

ДОКАЗАТЕЛЬСТВО. Если множество $S^{(m)}$ — антицепь в решётке (булеане) $2^{\{1, \dots, m\}}$, то все элементы $S^{(m)}$ минимальные, и утверждение теоремы тривиально. Если $S^{(m)}$ не антицепь, то покажем, что $S(q_1)$ минимальный элемент в этом множестве.

Предположим, напротив, что в $S^{(m)}$ имеется цепь с минимальным элементом $S(q_h) \subset S(q_1)$ при некотором $h \geq 2$. Без ограничения общности будем считать, что $h = 2$, и пусть $u = |S(q_2)|$, $v = |S(q_1)|$. Тогда $u < v$ и можно построить цепь разбиений из R_n^m

$$\bar{n}^{(0)} \leq \bar{n}^{(1)} \leq \dots \leq \bar{n}^{(v-u)}$$

такую, что $\bar{n}^{(t)} = (u + t, v - u - t, n_3, \dots, n_m)$, $t = 0, 1, \dots, v - u$. В этой цепи $\bar{n}^{(v-u)} = \bar{n}(Q)$, а разбиение $\bar{n}^{(0)} = \bar{n}(Q')$ соответствует маркеру $Q' = (q_2, q_1, q_3, \dots, q_m)$. В соответствии с теоремой 1 получаем $T(Q') < T(Q)$, что противоречит оптимальности маркера Q . Следовательно, $S(q_1)$ — минимальный элемент в $S^{(m)}$. Теорема 2 доказана.

Теорема 2 определяет m -шаговый алгоритм $A(S^{(m)})$ нахождения о. м. Обозначим через $M(X)$ множество минимальных элементов ч. у. м. X , а для маркера Q_r положим $S^{(m) \setminus Q_r} = S^{(m)} \setminus \{S(q_j)\}_{j=1}^r$, $1 \leq r < m$.

Алгоритм $A(S^{(m)})$

ШАГ 1. Вычислить $M(S^{(m)})$ и сформировать варианты для выборки $Q_1 = (q_1)$, где $S(q_1) \in M(S^{(m)})$.

ШАГ r ($2 \leq r \leq m - 1$). Для каждой выборки $Q_{r-1} = (q_1, \dots, q_{r-1})$, сформированной на шаге $r - 1$, сформировать варианты выборки $Q_r = (q_1, \dots, q_{r-1}, q_r)$, где $S(q_r) \in M(S^{(m) \setminus Q_{r-1}})$.

ШАГ m . Для каждой выборки $Q_{m-1} = (q_1, \dots, q_{m-1})$, сформированной на шаге $m - 1$, сформировать маркер $Q = (q_1, \dots, q_{m-1}, q_m)$, где $q_m \notin \{Q_{m-1}\}$.

Замечание 1. О. м. $Q = (q_1, \dots, q_m)$ является элементом декартова произведения $M(S^{(m)}) \times M(S^{(m) \setminus Q_1}) \times \dots \times M(S^{(m) \setminus Q_{m-1}})$ и в общем случае определён неоднозначно. Многозначность о. м. в худшем случае, когда $S(1) = \dots = S(m) = \{1, \dots, n\}$, оценивается величиной $m!$.

Многозначность и сложность поиска о. м. для систем уравнений (1) снижается, в частности, при следующих условиях:

1) если на шаге $r \geq 2$ сформированы односоставные выборки Q_r и Q'_r , то $M(S^{(m) \setminus Q_r}) = M(S^{(m) \setminus Q'_r})$, следовательно, на шаге $r + 1$ достаточно найти минимальный элемент лишь для одной из этих выборок;

2) если в $S^{(m)}$ имеются одинаковые множества.

Оценим возможное снижение трудоёмкости в связи со вторым условием. Для системы уравнений (1) числа i и j , $1 \leq i, j \leq m$, назовём *эквивалентными*, если $S(i) = S(j)$; в этом случае пишем $i \sim j$.

Пусть u — число классов эквивалентности в $\{1, \dots, m\}$ отношения $\sim$, $1 \leq u \leq m$, которые обозначим через H_s , $h_s = |H_s|$, $s = 1, \dots, u$.

Маркеры Q и Q' назовём *эквивалентными*, если $\bar{n} = \bar{n}'$, т. е. одинаковы соответствующие им разбиения $\bar{n}, \bar{n}' \in R_n^m$; при этом пишем $Q \sim Q'$.

Теорема 3. При разбиении на классы эквивалентности

$$\{1, \dots, m\} = H_1 \cup \dots \cup H_u$$

число попарно неэквивалентных о. м. не превышает $\frac{m!}{h_1! \dots h_u!}$.

ДОКАЗАТЕЛЬСТВО. Пусть на r -м шаге алгоритма $A(S^{(m)})$ сформирована выборка $Q_r = (q_1, \dots, q_{r-1}, q_r)$, где $q_r \in H_s$ и $q_1, \dots, q_{r-1} \notin H_s$. Тогда на шагах $r+1, \dots, r+h_s-1$ дополним Q_r остальными элементами класса H_s . Это соответствует алгоритму $A(S^{(m)})$, так как из того, что $S(q_r) \in M(S^{(m)} \setminus Q_{r-1})$, следует $S(q) \in M(S^{(m)} \setminus Q_{r-1})$ для любого $q \in H_s \setminus \{q_r\}$. Тем самым при любом порядке выбора элементов из H_s на шагах $r, \dots, r+h_s-1$ алгоритма $A(S^{(m)})$ получаем эквивалентные маркеры. Отсюда вытекает справедливость оценки из условия теоремы. Теорема 3 доказана.

Следствие 1. Если отношение эквивалентности на $\{1, \dots, m\}$ нетривиально (не вырождается в равенство), то алгоритм $A(S^{(m)})$ сводится к поиску о. м. для подсистемы, в которой номера уравнений образуют трансверсал этого отношения (т. е. систему представителей классов эквивалентности).

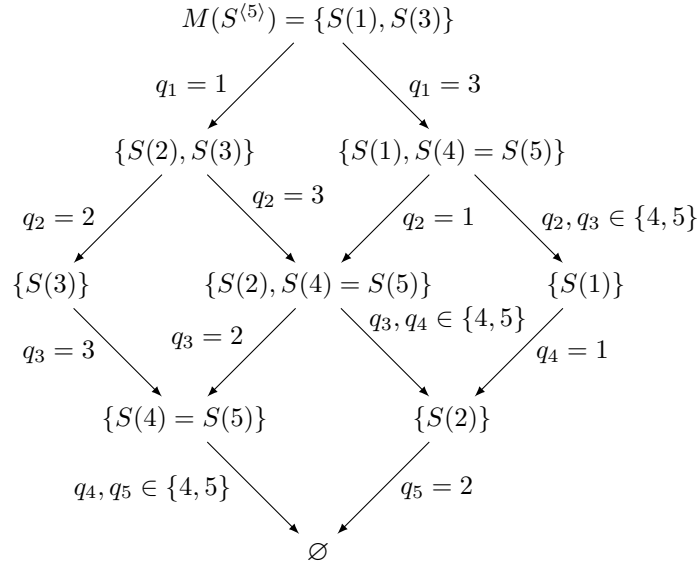
Замечание 2. Если на шагах $r \geq 1, r+1, \dots, r+h_s-1$ алгоритма $A(S^{(m)})$ элементы о. м. Q равны элементам множества H_s , $s \in \{1, \dots, u\}$, то $n_{r+1}(Q) = \dots = n_{r+h_s-1}(Q) = 0$.

Пример 1. Для класса систем пяти уравнений $f_j(x_1, \dots, x_6) = a_j$, $j = 1, \dots, 5$, определим о. м. и среднюю сложность решения при известном о. м. Множества $S(1), \dots, S(5)$ даны в табл. 1. Найдём о. м. $Q = (q_1, \dots, q_5)$ с учётом того, что $4 \sim 5$.

Таблица 1

Множества $S(j)$ системы функций

j	1	2	3	4	5
$S(j)$	$\{1, 2\}$	$\{1, 2, 4\}$	$\{3, 5\}$	$\{3, 4, 5\}$	$\{3, 4, 5\}$

Рис. 1. Схема алгоритма $A(S^{(m)})$ для системы из примера 1

Работа алгоритма $A(S^{(m)})$ представлена в виде схемы на рис. 1. В каждой вершине указано множество минимальных элементов $M(S^{(m)} \setminus Q_{r-1})$, рассматриваемое на текущем шаге r . На дугах отмечены элементы формируемых маркеров.

Таблица 2

Характеристики маркеров системы

№	Маркер Q	Разбиение числа 5	$T(Q)$
1	$(12345) \sim (12354)$	$2 + 1 + 2 + 0 + 0$	22
2	$(13452) \sim (13542)$	$2 + 2 + 1 + 0 + 0$	26
3	$(31245) \sim (31254)$	$2 + 2 + 1 + 0 + 0$	26
4	$(31452) \sim (31542)$	$2 + 2 + 1 + 0 + 0$	26
5	$(34512) \sim (35412)$	$2 + 1 + 0 + 2 + 0$	16

В табл. 2 приведены «кандидаты» в о. м., соответствующие им разбиения числа 5 и средняя вычислительная сложность а. п. о. для каждого маркера. В 5-й строке можно видеть два эквивалентных о. м. со сложностью $T(Q) = 16$. Отметим, что средняя сложность решения системы уравнений методом полного перебора равна 62.

Финансирование работы

Исследование выполнено за счёт бюджетов организаций, обозначенных автором на первой странице статьи. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
2. Коновальцев И. В. Об одном алгоритме решения систем линейных уравнений в конечных полях // Проблемы кибернетики. Т. 19. М.: Физматгиз, 1967. С. 269–274.
3. Strassen V. Gaussian elimination is not optimal // Numer. Math. 1969. V. 13. P. 354–356.
4. Courtois N. T. Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt // Information security and cryptology — ICISC 2002. Rev. Pap. 5th Int. Conf. (Seoul, Korea, Nov. 28–29, 2002). Heidelberg: Springer, 2003. P. 182–199. (Lect. Notes Comput. Sci.; V. 2587).
5. Фомичёв В. М. Методы дискретной математики в криптологии. М.: ДИАЛОГ-МИФИ, 2010. 424 с.
6. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002. 1040 с.
7. Merkle R. C., Hellman M. On the security of multiple encryption // Commun. ACM. 1981. V. 24, No. 7. P. 465–467.
8. Мелузов А. С. Построение эффективных алгоритмов решения систем полиномиальных булевых уравнений методом опробования части переменных // Дискрет. математика. 2011. Т. 23, № 4. С. 66–79.
9. Фомичёв В. М. Оценка характеристик нелинейности итеративных преобразований векторного пространства // Дискрет. анализ и исслед. операций. 2020. Т. 27, № 4. С. 131–151.
10. Rivest R. L. Cryptography // Handbook of theoretical computer science. V. A: Algorithms and complexity. Cambridge, MA: MIT Press, 1990. P. 617–755.
11. Stinson D. R. Cryptography: Theory and practice. Boca Raton: CRC Press, 1995. 434 p.

Фомичёв Владимир Михайлович

Статья поступила
30 марта 2023 г.
После доработки —
17 октября 2023 г.
Принята к публикации
22 декабря 2023 г.

ON THE COMPLEXITY OF THE SEQUENTIAL
SAMPLING METHODV. M. Fomichev^{1,2}¹ Security Code LLC,

10 Bld. 1 Pervyi Nagatinskii Driveway, 115230 Moscow, Russia

² Institute of Informatics Problems of FRC CSC RAS,

44 Bld. 2 Vavilov Street, 119333 Moscow, Russia

E-mail: fomichev.2016@yandex.ru

Abstract. A system of m Boolean equations can be solved by a sequential sampling method using an m -step algorithm, where at the i -th step the values of all variables essential for the first i equations are sampled and false solutions are rejected based on the right-hand sides parts of the equations, $i = 1, \dots, m$. The estimate of the complexity of the method depends on the structure of the sets of essential variables of the equations and attains its minimum after some permutation of the system equations. For the optimal permutation of equations we propose an algorithm that minimizes the average computational complexity of the algorithm under natural probabilistic assumptions. In a number of cases, the construction of such a permutation is computationally difficult; in this connection, other permutations are proposed which are computed in a simpler way but may lead to nonoptimal estimates of the complexity of the method. The results imply conditions under which the sequential sampling method degenerates into the exhaustive search method. An example of constructing an optimal permutation is given. Tab. 2, illustr. 1, bibliogr. 11.

Keywords: Boolean function, essential variable, subset lattice of a set, chain in a lattice.

References

1. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979; Mir, Moscow, 1982 [Russian]).

2. **I. V. Konovaltsev**, An algorithm for solving systems of linear equations in finite fields, in *Problems of Cybernetics*, Vol. 19 (Fizmatgiz, Moscow, 1967), pp. 269–274 [Russian].
3. **V. Strassen**, Gaussian elimination is not optimal, *Numer. Math.* **13**, 354–356 (1969).
4. **N. T. Courtois**, Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt, in *Information Security and Cryptology — ICISC 2002* (Rev. Pap. 5th Int. Conf., Seoul, Korea, Nov. 28–29, 2002) (Springer, Heidelberg, 2003), pp. 182–199 (Lect. Notes Comput. Sci., Vol. 2587).
5. **V. M. Fomichev**, *Methods of Discrete Mathematics in Cryptology* (DIALOG-MIFI, Moscow, 2010) [Russian].
6. **B. Schneier**, *Applied Cryptography. Protocols, Algorithms, and Source Code in C* (Wiley, Hoboken, NJ, 1993; TRIUMF, Moscow, 2002 [Russian]).
7. **R. C. Merkle** and **M. Hellman**, On the security of multiple encryption, *Commun. ACM* **24** (7), 465–467 (1981).
8. **A. S. Meluzov**, On construction of efficient algorithms for solving systems of polynomial Boolean equations by testing a part of variables, *Diskretn. Mat.* **23** (4), 66–79 (2011) [Russian] [*Discrete Math. Appl.* **21** (3), 381–395 (2011)].
9. **V. M. Fomichev**, Estimating nonlinearity characteristics for iterative transformations of a vector space, *Diskretn. Anal. Issled. Oper.* **27** (4), 131–151 (2020) [Russian] [*J. Appl. Ind. Math.* **14** (4), 610–622 (2020)].
10. **R. L. Rivest**, Cryptography, in *Handbook of Theoretical Computer Science*, Vol. A: Algorithms and Complexity (MIT Press, Cambridge, MA, 1990), pp. 617–755.
11. **D. R. Stinson**, *Cryptography: Theory and Practice* (CRC Press, Boca Raton, 1995).

Vladimir M. Fomichev

Received March 30, 2023

Revised October 17, 2023

Accepted December 22, 2023

ДИСКРЕТНЫЙ АНАЛИЗ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

2024. Том 31, № 2

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 15.07.2024 г.
Формат 70×100 1/16. Усл. печ. л. 12,7. Объем 2,03 МБ.

Издательство Института математики,
пр. Академика Коптюга, 4, 630090 Новосибирск, Россия