МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2025 № 68

Зарегистрирован в Федеральной службе по надзору
в сфере связи и массовых коммуникаций

Свидетельство о регистрации ПИ № ФС 77-33762 от 16 октября 2008 г.

Подписной индекс в объединённом каталоге «Пресса России» 38696

УЧРЕДИТЕЛЬ
Томский государственный университет

Адрес редакции и издателя: 634050, г. Томск, пр. Ленина, 36
E-mail: pank@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надёжности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

Редактор *Н. И. Шидловская*
Редактор-переводчик *Т. В. Бутузова*
Верстка *И. А. Панкратовой*

Подписано к печати 27.05.2025. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 14,3. Тираж 300 экз.
Заказ № 6352. Цена свободная. Дата выхода в свет 24.06.2025.

Отпечатано на оборудовании
Издательства Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел.: 8(3822)53-15-28, 52-98-49

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 512.563+519.85+517.518.244 DOI 10.17223/20710410/68/1

О ПОРЯДКЕ ГЛАДКОСТИ НАИМЕНЬШЕГО ВОГНУТОГО ПРОДОЛЖЕНИЯ БУЛЕВОЙ ФУНКЦИИ

Д. Н. Баротов*, Р. Н. Баротов**

*Финансовый университет при Правительстве РФ, г. Москва, Россия
**Худжандский государственный университет им. акад. Б. Гафурова, г. Худжанд, Таджикистан

E-mail: DNBarotov@fa.ru, ruzmet.tj@mail.ru

Исследуется порядок гладкости $f_{NR}(x_1, x_2, \ldots, x_n)$ — наименьшего вогнутого продолжения на $[0,1]^n$ произвольной булевой функции $f_B(x_1, x_2, \ldots, x_n)$. Доказано, что если булева функция $f_B(x_1, x_2, \ldots, x_n)$ существенно зависит не более чем от одной переменной, то на $[0,1]^n$ её наименьшее вогнутое продолжение $f_{NR}(x_1, x_2, \ldots, x_n)$ бесконечно дифференцируемо, в противном случае продолжение $f_{NR}(x_1, x_2, \ldots, x_n)$ на $[0,1]^n$ всего лишь непрерывно. Продемонстрировано применение наименьшего вогнутого продолжения к решению систем булевых уравнений.

Ключевые слова: *вогнутое продолжение булевой функции, булева функция, вогнутая функция, глобальная оптимизация, локальный экстремум.*

ON THE ORDER OF SMOOTHNESS OF THE SMALLEST CONCAVE EXTENSION OF A BOOLEAN FUNCTION

D. N. Barotov*, R. N. Barotov**

*Financial University under the Government of the Russian Federation, Moscow, Russia
**Khujand state university named after academician Bobojon Gafurov, Khujand, Tajikistan

In this paper, we study the order of smoothness of $f_{NR}(x_1, x_2, \ldots, x_n)$ — the least concave extension on $[0,1]^n$ of an arbitrary Boolean function $f_B(x_1, x_2, \ldots, x_n)$. We prove that if the Boolean function $f_B(x_1, x_2, \ldots, x_n)$ essentially depends on at most one variable, then on $[0,1]^n$ its least concave extension $f_{NR}(x_1, x_2, \ldots, x_n)$ is infinitely differentiable, otherwise the extension $f_{NR}(x_1, x_2, \ldots, x_n)$ on $[0,1]^n$ is only continuous. We demonstrate how the least concave extension can be used to solve systems of Boolean equations.

Keywords: *concave extension of a Boolean function, Boolean function, concave function, global optimization, local extremum.*

Введение

Различные труднорешаемые дискретные задачи, возникающие во многих областях, включая комбинаторику, современную кибернетику, биоинформатику, автоматизацию проектирования микроэлектроники, проектирование классических логических цепей, распознавание образов, функционирование конечных автоматов специального вида, а также криптографию, могут быть сведены к системам булевых уравнений [1–5]. Поэтому, с одной стороны, решению систем булевых уравнений посвящено значительное количество работ, разработано несколько направлений исследования и алгоритмов их решения, а, с другой стороны, в связи с тем, что задача решения системы булевых уравнений в общем случае является NP-трудной, в научном сообществе продолжает расти интерес к поиску новых алгоритмов как в классических, так и в квантовых моделях вычислений [6–9]. Одним из таких направлений является то, что задача решения системы булевых уравнений, в том числе путём представления некоторого вещественного продолжения (аналога) для каждой булевой функции, преобразуется в задачу с вещественными переменными, которая может быть либо задачей оптимизации некоторой функции, что позволяет применять оптимизационные методы вычислительной математики [10–14], либо задачей MILP или QUBO, решаемой классическими дискретными алгоритмами оптимизации или квантовыми алгоритмами [15, 16], либо системой полиномиальных уравнений, решаемой на множестве целых чисел [17], либо эквивалентной системой полиномиальных уравнений, решаемой и анализируемой символьными методами [18].

Отметим, что существует много способов, каждый из которых, используя вещественное продолжение булевой функции, выбранное на основе некоторого соображения, позволяет преобразовать систему булевых уравнений в задачу непрерывной оптимизации, так как принципиальное отличие таких способов от переборных алгоритмов локального поиска состоит в том, что на каждой итерации алгоритма сдвиг по градиенту (антиградиенту) производится по всем переменным одновременно [19]. Одна из основных проблем, возникающая при применении этих способов, заключается в том, что оптимизируемая целевая функция в искомой области может иметь множество локальных экстремумов, что значительно усложняет их практическое использование [14, 20, 21].

По изложенной проблеме в [14, 20–26] получены некоторые результаты, а именно: в [14, 20] рассмотрено конструирование полилинейного продолжения булевой функции и аргументировано, что задача решения произвольной системы булевых уравнений с n переменными может быть сведена к задаче непрерывной минимизации на $[0,1]^n$ целевой функции, не имеющей строгих локальных минимумов внутри любой k-мерной грани куба $[0,1]^n$, $k \in \{1,2,\ldots,n\}$, а в [21–25] построены выпуклые (вогнутые) продолжения булевых функций n переменных на $[0,1]^n$ и на основе построенных продолжений конструктивно доказано, что задача решения системы булевых уравнений может быть сведена к задаче минимизации (максимизации) целевой функции, любой локальный минимум (максимум) которой в искомой области является глобальным минимумом (максимумом), а также что для любой булевой функции от n переменных существует единственная вещественная функция, являющаяся максимумом (минимумом) среди всех её выпуклых (вогнутых) продолжений на $[0,1]^n$. В [26] проведено сравнительное исследование между выпуклыми, полилинейными и вогнутыми продолжениями булевых функций. Поэтому построение вещественных продолжений булевых функций, представляющих интерес при преобразовании систем булевых уравнений к задаче непрерывной оптимизации, и изучение их свойств также являются важными.

Данная работа посвящается исследованию порядка гладкости наименьшего вогнутого продолжения на $[0,1]^n$ произвольной булевой функции $f_B : \{0,1\}^n \to \{0,1\}$, представленного в [24], и является продолжением работ [14, 18, 20–26]. Установлен порядок дифференцируемости наименьшего вогнутого продолжения на $[0,1]^n$ произвольной булевой функции $f_B : \{0,1\}^n \to \{0,1\}$, а именно: во-первых, оценивая наименьшее вогнутое продолжение на $[0,1]^n$ произвольной булевой функции $f_B(x)$ с обеих сторон, доказано, что оно непрерывно на $[0,1]^n$; во-вторых, доказано, что если число существенных переменных булевой функции $f_B(x)$ меньше двух, то наименьшее вогнутое продолжение является бесконечно дифференцируемым, а иначе — лишь непрерывным.

1. Используемые определения и обозначения

Пусть $\mathbb{B}^n = \{(b_1, \ldots, b_n) : b_1, \ldots, b_n \in \{0,1\}\}$ — множество всевозможных двоичных слов (булевых векторов) длины n; $\mathbb{K}^n = \{(x_1, \ldots, x_n) : x_1, \ldots, x_n \in [0,1]\}$ — n-мерный куб, натянутый на булевы векторы длины n; $\mathrm{int}(\mathbb{K}^n) = \{(x_1, \ldots, x_n) : x_1, \ldots, x_n \in (0,1)\}$ — множество внутренних точек куба $\mathbb{K}^n$.

Определение 1. Отображение вида $f_B : \mathbb{B}^n \to \{0,1\}$ называется булевой функцией.

Определение 2. Переменная x_k, $k \in \{1, \ldots, n\}$, булевой функции $f_B(x_1, \ldots, x_n)$ называется существенной (функция $f_B(x_1, x_2, \ldots, x_n)$ существенно зависит от x_k), если

$$f_B(x_1, \ldots, x_{k-1}, 0, x_{k+1}, \ldots, x_n) \not\equiv f_B(x_1, \ldots, x_{k-1}, 1, x_{k+1}, \ldots, x_n).$$

Пусть $\mathbf{\Lambda}(x_1, x_2, \ldots, x_n) = \Big\{ \big(\lambda_{(0,0,\ldots,0)}, \lambda_{(0,0,\ldots,1)}, \ldots, \lambda_{(1,1,\ldots,1)}\big) \in \mathbb{K}^{2^n} :$

$$\sum_{(b_1,b_2,\ldots,b_n)\in\mathbb{B}^n} \lambda_{(b_1,b_2,\ldots,b_n)}(b_1, b_2, \ldots, b_n, 1) = (x_1, x_2, \ldots, x_n, 1)\Big\}$$

— множество весовых коэффициентов, используемых для представления точки $(x_1, x_2, \ldots, x_n)$ в виде выпуклой комбинации вершин куба $\mathbb{K}^n$.

Определение 3. Отображение вида $f : \mathbb{K}^n \to \mathbb{R}$ называется вогнутой функцией на $\mathbb{K}^n$, если для любых $x, y \in \mathbb{K}^n$ и любого $\alpha \in [0,1]$ выполняется

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

Определение 4. Отображение вида $f_C : \mathbb{K}^n \to \mathbb{R}$ назовём вогнутым продолжением на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \to \mathbb{B}$, если выполняются следующие два условия:

1) отображение f_C на $\mathbb{K}^n$ является вогнутой функцией;
2) имеет место равенство $f_C(b_1, \ldots, b_n) = f_B(b_1, \ldots, b_n)$ для всех $(b_1, \ldots, b_n) \in \mathbb{B}^n$.

Определение 5. Отображение вида $f_{NR} : \mathbb{K}^n \to \mathbb{R}$ назовём наименьшим среди всех вогнутых продолжений на $\mathbb{K}^n$ булевой функции $f_B : \mathbb{B}^n \to \mathbb{B}$, если выполняются следующие два условия:

1) отображение f_{NR} является вогнутым продолжением булевой функции f_B на $\mathbb{K}^n$;
2) для любого f_C — вогнутого продолжения на $\mathbb{K}^n$ булевой функции f_B — и любого $(x_1, \ldots, x_n) \in \mathbb{K}^n$ справедливо неравенство $f_{NR}(x_1, \ldots, x_n) \leqslant f_C(x_1, \ldots, x_n)$.

2. Установление порядка дифференцируемости наименьшего вогнутого продолжения на $\mathbb{K}^n$ произвольной булевой функции

Лемма 1. Для каждой булевой функции $f_B(x_1, x_2)$, которая существенно зависит от переменных x_1 и x_2, справедливо неравенство

$$f_B(0,0) - f_B(0,1) - f_B(1,0) + f_B(1,1) \neq 0. \tag{1}$$

Доказательство. Переменные x_1 и x_2 являются существенными для булевой функции $f_B(x_1, x_2)$ тогда и только тогда, когда

$$\begin{gathered}(f_B(0,0), f_B(0,1), f_B(1,0), f_B(1,1)) \in \big\{(0,0,0,1), (0,0,1,0), (0,1,0,0),\\ (0,1,1,0), (0,1,1,1), (1,0,0,0), (1,0,0,1), (1,0,1,1), (1,1,0,1), (1,1,1,0)\big\}.\end{gathered} \tag{2}$$

Легко заметить, что из (2) следует справедливость (1). ■

В [24] доказано, что для произвольной булевой функции $f_B(x_1, \ldots, x_n)$ вещественная функция

$$f_{NR}(x_1, \ldots, x_n) = \max_{\lambda \in \mathbf{\Lambda}(x_1,\ldots,x_n)} \left[\sum_{(b_1,\ldots,b_n)\in \mathbb{B}^n} \lambda_{(b_1,\ldots,b_n)} f_B(b_1, \ldots, b_n) \right] \tag{3}$$

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

Вообще говоря, ограниченная вогнутая (выпуклая) функция, определённая на множестве $\mathbb{K}^n$, непрерывна во внутренних точках множества $\mathbb{K}^n$ и разрывна в его граничных точках. В качестве иллюстрирующего примера приведём вещественную разрывную вогнутую функцию

$$f_C(x) = \begin{cases} 0, & \text{если } x \in \{0, 1\}, \\ 1, & \text{если } x \in (0, 1), \end{cases}$$

которая также является вогнутым продолжением на $[0, 1]$ булевой функции $f_B(x) = 0$. Но в нашем случае вещественная функция $f_{NR}(x_1, \ldots, x_n)$, являющаяся наименьшим вогнутым продолжением на $\mathbb{K}^n$ булевой функции $f_B(x_1, \ldots, x_n)$, непрерывна на $\mathbb{K}^n$ для каждого натурального n. Ниже, ради полноты изложения, путём построения двусторонней оценки мы предъявим полное доказательство непрерывности $f_{NR}(x_1, \ldots, x_n)$ для произвольного натурального n и установим порядок дифференцируемости $f_{NR}(x_1, \ldots, x_n)$.

Теорема 1. Функция $f_{NR}(x_1, \ldots, x_n)$, определённая формулой (3), на $\mathbb{K}^n$ непрерывна.

Доказательство. Индукция по n.

Б а з а и н д у к ц и и . Согласно [24, следствия 1 и 2], имеем, что, во-первых, для любой булевой функции $f_B(x)$, зависящей от одной переменной, вещественная функция

$$f_{NR}(x) = (1 - x) f_B(0) + x\, f_B(1) \tag{4}$$

является единственным наименьшим среди всех её вогнутых продолжений на $\mathbb{K}$; во-вторых, для любой булевой функции $f_B(x, y)$, зависящей от двух переменных, вещественная функция вида

$$\begin{gathered} f_{NR}(x, y) = (1 - x - y) f_B(0, 0) + x\, f_B(1, 0) + y\, f_B(0, 1) + \\ + \frac{f_B(0,0) - f_B(0,1) - f_B(1,0) + f_B(1,1)}{4} (2x + 2y - 1 - |x - y| + |x + y - 1|) - \\ - \frac{|f_B(0,0) - f_B(0,1) - f_B(1,0) + f_B(1,1)|}{4} (|x - y| + |x + y - 1| - 1) \end{gathered} \tag{5}$$

является единственным наименьшим среди всех её вогнутых продолжений на $\mathbb{K}^2$. Непрерывность функций $f_{NR}(x)$ и $f_{NR}(x, y)$, ввиду (4) и (5), очевидна.

П р е д п о л о ж е н и е и н д у к ц и и . Пусть при $n = k$ для произвольной булевой функции $f_B(x_1, \ldots, x_k)$ наименьшее вогнутое продолжение $f_{NR}(x_1, \ldots, x_k)$ непрерывно на $\mathbb{K}^k$.

Ш а г и н д у к ц и и . Докажем, что функция $f_{NR}(x_1, \ldots, x_{k+1})$, являющаяся наименьшим вогнутым продолжением на $\mathbb{K}^{k+1}$ булевой функции $f_B(x_1, \ldots, x_{k+1})$, непрерывна на $\mathbb{K}^{k+1}$. Пусть $(x_1^*, \ldots, x_{k+1}^*)$ — произвольная точка куба $\mathbb{K}^{k+1}$. Покажем, что имеет место равенство

$$\lim_{(\Delta x_1, \ldots, \Delta x_{k+1}) \to (0, \ldots, 0)} f_{NR}(x_1^* + \Delta x_1, \ldots, x_{k+1}^* + \Delta x_{k+1}) = f_{NR}(x_1^*, \ldots, x_{k+1}^*). \tag{6}$$

Рассмотрим два случая.

С л у ч а й 1 . Пусть $(x_1^*, \ldots, x_{k+1}^*) \in \operatorname{int}(\mathbb{K}^{k+1})$. Ввиду открытости множества $\operatorname{int}(\mathbb{K}^{k+1})$ доказательство можно провести по известной схеме, например [27].

С л у ч а й 2 . Пусть $(x_1^*, \ldots, x_{k+1}^*) \in \partial(\mathbb{K}^{k+1}) = \mathbb{K}^{k+1} \setminus \operatorname{int}(\mathbb{K}^{k+1})$. Тогда существует $i \in \{1, \ldots, k+1\}$, такое, что $x_i^* \in \{0, 1\}$. Согласно предположению индукции, функции, полученные путём сужения, вида

$$\begin{gathered} f_0(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1}) = f_{NR}(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_{k+1}) = \\ = \max_{\substack{\lambda \in \mathbf{\Lambda}(x_1, \ldots, x_{i-1}, \\ x_{i+1}, \ldots, x_{k+1})}} \left[\sum_{\substack{(b_1, \ldots, b_{i-1}, \\ b_{i+1}, \ldots, b_{k+1}) \in \mathbb{B}^k}} \lambda_{(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1})} f_B(b_1, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_{k+1}) \right] \end{gathered} \tag{7}$$

и

$$\begin{gathered} f_1(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1}) = f_{NR}(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_{k+1}) = \\ = \max_{\substack{\lambda \in \mathbf{\Lambda}(x_1, \ldots, x_{i-1}, \\ x_{i+1}, \ldots, x_{k+1})}} \left[\sum_{\substack{(b_1, \ldots, b_{i-1}, \\ b_{i+1}, \ldots, b_{k+1}) \in \mathbb{B}^k}} \lambda_{(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1})} f_B(b_1, \ldots, b_{i-1}, 1, b_{i+1}, \ldots, b_{k+1}) \right] \end{gathered} \tag{8}$$

непрерывны на $\mathbb{K}^k$. Докажем, что вещественная непрерывная функция вида

$$\begin{gathered} g(x_1, \ldots, x_{k+1}) = \min(1 - x_i, f_0(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1})) + \\ + \min(x_i, f_1(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1})) \end{gathered} \tag{9}$$

является вогнутым продолжением на $\mathbb{K}^{k+1}$ булевой функции $f_B(x_1, \ldots, x_{k+1})$. Для этого достаточно показать справедливость следующих двух свойств:

1) $g(b_1, \ldots, b_{k+1}) = f_B(b_1, \ldots, b_{k+1})$ для всех $(b_1, \ldots, b_{k+1}) \in \mathbb{B}^{k+1}$;
2) функция $g(x_1, \ldots, x_{k+1})$ на $\mathbb{K}^{k+1}$ является вогнутой.

Обоснуем эти свойства:

1) Для всех $(b_1, \ldots, b_{k+1}) \in \mathbb{B}^{k+1}$ имеем

$$\begin{gathered} g(b_1, \ldots, b_{k+1}) = \min(1 - b_i, f_0(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1})) + \\ + \min(b_i, f_1(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1})) = \\ = (1 - b_i) f_0(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1}) + b_i\, f_1(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{k+1}) = \\ = (1 - b_i) f_{NR}(b_1, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_{k+1}) + b_i\, f_{NR}(b_1, \ldots, b_{i-1}, 1, b_{i+1}, \ldots, b_{k+1}) = \\ = \overline{b_i}\, f_{NR}(b_1, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_{k+1}) \oplus b_i\, f_{NR}(b_1, \ldots, b_{i-1}, 1, b_{i+1}, \ldots, b_{k+1}) = f_B(b_1, \ldots, b_{k+1}). \end{gathered}$$

2) Функции $f_0(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1})$ и $f_1(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1})$, ввиду (7) и (8), на $\mathbb{K}^k$ являются вогнутыми и, следовательно, для любых $x, y \in \mathbb{K}^{k+1}$ и любого $\alpha \in [0, 1]$ имеем

$$g(\alpha x + (1 - \alpha) y) = \min \big(1 - (\alpha x_i + (1 - \alpha) y_i), f_0(\alpha x_1 + (1 - \alpha) y_1, \ldots, \alpha x_{i-1} + (1 - \alpha) y_{i-1},$$

$$\alpha x_{i+1}+(1-\alpha)y_{i+1},\dots,\alpha x_{k+1}+(1-\alpha)y_{k+1})\big)+\min\big(\alpha x_i+(1-\alpha)y_i, f_1(\alpha x_1+(1-\alpha)y_1,$$
$$\dots,\alpha x_{i-1}+(1-\alpha)y_{i-1},\alpha x_{i+1}+(1-\alpha)y_{i+1},\dots,\alpha x_{k+1}+(1-\alpha)y_{k+1})\big)\geqslant$$
$$\geqslant\min\big(\alpha(1-x_i)+(1-\alpha)(1-y_i),\alpha f_0(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{k+1})+$$
$$+(1-\alpha)f_0(y_1,\dots,y_{i-1},y_{i+1},\dots,y_{k+1})\big)+\min\big(\alpha x_i+(1-\alpha)y_i,$$
$$\alpha f_1(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{k+1})+(1-\alpha)f_1(y_1,\dots,y_{i-1},y_{i+1},\dots,y_{k+1})\big)\geqslant$$
$$\geqslant\alpha\min(1-x_i,f_0(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{k+1}))+(1-\alpha)\min(1-y_i,f_0(y_1,\dots,y_{i-1},y_{i+1},$$
$$\dots,y_{k+1}))+\alpha\min(x_i,f_1(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{k+1}))+$$
$$+(1-\alpha)\min(y_i,f_1(y_1,\dots,y_{i-1},y_{i+1},\dots,y_{k+1}))=\alpha g(x)+(1-\alpha)g(y).$$

Далее, с одной стороны, ввиду определения наименьшего вогнутого продолжения, для $(x^*+\Delta x)\in\mathbb{K}^{k+1}$ выполняется

$$f_{NR}(x^*+\Delta x)-f_{NR}(x^*)\leqslant g(x^*+\Delta x)-f_{NR}(x^*),\tag{10}$$

а, с другой стороны, ввиду вогнутости функции $f_{NR}(x)$, для $(x^*+\Delta x)\in\mathbb{K}^{k+1}$ имеем

$$\begin{gathered}f_{NR}(x^*+\Delta x)-f_{NR}(x^*)=f_{NR}\big((1-(1-2x_i^*)\Delta x_i)(x_1^*+\Delta x_1,\dots,x_{i-1}^*+\Delta x_{i-1},\\ x_i^*,x_{i+1}^*+\Delta x_{i+1},\dots,x_{k+1}^*)+(1-2x_i^*)\Delta x_i(x_1^*+\Delta x_1,\dots,x_{i-1}^*+\Delta x_{i-1},1-x_i^*,\\ x_{i+1}^*+\Delta x_{i+1},\dots,x_{k+1}^*)\big)-f_{NR}(x^*)\geqslant\\ \geqslant(1-(1-2x_i^*)\Delta x_i)f_{NR}(x_1^*+\Delta x_1,\dots,x_{i-1}^*+\Delta x_{i-1},x_i^*,x_{i+1}^*+\Delta x_{i+1},\dots,x_{k+1}^*)+\\ +(1-2x_i^*)\Delta x_i\,f_{NR}(x_1^*+\Delta x_1,\dots,x_{i-1}^*+\Delta x_{i-1},1-x_i^*,\\ x_{i+1}^*+\Delta x_{i+1},\dots,x_{k+1}^*)-f_{NR}(x^*).\end{gathered}\tag{11}$$

Ввиду (9) и непрерывности функций $f_0(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{k+1})$ и $f_1(x_1,\dots,x_{i-1},$ $x_{i+1},\dots,x_{k+1})$, переходя к пределу в оценках (10) и (11) при $(\Delta x_1,\dots,\Delta x_{k+1})\to$ $\to(0,\dots,0)$, получим (6). ■

Теорема 2. Если у булевой функции $f_B(x_1,\dots,x_n)$ не меньше двух существенных переменных, то вещественная функция $f_{NR}(x_1,\dots,x_n)$, которая является её наименьшим вогнутым продолжением на $\mathbb{K}^n$, не дифференцируема на $\mathbb{K}^n$.

Доказательство. Не теряя общности, будем считать, что все переменные функции $f_B(x_1,\dots,x_n)$ являются существенными. Докажем от противного: пусть вещественная функция $f_{NR}(x_1,\dots,x_n)$, которая является наименьшим вогнутым продолжением на $\mathbb{K}^n$ булевой функции $f_B(x_1,\dots,x_n)$, является дифференцируемой в каждой точке $(x_1^*,\dots,x_n^*)$ куба $\mathbb{K}^n$ при $n\geqslant 2$. Тогда для всех $(b_1,\dots,b_{i-1},$ $b_{i+1},\dots,b_{j-1},b_{j+1},\dots,b_n)\in\mathbb{B}^{n-2}$ суженная вещественная функция $f_{NR}(b_1,\dots,b_{i-1},x_i,$ $b_{i+1},\dots,b_{j-1},x_j,b_{j+1},\dots,b_n)$ является дифференцируемой в каждой точке (x_i^*,x_j^*) квадрата $\mathbb{K}^2$. Согласно [28], существует $(b_1^*,\dots,b_{i-1}^*,b_{i+1}^*,\dots,b_{j-1}^*,b_{j+1}^*,\dots,b_n^*)\in\mathbb{B}^{n-2}$, такой, что переменные x_i и x_j суженной булевой функции $f_B(b_1^*,\dots,b_{i-1}^*,x_i,b_{i+1}^*,\dots,b_{j-1}^*,x_j,$ $b_{j+1}^*,\dots,b_n^*)$ являются существенными. Отсюда получим, что вещественная функция

$$g_{NR}(x_i,x_j)=f_{NR}(b_1^*,\dots,b_{i-1}^*,x_i,b_{i+1}^*,\dots,b_{j-1}^*,x_j,b_{j+1}^*,\dots,b_n^*),$$

которая является наименьшим вогнутым продолжением на $\mathbb{K}^2$ булевой функции

$$g_B(x_i,x_j)=f_B(b_1^*,\dots,b_{i-1}^*,x_i,b_{i+1}^*,\dots,b_{j-1}^*,x_j,b_{j+1}^*,\dots,b_n^*),$$

является дифференцируемой в каждой точке (x_i^*, x_j^*) квадрата $\mathbb{K}^2$. Теперь, с одной стороны, ввиду леммы 1, имеем, что

$$g_B(0,0) - g_B(0,1) - g_B(1,0) + g_B(1,1) \neq 0, \tag{12}$$

а, с другой стороны, ввиду [24, следствие 2], получаем

$$\begin{gathered} g_{NR}(x_i, x_j) = (1-x_i-x_j)g_B(0,0)+x_i\, g_B(1,0)+x_j\, g_B(0,1)+ \\ +\frac{g_B(0,0)-g_B(0,1)-g_B(1,0)+g_B(1,1)}{4}\,(2x_i+2x_j-1-|x_i-x_j|+|x_i+x_j-1|)- \\ -\frac{|g_B(0,0)-g_B(0,1)-g_B(1,0)+g_B(1,1)|}{4}\,(|x_i-x_j|+|x_i+x_j-1|-1). \end{gathered} \tag{13}$$

В силу (12) из (13) следует, что функция $g_{NR}(x_i, x_j)$ не является дифференцируемой на $\mathbb{K}^2$, т. е. не является дифференцируемой в каждой точке (x_i^*, x_j^*) квадрата $\mathbb{K}^2$. Полученное противоречие завершает доказательство теоремы 2. ■

Проиллюстрируем одно из возможных применений наименьшего вогнутого продолжения к решению систем булевых уравнений на примере системы из двух уравнений

$$\begin{cases} p_1(x,y) = x \cdot y \oplus x = 1, \\ p_2(x,y) = x \oplus y = 1, \end{cases} \tag{14}$$

приведённой в [14], т. е. проиллюстрируем общий метод, развитый в [24]. Систему (14) на основе (5) преобразуем в систему вогнутых уравнений вида

$$\begin{cases} f_1(x,y) = \dfrac{1}{2}\,(x - y + 1 - |x + y - 1|) = 1, \\ f_2(x,y) = 1 - |x + y - 1| = 1, \end{cases} \tag{15}$$

где $f_k(x,y)$ — наименьшее вогнутое продолжение на $\mathbb{K}^2$ функции $p_k(x,y)$, $k \in \{1,2\}$. В свою очередь, для системы (15) конструируем максимизируемую вогнутую целевую функцию вида

$$f(x,y) = f_1(x,y) + f_2(x,y). \tag{16}$$

Ввиду (3) имеем, что для всех $(x,y) \in \mathbb{K}^2$ и $k \in \{1,2\}$ имеют место неравенства

$$0 \leqslant f_k(x,y) \leqslant 1. \tag{17}$$

Из (16) и (17) получаем, что $(x^*, y^*) \in \mathbb{K}^2$ — решение системы (15) тогда и только тогда, когда $\max\limits_{(x,y)\in\mathbb{K}^2} f(x,y) = f(x^*, y^*) = 2$. На рис. 1 приведён график функции $f(x,y)$; нетрудно заметить, что $(x^*, y^*) = (1,0)$ — единственная точка максимума функции $f(x,y)$ на $\mathbb{K}^2$.

Отметим, что если при преобразовании для каждой булевой функции брать в качестве вогнутого продолжения не наименьшее, а, например,

$$\tilde{f}_1(x,y) = \frac{1}{2}(x - y - |x - y|) + 1 - |x - 1 + y| \quad \text{и} \quad \tilde{f}_2(x,y) = 1 - |x + y - 1|,$$

то система вогнутых уравнений

$$\begin{cases} \tilde{f}_1(x,y) = 1, \\ \tilde{f}_2(x,y) = 1 \end{cases} \tag{18}$$

будет иметь решение, не являющееся решением булевой системы (14), т. е. целевая функция

$$\tilde{f}(x,y) = \tilde{f}_1(x,y) + \tilde{f}_2(x,y)$$

имеет точку максимума, например $(3/4, 1/4)$, которая является решением системы (18), но не является решением булевой системы (14) (рис. 2).

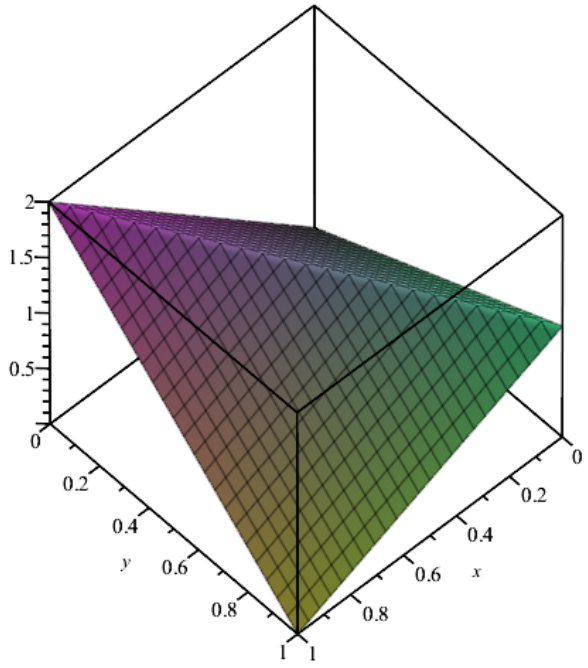


Рис. 1. График функции $f(x,y)$

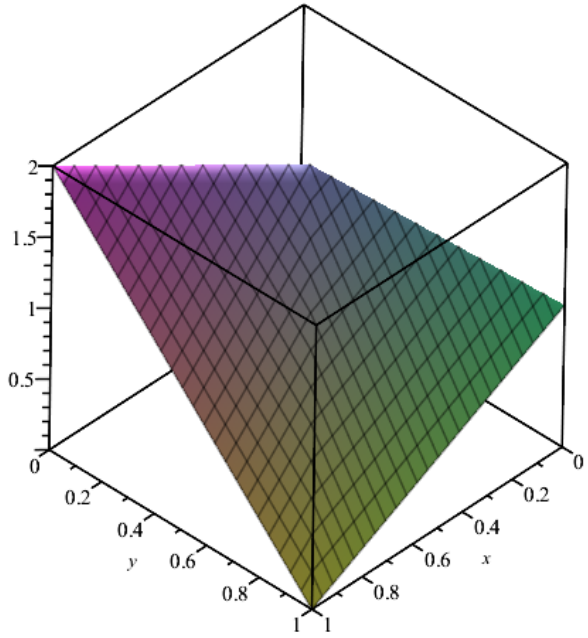


Рис. 2. График функции $\tilde{f}(x,y)$

Заключение

Таким образом, ввиду формулы (4) и теорем 1 и 2 имеем, что если число существенных переменных произвольной булевой функции не меньше двух, то её наименьшее вогнутое продолжение на $\mathbb{K}^n$ непрерывно, но не дифференцируемо на $\mathbb{K}^n$, а если меньше двух, то линейно и, следовательно, бесконечно дифференцируемо на $\mathbb{K}^n$.

Авторы выражают благодарность рецензентам за полезные замечания, исправление которых позволило улучшить содержание статьи.

ЛИТЕРАТУРА

1. *Заикин О. С., Семёнов А. А., Посыпкин М. А.* Процедуры построения декомпозиционных множеств для распределенного решения SAT-задач в проекте добровольных вычислений SAT@home // Управление большими системами. 2013. Вып. 43. С. 138–156.
2. *Заикин О. С., Семенов А. А.* Применение метода Монте-Карло к прогнозированию времени параллельного решения проблемы булевой выполнимости // Выч. мет. программирование. 2014. Т. 15. № 1. С. 22–35.
3. *Мальцева М. А., Румянцев А. С.* Проверка выполнимости булевых формул с помощью квантового отжига // Труды Карельского научного центра РАН. 2023. № 4. С. 41–49.
4. *Леонтьев В. К., Нурлыбаев А. Н.* Об одном классе систем булевых уравнений // Ж. вычисл. матем. и матем. физ. 1975. Т. 15. № 6. С. 1568–1579.
5. *Леонтьев В. К., Тоноян Г. П.* О системах булевых уравнений // Ж. вычисл. матем. и матем. физ. 2013. Т. 53. № 5. С. 800–807.

6. *Ramos-Calderer S., Bravo-Prieto C., Lin R., et al.* Solving systems of Boolean multivariate equations with quantum annealing // Phys. Rev. Res. 2022. V. 4. No. 1. Paper 013096.
7. *Bennakhi A., Byrd G. T., and Franzon P.* Solving the B-SAT problem using quantum computing: Smaller is sometimes better // Entropy. 2024. V. 26. No. 10. Paper 875.
8. *Леонтьев В. К., Тоноян Г. П.* Приближенные решения систем булевых уравнений // Ж. вычисл. матем. и матем. физ. 1993. Т. 33. № 9. С. 1383–1390.
9. *Леонтьев В. К., Гордеев Э. Н.* О числе решений системы булевых уравнений // Автомат. и телемех. 2021. № 9. С. 150–168.
10. *Gu J.* How to Solve Very Large-Scale Satisfiability (VLSS) Problems. Technical Report UCECETR-90-002. Calgary: University of Calgary, 1990.
11. *Gu J.* On optimizing a search problem // N. G. Burbakis (ed). Artificial Intelligence Methods and Applications. 1992. P. 63–105.
12. *Gu J.* Global optimization for satisfiability (SAT) problem // IEEE Trans. Knowledge Data Engin. 1994. V. 6. No. 3. P. 361–381.
13. *Gu J., Gu Q., and Du D.* On optimizing the satisfiability (SAT) problem // J. Computer Sci. Technology. 1999. V. 14. No. 1. P. 1–17.
14. *Barotov D. N.* Target function without local minimum for systems of logical equations with a unique solution // Mathematics. 2022. V. 10. No. 12:2097.
15. *Pakhomchik A. I., Voloshinov V. V., Vinokur V. M., and Lesovik G. B.* Converting of Boolean expression to linear equations, inequalities and QUBO penalties for cryptanalysis // Algorithms. 2022. V. 15. No. 2:33.
16. *Burek E., Wronski M., Mank K., and Misztal M.* Algebraic attacks on block ciphers using quantum annealing // IEEE Trans. Emerging Topics in Computing. 2022. V. 10. No. 2. P. 678–689.
17. *Abdel-Gawad A. H., Atiya A. F., and Darwish N. M.* Solution of systems of Boolean equations via the integer domain // Inform. Sci. 2010. V. 180. No. 2. P. 288–300.
18. *Barotov D. N., Barotov R. N., Soloviev V., et al.* The development of suitable inequalities and their application to systems of logical equations // Mathematics. 2022. V. 10. No. 11:1851.
19. *Файзуллин Р. Т., Дулькейт В. И., Огородников Ю. Ю.* Гибридный метод поиска приближенного решения задачи 3-выполнимость, ассоциированной с задачей факторизации // Труды Института математики и механики УрО РАН. 2013. Т. 19. № 2. С. 285–294.
20. *Баротов Д. Н., Баротов Р. Н.* Полилинейные продолжения некоторых дискретных функций и алгоритм их нахождения // Выч. мет. программирование. 2023. Т. 24. № 1. С. 10–23.
21. *Баротов Д. Н., Баротов Р. Н.* Конструирование гладких выпуклых продолжений булевых функций // Вестник российских университетов. Математика. 2024. Т. 29. № 145. С. 20–28.
22. *Баротов Д. Н.* Выпуклое продолжение булевой функции и его приложения // Дискрет. анализ исслед. опер. 2024. Т. 31. № 1. С. 5–18.
23. *Баротов Д. Н.* О существовании и свойствах выпуклых продолжений булевых функций // Матем. заметки. 2024. Т. 115. № 4. С. 533–551.
24. *Баротов Д. Н.* Вогнутые продолжения булевых функций и некоторые их свойства и приложения // Изв. Иркут. ун-та. Сер. Математика. 2024. Т. 49. С. 105–123.
25. *Баротов Д. Н.* Выпуклые продолжения некоторых дискретных функций // Дискрет. анализ исслед. опер. 2024. Т. 31. № 3. С. 5–23.
26. *Баротов Д. Н., Судаков В. А.* О неравенствах между выпуклыми, вогнутыми и полилинейными продолжениями булевых функций // Препринты ИПМ им. М. В. Келдыша. 2024. № 30. С. 1–13.

27. *Экланд И., Темам Р.* Выпуклый анализ и вариационные проблемы. М.: Мир, 1979. 399 с.
28. *Salomaa A.* On essential variables of functions, especially in the algebra of logic // Ann. Acad. Sci. Fenn. Ser. AI Math. 1963. No. 339. P. 1–11.
29. *Jensen J. L. W. V.* Sur les fonctions convexes et les inegalites entre les valeurs moyennes // Acta Mathematica. 1906. V. 30. No. 1. P. 175–193.

REFERENCES

1. *Zaikin O. S., Semenov A. A., and Posypkin M. A.* Protsedury postroeniya dekompozitsionnykh mnozhestv dlya raspredelennogo resheniya SAT-zadach v proekte dobrovol'nykh vychisleniy SAT@home [Constructing decomposition sets for distributed solution of SAT problems in volunteer computing project SAT@home]. Upravlenie Bol'shimi Sistemami, 2013, iss. 43, pp. 138–156. (in Russian)
2. *Zaikin O. S. and Semenov A. A.* Primenenie metoda Monte-Karlo k prognozirovaniyu vremeni parallel'nogo resheniya problemy bulevoy vypolnimosti [Application of the Monte Carlo method for estimating the total time of solving the SAT problem in parallel]. Vychislitel'nye Metody i Programmirovanie, 2014, vol. 15, no. 1, pp. 22–35. (in Russian)
3. *Maltseva M. A. and Rumyantsev A. S.* Proverka vypolnimosti bulevykh formul s pomoshch'yu kvantovogo otzhiga [Boolean satisfiability verification by quantum annealing]. Trudy Karel'skogo nauchnogo tsentra RAN, 2023, no. 4, pp. 41–49. (in Russian)
4. *Leont'ev V. K. and Nurlybaev A. N.* A certain class of systems of Boolean equations. U.S.S.R. Comput. Math. Math. Phys., 1975, vol. 15, no. 6, pp. 198–210.
5. *Leont'ev V. K. and Tonoyan G. P.* On systems of Boolean equations. Comput. Math. Math. Phys., 2013, vol. 53, no. 5, pp. 632–639.
6. *Ramos-Calderer S., Bravo-Prieto C., Lin R., et al.* Solving systems of Boolean multivariate equations with quantum annealing. Phys. Rev. Res., 2022, vol. 4, no. 1, paper 013096.
7. *Bennakhi A., Byrd G. T., and Franzon P.* Solving the B-SAT problem using quantum computing: Smaller is sometimes better. Entropy, 2024, vol. 26, no. 10, paper 875.
8. *Leont'ev V. K. and Tonoyan G. P.* Approximate solutions of systems of Boolean equations. Comput. Math. Math. Phys., 1993, vol. 33, no. 9, pp. 1221–1227.
9. *Leont'ev V. K. and Gordeev E. N.* On the number of solutions to a system of Boolean equations. Autom. Remote Control, 2021, vol. 82, no. 9, pp. 1581–1596.
10. *Gu J.* How to Solve Very Large-Scale Satisfiability (VLSS) Problems. Technical Report UCECETR-90-002, Calgary, University of Calgary, 1990.
11. *Gu J.* On optimizing a search problem. N. G. Burbakis (ed). Artificial Intelligence Methods and Applications, 1992, pp. 63–105.
12. *Gu J.* Global optimization for satisfiability (SAT) problem. IEEE Trans. Knowledge Data Engin., 1994, vol. 6, no. 3, pp. 361–381.
13. *Gu J., Gu Q., and Du D.* On optimizing the satisfiability (SAT) problem. J. Computer Sci. Technology, 1999, vol. 14, no. 1, pp. 1–17.
14. *Barotov D. N.* Target function without local minimum for systems of logical equations with a unique solution. Mathematics, 2022, vol. 10, no. 12:2097.
15. *Pakhomchik A. I., Voloshinov V. V., Vinokur V. M., and Lesovik G. B.* Converting of Boolean expression to linear equations, inequalities and QUBO penalties for cryptanalysis. Algorithms, 2022, vol. 15, no. 2:33.
16. *Burek E., Wronski M., Mank K., and Misztal M.* Algebraic attacks on block ciphers using quantum annealing. IEEE Trans. Emerging Topics in Computing, 2022, vol. 10, no. 2, pp. 678–689.

17. *Abdel-Gawad A. H., Atiya A. F., and Darwish N. M.* Solution of systems of Boolean equations via the integer domain. Inform. Sci., 2010, vol. 180, no. 2, pp. 288–300.
18. *Barotov D. N., Barotov R. N., Soloviev V., et al.* The development of suitable inequalities and their application to systems of logical equations. Mathematics, 2022, vol. 10, no. 11:1851.
19. *Faizullin R. T., Dulkeyt V. I., and Ogorodnikov Y. Y.* Gibridnyy metod poiska priblizhennogo resheniya zadachi 3-vypolnimost', assotsiirovannoy s zadachey faktorizatsii [Hybrid method for the approximate solution of the 3-satisfiability problem associated with the factorization problem]. Tr. Inst. Mat. Mekh., 2013, vol. 19, no. 2, pp. 285–294. (in Russian)
20. *Barotov D. N. and Barotov R. N.* Polilineynyye prodolzheniya nekotorykh diskretnykh funktsiy i algoritm ikh nakhozhdeniya [Polylinear continuations of some discrete functions and an algorithm for finding them]. Vychislitel'nye Metody i Programmirovanie, 2023, vol. 24, pp. 10–23. (in Russian)
21. *Barotov D. N. and Barotov R. N.* Konstruirovaniye gladkikh vypuklykh prodolzheniy bulevykh funktsiy [Construction of smooth convex extensions of Boolean functions]. Vestnik Rossiyskikh Universitetov. Matematika, 2024, vol. 29, no. 145, pp. 20–28. (in Russian)
22. *Barotov D. N.* Convex continuation of a Boolean function and its applications. J. Appl. Industr. Math., 2024, vol. 18, no. 1, pp. 1–9.
23. *Barotov D. N.* On the existence and properties of convex extensions of Boolean functions. Math. Notes, 2024, vol. 115, no. 4, pp. 489–505.
24. *Barotov D. N.* Vognutyye prodolzheniya bulevykh funktsiy i nekotoryye ikh svoystva i prilozheniya [Concave continuations of Boolean functions and some of their properties and applications]. Izvestiya Irkutskogo Gosuniversiteta. Ser. Matematika, 2024, vol. 49, pp. 105–123. (in Russian)
25. *Barotov D. N.* Convex continuations of some discrete functions. J. Appl. Industr. Math., 2024, vol. 18, no. 3, pp. 412–423.
26. *Barotov D. N. and Sudakov V. A.* O neravenstvakh mezhdu vypuklymi, vognutymi i polilineynymi prodolzheniyami bulevykh funktsiy [On inequalities between convex, concave, and multilinear continuations of Boolean functions]. Preprinty IPM im. M. V. Keldysha, 2024, no. 30, pp. 1–13. (in Russian)
27. *Ekeland I. and Temam R.* Convex Analysis and Variational Problems. Amsterdam, North-Holland, 1976. 402 p.
28. *Salomaa A.* On essential variables of functions, especially in the algebra of logic. Ann. Acad. Sci. Fenn. Ser. AI Math., 1963, no. 339, pp. 1–11.
29. *Jensen J. L. W. V.* Sur les fonctions convexes et les inegalites entre les valeurs moyennes. Acta Mathematica, 1906, vol. 30, no. 1, pp. 175–193. (in French)

УДК 519.716.32+512.547 DOI 10.17223/20710410/68/2

КРИВИЗНА НЕКОТОРЫХ КЛАССОВ БУЛЕВЫХ ФУНКЦИЙ

А. А. Панпурин

РТУ МИРЭА, г. Москва, Россия

E-mail: aa.panpurin@yandex.ru

Исследуется кривизна различных классов булевых функций, построенных с помощью суперпозиции, симметрических многочленов и бент-функций. Получаются оценки и точные значения для коэффициентов Уолша — Адамара, кривизны и нелинейности рассматриваемых классов булевых функций. Устанавливается связь кривизны и нелинейности произвольных булевых функций.

Ключевые слова: *булевы функции, бент-функции, кривизна булевой функции, нелинейность булевой функции.*

CURVATURE OF SOME CLASSES OF BOOLEAN FUNCTIONS

A. A. Panpurin

MIREA — Russian Technological University, Moscow, Russia

The curvature $\sigma(f)$ of Boolean function f is defined as the sum of the absolute values of its Walsh coefficients. In the paper, the curvature of various classes of Boolean functions constructed using superposition, symmetric polynomials and bent functions is investigated. Estimates and exact values have been obtained for the Walsh coefficients, curvature, and nonlinearity of the classes of Boolean functions under consideration. Let n be an odd number and f be a Boolean function in n variables, constructed according to the rule $f(x_1,\dots,x_n) = x_n\varphi_0(x_1,\dots,x_{n-1}) \oplus \oplus\, \bar{x}_n\varphi_1(x_1,\dots,x_{n-1})$, where φ_0,φ_1 are bent functions in $n-1$ variables. It was shown that for such a function $\sigma(f) = 2^{(3n-1)/2}$. We also examine a function of the form $f = f(x_1,\dots,x_n) = x_n x_{n-1}\varphi(x_1,\dots,x_{n-2})$ with an odd number of variables, where $n \geqslant 6$, φ is a bent function in $n-2$ variables. For this function $\sigma(f) = (2^n-4)2^{(n-2)/2} + 3\cdot 2^{n-1} - 2W_\varphi(0,\dots,0)$, where $W_\varphi(0,\dots,0)$ is the Walsh coefficient of the function φ. Moreover, an inequality is provided that demonstrates the relationship between the curvature and nonlinearity of arbitrary Boolean functions.

Keywords: *Boolean functions, bent functions, curvature of Boolean function, nonlinearity of Boolean function.*

Введение

Пусть n — натуральное число, f — булева функция от n переменных. Всюду далее $\Omega = \{0,1\}$. Коэффициент $W_f(\mathbf{a})$ Уолша — Адамара функции f определяется для каждого вектора $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \Omega^n$ равенством [1]

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}=(x_1,\dots,x_n)\in\Omega^n} (-1)^{f(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle},$$

где $\langle \mathbf{a}, \mathbf{x} \rangle = a_1 x_1 \oplus \ldots \oplus a_n x_n$. Введём обозначение

$$\sigma(f) = \sum_{\mathbf{a} \in \Omega^n} |W_f(\mathbf{a})|$$

и назовём величину $\sigma(f)$ кривизной булевой функции f.

Исследованию величины $\sigma(f)$ посвящена работа [2], где получены следующие оценки, справедливые для каждой булевой функции f от n переменных:

$$2^n \leqslant \sigma(f) \leqslant 2^{3n/2}.$$

Нижняя оценка обращается в равенство тогда и только тогда, когда f — аффинная функция. Верхняя оценка обращается в равенство тогда и только тогда, когда f — бент-функция. Кроме того, в [2] показано, что среднее значение параметра $\sigma(f)$ в классе всех булевых функций от n переменных и в его подклассе из всех сбалансированных булевых функций эквивалентно при $n \to \infty$ величине

$$\sqrt{2/\pi} 2^{3n/2}.$$

В этой же работе приводятся точные значения величин $\sigma(f)$ в случае, когда f — сбалансированная функция, полученная из нормальной бент-функции методом Г. Доббертина [3]. Доказывается, что для этих функций при $n \to \infty$

$$\sigma(f) \sim 2^{3n/2}.$$

В работе [4] получено равенство для кривизны мажоритарной булевой функции $f(x_1, \ldots, x_n)$ от нечётного числа переменных:

$$\sigma(f) = \sigma(n) = 2n \binom{n-1}{(n-1)/2} \sum_{i=0}^{(n-1)/2} \frac{1}{2i+1} \binom{(n-1)/2}{i},$$

а также доказано, что при $n \to \infty$ имеет место соотношение

$$\sigma(f) = \sigma(n) \sim \frac{2}{\sqrt{\pi n}} 2^{3n/2}.$$

Формула для вычисления кривизны $\sigma(f)$ мажоритарной булевой функции $f(x_1, \ldots, x_n)$ от чётного числа переменных получена в работе [5], в которой доказано, что

$$\sigma(f) = \frac{\sigma(n+1)}{2} \sim \frac{2\sqrt{2}}{\sqrt{\pi n}} 2^{3n/2}, \quad n \to \infty.$$

В [6] рассматривается подход к классификации булевых функций на основе равенства их функций автокорреляции. Доказано, что если функции f и g лежат в одном классе рассматриваемой эквивалентности, то $\sigma(f) = \sigma(g)$.

В [7] булевы функции исследуются как точки на гиперсфере в евклидовом пространстве. В ней доказываются свойства кривизны булевой функции с точки зрения геометрии и свойств евклидова пространства. Понятие «кривизна булевой функции» впервые было введено в этой работе.

В работах [8–10] величина $\sigma(f)$ используется для установления оценок частот появлений элементов на отрезках выходных последовательностей фильтрующих и комбинирующих генераторов с функцией усложнения f. Чем меньше значение $\sigma(f)$, тем более точные оценки частот удаётся получить.

В [11] понятие кривизны расширяется на векторные булевы функции и используется для исследования свойств S-боксов.

В данной работе исследуется кривизна различных классов булевых функций, построенных с помощью суперпозиции, симметрических многочленов и бент-функций. В процессе исследований получаются точные значения для коэффициентов Уолша — Адамара, поэтому наряду с кривизной, как правило, приводятся результаты о нелинейности построенных функций. Устанавливается связь кривизны и нелинейности произвольных булевых функций.

1. Кривизна некоторых классов булевых функций

Рассмотрим кривизну классов булевых функций, полученных в результате суперпозиции. Обозначим: $\mathbf{1}^n = (1, \ldots, 1) \in \Omega^n$, $\mathbf{0}^n = (0, \ldots, 0) \in \Omega^n$; $\|\mathbf{u}\|$ — вес Хэмминга вектора $\mathbf{u}$.

Утверждение 1. Пусть $h(x_1, \ldots, x_n)$ — булева функция от n переменных, такая, что $h(x_1, \ldots, x_n) = f(x_1, \ldots, x_k) \oplus g(x_{k+1}, \ldots, x_n)$ для некоторого $k \in \{1, \ldots, n\}$. Тогда

$$\sigma(h) = \sigma(f)\,\sigma(g).$$

Доказательство. Рассмотрим коэффициент Уолша — Адамара функции $h(\mathbf{x})$ на произвольном векторе $\mathbf{u} \in \Omega^n$. Введём обозначения $\mathbf{x}' = (x_1, \ldots, x_k)$, $\mathbf{x}'' = (x_{k+1}, \ldots, x_n)$, $\mathbf{u}' = (u_1, \ldots, u_k)$, $\mathbf{u}'' = (u_{k+1}, \ldots, u_n)$. Тогда $W_h(\mathbf{u}) = W_f(\mathbf{u}')W_g(\mathbf{u}'')$ и для суммы модулей всех рассматриваемых коэффициентов справедливо соотношение

$$\sigma(h) = \sum_{\mathbf{u}\in\Omega^n} |W_h(\mathbf{u})| = \sum_{\mathbf{u}'\in\Omega^k} |W_f(\mathbf{u}')| \sum_{\mathbf{u}''\in\Omega^{n-k}} |W_g(\mathbf{u}'')| = \sigma(f)\,\sigma(g).$$

Утверждение 1 доказано. ■

Утверждение 2. Пусть $h(x_1, \ldots, x_n)$ — булева функция от n переменных, определяемая равенством

$$h(\mathbf{x}) = x_1 \ldots x_{k_1} \oplus x_{k_1+1} \ldots x_{k_1+k_2} \oplus \ldots \oplus x_{k_1+\ldots+k_{t-1}+1} \ldots x_{k_1+\ldots+k_t} \tag{1}$$

для некоторых $k_1, \ldots, k_t \in \{1, \ldots, n\}$, таких, что $k_1 + \ldots + k_t = n$. Тогда

$$\sigma(h) = (3 \cdot 2^{k_1} - 4)(3 \cdot 2^{k_2} - 4) \cdot \ldots \cdot (3 \cdot 2^{k_t} - 4). \tag{2}$$

Доказательство. Найдём кривизну функции $f(x_1, \ldots, x_k) = x_1 \ldots x_k$. Для этого рассмотрим произвольный коэффициент Уолша — Адамара функции f на векторе $\mathbf{u}$:

$$W_f(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega^k} (-1)^{x_1\ldots x_k \oplus \langle \mathbf{x},\mathbf{u}\rangle} = \sum_{\mathbf{x}\in\Omega^k, \mathbf{x}\neq\mathbf{1}^k} (-1)^{\langle \mathbf{x},\mathbf{u}\rangle} - (-1)^{\|\mathbf{u}\|} = \sum_{\mathbf{x}\in\Omega^k} (-1)^{\langle \mathbf{x},\mathbf{u}\rangle} - 2\,(-1)^{\|\mathbf{u}\|}.$$

Тогда

$$W_f(\mathbf{u}) = \begin{cases} 2^k - 2, & \text{если } \mathbf{u} = \mathbf{0}^k, \\ -2\,(-1)^{\|\mathbf{u}\|}, & \text{если } \mathbf{u} \neq \mathbf{0}^k. \end{cases} \tag{3}$$

Отсюда получаем

$$\sigma(f) = \sum_{\mathbf{u}\in\Omega^k} |W_f(\mathbf{u})| = 2^k - 2 + (2^k - 1)\,2 = 3 \cdot 2^k - 4.$$

Из утверждения 1 и определения функции h следует (2). ■

Пусть $h(\mathbf{x})$ — произвольная булева функция от n переменных; $\mathrm{nl}(h)$ — её нелинейность (расстояние до класса аффинных функций). Известно, что при чётном n

$$\mathrm{nl}(h) \leqslant 2^{n-1} - 2^{n/2-1}, \tag{4}$$

а при нечётном n

$$\mathrm{nl}(h) \leqslant 2^{n-1} - 2^{(n-1)/2}. \tag{5}$$

Покажем, что данные оценки достигаются для функций из класса, определённого в утверждении 2.

Утверждение 3. Пусть $h(\mathbf{x})$ — булева функция от n переменных, определяемая равенством (1). Тогда при чётном n неравенство (4) обращается в равенство тогда и только тогда, когда $k_i = 2$ для всех $i \in \{1, \ldots, t\}$. При нечётном n оценка (5) достижима тогда и только тогда, когда $k_j = 1$ для некоторого $j \in \{1, \ldots, t\}$ и $k_i = 2$ для всех $i \neq j$.

Доказательство. Обозначим через $f_i = x_{k_1+\ldots+k_{i-1}+1} \ldots x_{k_i}$ булеву функцию от k_i переменных. Рассмотрим коэффициент Уолша — Адамара функции f_i на произвольном векторе $\mathbf{u} \in \Omega^{k_i}$. Согласно равенству (3), имеем

$$W_{f_i}(\mathbf{u}) = \begin{cases} 2^{k_i} - 2, & \text{если } \mathbf{u} = \mathbf{0}^{k_i}, \\ -2\,(-1)^{\|\mathbf{u}\|}, & \text{если } \mathbf{u} \neq \mathbf{0}^{k_i}. \end{cases}$$

Пусть n — чётное число. При $k_i = 2$ для всех $i \in \{1, \ldots, t\}$ справедливо равенство $h(\mathbf{x}) = x_1x_2 \oplus x_3x_4 \oplus \ldots \oplus x_{n-1}x_n$. Функция h является бент-функцией от n переменных [1] и $\mathrm{nl}(h) = 2^{n-1} - 2^{n/2-1}$. Покажем, что при других k_i, где $i \in \{1, \ldots, t\}$, оценка (4) недостижима. Пусть $k_i \geqslant 3$ для некоторого $i \in \{1, \ldots, t\}$, тогда $|W_{f_i}(0, \ldots, 0)| = 2^{k_i} - 2$. Пусть f' — сумма всех остальных слагаемых. Найдётся $\mathbf{u}' \in \Omega^{n-k_i}$, такой, что $|W_{f'}(\mathbf{u}')| \geqslant 2^{(n-k_i)/2}$; значит, по утверждению 1

$$\max_{\mathbf{u} \in \Omega^n} |W_h(\mathbf{u})| \geqslant (2^{k_i} - 2)2^{(n-k_i)/2} > 2^{n/2},$$

следовательно, оценка (4) не достигается. Предположим, что есть хотя бы два слагаемых степени 1. Без ограничения общности считаем, что $f_1 = x_1 \oplus x_2$, f_2 — сумма остальных слагаемых и $h = f_1 \oplus f_2$. Справедливо равенство $|W_{f_1}(1, 1)| = 4$ и найдётся $\mathbf{u}' \in \Omega^{n-2}$, такой, что $|W_{f_2}(\mathbf{u}')| \geqslant 2^{(n-2)/2}$. Тогда

$$\max_{\mathbf{u} \in \Omega^n} |W_h(\mathbf{u})| \geqslant 4 \cdot 2^{(n-2)/2} = 2^{(n+2)/2} > 2^{\frac{n}{2}},$$

а значит, вновь оценка (4) недостижима.

Пусть n — нечётное число. Покажем, что оценка достигается при $k_j = 1$ для некоторого $j \in \{1, \ldots, t\}$ и $k_i = 2$ для всех $i \neq j$. Без ограничения общности считаем, что $j = t$. Рассмотрим функцию

$$h(\mathbf{x}) = x_1x_2 \oplus x_3x_4 \oplus \ldots \oplus x_{n-2}x_{n-1} \oplus x_n.$$

Нетрудно проверить, что $\mathrm{nl}(h) = 2^{n-1} - 2^{(n-1)/2}$, а значит, неравенство (5) обращается в равенство. Покажем, что для других функций оценка недостижима. Пусть $k_i \geqslant 3$ для некоторого $i \in \{1, \ldots, t\}$. Аналогично случаю чётного n получаем

$$\max_{\mathbf{u} \in \Omega^n} |W_h(\mathbf{u})| \geqslant (2^{k_i} - 2)2^{(n-k_i)/2} > 2^{(n+1)/2}.$$

Пусть в многочлене Жегалкина функции f есть хотя бы три монома степени 1, тогда аналогично случаю чётного n проверяется, что

$$\max_{\mathbf{u}\in\Omega^n}|W_h(\mathbf{u})| \geqslant 2^3\cdot 2^{(n-3)/2} = 2^{(n+3)/2} > 2^{(n+1)/2}.$$

В обоих случаях $\max\limits_{\mathbf{u}\in\Omega^n}|W_h(\mathbf{u})| > 2^{(n+1)/2}$, следовательно, оценка (5) не достигается. ■

Покажем, что аффинные преобразования не меняют кривизну булевой функции.

Утверждение 4. Пусть $f(x_1,\ldots,x_n)$ — булева функция от n переменных; функция $g(x_1,\ldots,x_n)$ получена из f путём следующего преобразования:

$$g(\mathbf{x}) = f(\mathbf{x}A\oplus\mathbf{a})\oplus\langle\mathbf{b},\mathbf{x}\rangle\oplus c.$$

Здесь A — обратимая матрица над полем GF(2); $\mathbf{a},\mathbf{b}\in\Omega^n$; $c\in\Omega$. Тогда $\sigma(g)=\sigma(f)$.

Доказательство. Непосредственно следует из равенства [1]

$$W_g(\mathbf{u}) = (-1)^{\langle(\mathbf{b}\oplus\mathbf{u})(A^{-1})^{\mathrm{T}},\mathbf{a}\rangle\oplus c}\,W_f\big((\mathbf{b}\oplus\mathbf{u})(A^{-1})^{\mathrm{T}}\big).$$

Утверждение 4 доказано. ■

Пусть $n=2k$. Рассмотрим отображение $\Phi:\Omega^k\to\Omega^k$, обладающее следующими свойствами:

1) $\Phi(\mathbf{a})\neq\mathbf{0}^k$ для всех $\mathbf{a}\in\Omega^k$;
2) $\Phi(\mathbf{b})=\Phi(\widehat{\mathbf{b}})=\mathbf{c}$ для некоторых различных элементов $\mathbf{b},\widehat{\mathbf{b}}\in\Omega^k$;
3) отображение $\Phi':\Omega^k\backslash\{\mathbf{b},\widehat{\mathbf{b}}\}\to\Omega^k\backslash\{\mathbf{0}^k,\mathbf{c}\}$, определяемое равенством $\Phi'(\mathbf{x})=\Phi(\mathbf{x})$ для всех $\mathbf{x}\in\Omega^k\setminus\{\mathbf{b},\widehat{\mathbf{b}}\}$, инъективно.

В работе [11] исследована функция $f_\Phi(\mathbf{x},\mathbf{y})=\langle\Phi(\mathbf{x}),\mathbf{y}\rangle$, $\mathbf{x},\mathbf{y}\in\Omega^k$. Рассмотрим усложнение этой функции, прибавив к ней произвольную булеву функцию $h(\mathbf{x})$ от k переменных. В итоге получим конструкцию Елисеева — Мэйорана — МакФарланда [1]. Изучим функцию

$$f_\Phi(\mathbf{x},\mathbf{y}) = \langle\Phi(\mathbf{x}),\mathbf{y}\rangle\oplus h(\mathbf{x}),\ \ \mathbf{x},\mathbf{y}\in\Omega^k. \tag{6}$$

Утверждение 5. Пусть функция $f_\Phi(\mathbf{x},\mathbf{y})$ задана формулой (6). Тогда f_Φ — сбалансированная функция и

$$\sigma(f_\Phi) = 2^{3n/2}-2^n.$$

Доказательство. Коэффициент Уолша — Адамара функции $f_\Phi(\mathbf{x},\mathbf{y})$, соответствующий вектору $(\mathbf{v},\mathbf{w})$, где $\mathbf{v},\mathbf{w}\in\Omega^k$, равен

$$W_{f_\Phi}(\mathbf{v},\mathbf{w}) = \sum_{(\mathbf{x},\mathbf{y})\in\Omega^n}(-1)^{f_\Phi(\mathbf{x},\mathbf{y})\oplus\langle(\mathbf{x},\mathbf{y}),(\mathbf{v},\mathbf{w})\rangle} = \sum_{\mathbf{x}\in\Omega^k}(-1)^{h(\mathbf{x})\oplus\langle\mathbf{x},\mathbf{v}\rangle}\sum_{\mathbf{y}\in\Omega^k}(-1)^{\langle\mathbf{y},\Phi(\mathbf{x})\oplus\mathbf{w}\rangle}.$$

Заметим, что

$$\sum_{\mathbf{y}\in\Omega^k}(-1)^{\langle\mathbf{y},\Phi(\mathbf{x})\oplus\mathbf{w}\rangle} = \begin{cases}0, & \text{если } \Phi(\mathbf{x})\oplus\mathbf{w}\neq\mathbf{0}^k,\\ 2^k, & \text{если } \Phi(\mathbf{x})\oplus\mathbf{w}=\mathbf{0}^k.\end{cases}$$

Значит,

$$W_{f_\Phi}(\mathbf{v},\mathbf{w}) = 2^k\sum_{\mathbf{x}\in\Phi^{-1}(\mathbf{w})}(-1)^{h(\mathbf{x})\oplus\langle\mathbf{x},\mathbf{v}\rangle},$$

где $\Phi^{-1}(\mathbf{w})$ — полный прообраз вектора $\mathbf{w}$ при отображении Φ. Следовательно, получаем

$$W_{f_\Phi}(\mathbf{v},\mathbf{w}) = \begin{cases} 2^k(-1)^{\langle \mathbf{v},\Phi^{-1}(\mathbf{w})\rangle \oplus h(\Phi^{-1}(\mathbf{w}))}, & \text{если } \mathbf{w} \notin \{\mathbf{0}^k, \mathbf{c}\}, \\ 0, & \text{если } \mathbf{w} = \mathbf{0}^k, \\ 2^k((-1)^{\langle \mathbf{v},\mathbf{b}\rangle \oplus h(\mathbf{b})} + (-1)^{\langle \mathbf{v},\widehat{\mathbf{b}}\rangle \oplus h(\widehat{\mathbf{b}})}), & \text{если } \mathbf{w} = \mathbf{c}. \end{cases}$$

Так как $W_{f_\Phi}(\mathbf{0}) = 0$, то функция f_Φ сбалансированная [1].

Рассмотрим, при каких векторах $\mathbf{v}$ выражение $(-1)^{\langle \mathbf{v},\mathbf{b}\rangle \oplus h(\mathbf{b})} + (-1)^{\langle \mathbf{v},\widehat{\mathbf{b}}\rangle \oplus h(\widehat{\mathbf{b}})}$ не равно 0, то есть $\langle \mathbf{v},\mathbf{b}\rangle \oplus h(\mathbf{b}) = \langle \mathbf{v},\widehat{\mathbf{b}}\rangle \oplus h(\widehat{\mathbf{b}})$:

1) если $h(\mathbf{b}) \oplus h(\widehat{\mathbf{b}}) = 0$, то $\langle \mathbf{v}, \mathbf{b} \oplus \widehat{\mathbf{b}}\rangle = 0$ — данное уравнение относительно $\mathbf{v}$ имеет 2^{k-1} различных решений;
2) если $h(\mathbf{b}) \oplus h(\widehat{\mathbf{b}}) = 1$, то $\langle \mathbf{v}, \mathbf{b} \oplus \widehat{\mathbf{b}}\rangle = 1$ — данное уравнение относительно $\mathbf{v}$ также имеет 2^{k-1} различных решений.

Значит,

$$\sigma(f_\Phi) = \sum_{(\mathbf{v},\mathbf{w}) \in \Omega^{2k}} |W_{f_\Phi}(\mathbf{v},\mathbf{w})| = 2^k \cdot 2^k\,(2^k - 2) + 2^{k+1} \cdot 2^{k-1} = 2^{3k} - 2^{2k} = 2^{3n/2} - 2^n.$$

Утверждение 5 доказано. ■

Из доказательства утверждения 5 следует, что для булевой функции $f_\Phi(\mathbf{x},\mathbf{y})$ справедливо равенство

$$\mathrm{nl}(f_\Phi) = 2^{n-1} - 2^{n/2}.$$

Утверждение 6. Пусть $n \geqslant 3$, $f(x_1,\ldots,x_n) = \sigma_{n-1}(x_1,\ldots,x_n)$ — элементарный симметрический многочлен степени $n-1$ от n переменных, определяемый равенством $\sigma_{n-1}(x_1,\ldots,x_n) = \bigoplus\limits_{1 \leqslant i_1 < \ldots < i_{n-1} \leqslant n} x_{i_1} x_{i_2} \ldots x_{i_{n-1}}$. Тогда
при чётном n

$$\sigma(f) = 2^n - 4n + 2n\binom{n}{n/2},$$

а при нечётном n

$$\sigma(f) = 2^n - 4n - 4 + 4n\binom{n-1}{(n-1)/2}.$$

Доказательство. Рассмотрим коэффициент Уолша — Адамара $W_f(\mathbf{a})$ функции $f(\mathbf{x})$ на произвольном векторе $\mathbf{a}$. Разобьём множество Ω^n на подмножества: Ω_1 — множество векторов $\mathbf{x}$, у которых не менее двух координат нулевые; Ω_2 — множество векторов $\mathbf{x}$, у которых ровно одна нулевая координата; $\Omega_3 = \{\mathbf{1}^n\}$.

Нетрудно видеть, что

$$f(\mathbf{x}) = \begin{cases} 0, & \text{если } \mathbf{x} \in \Omega_1, \\ 1, & \text{если } \mathbf{x} \in \Omega_2, \\ n \bmod 2, & \text{если } \mathbf{x} \in \Omega_3. \end{cases}$$

Тогда

$$W_f(\mathbf{a}) = \sum_{\mathbf{x} \in \Omega_1} (-1)^{\langle \mathbf{a},\mathbf{x}\rangle} + \sum_{\mathbf{x} \in \Omega_2} (-1)^{1 \oplus \langle \mathbf{a},\mathbf{x}\rangle} + \sum_{\mathbf{x} \in \Omega_3} (-1)^{n \bmod 2 \oplus \langle \mathbf{a},\mathbf{x}\rangle}.$$

Используя равенство $\Omega_1 = (\Omega^n \setminus \Omega_2) \cup (\Omega^n \setminus \Omega_3)$, получим

$$W_f(\mathbf{a}) = \sum_{\mathbf{x} \in \Omega^n} (-1)^{\langle \mathbf{a},\mathbf{x}\rangle} - 2\sum_{\mathbf{x} \in \Omega_2} (-1)^{\langle \mathbf{a},\mathbf{x}\rangle} - \sum_{\mathbf{x} \in \Omega_3} (-1)^{\langle \mathbf{a},\mathbf{x}\rangle} + \sum_{\mathbf{x} \in \Omega_3} (-1)^{n \oplus \langle \mathbf{a},\mathbf{x}\rangle}.$$

Заметим, что

$$\sum_{\mathbf{x}\in\Omega^n}(-1)^{\langle\mathbf{a},\mathbf{x}\rangle}=\begin{cases}0, & \text{если } \mathbf{a}\neq\mathbf{0}^n,\\ 2^n, & \text{если } \mathbf{a}=\mathbf{0}^n.\end{cases}$$

Значит,

$$W_f(\mathbf{a})=2^n\delta_{\mathbf{a},\mathbf{0}^n}-2\sum_{\mathbf{x}\in\Omega_2}(-1)^{\langle\mathbf{a},\mathbf{x}\rangle}+(-1)^{n\oplus\|\mathbf{a}\|}-(-1)^{\|\mathbf{a}\|},$$

где $\delta_{\mathbf{a},\mathbf{0}^n}$ — символ Кронекера, определённый равенствами

$$\delta_{\mathbf{a},\mathbf{0}^n}=\begin{cases}0, & \text{если } \mathbf{a}\neq\mathbf{0}^n,\\ 1, & \text{если } \mathbf{a}=\mathbf{0}^n.\end{cases}$$

Верно равенство

$$\sum_{\mathbf{x}\in\Omega_2}(-1)^{\langle\mathbf{a},\mathbf{x}\rangle}=\|\mathbf{a}\|(-1)^{\|\mathbf{a}\|+1}+(n-\|\mathbf{a}\|)(-1)^{\|\mathbf{a}\|},$$

и коэффициенты Уолша — Адамара функции f принимают следующий вид:

$$W_f(\mathbf{a})=2^n\delta_{\mathbf{a},\mathbf{0}^n}+(-1)^{\|\mathbf{a}\|}(4\|\mathbf{a}\|-2n-1+(-1)^n).$$

Следовательно,

$$|W_f(\mathbf{a})|=\begin{cases}2^n-2n-1+(-1)^n, & \text{если } \mathbf{a}=\mathbf{0}^n,\\ \big|\,4\|\mathbf{a}\|-2n-1+(-1)^n\big|, & \text{если } \mathbf{a}\neq\mathbf{0}^n.\end{cases}\tag{7}$$

Тогда

$$\sigma(f)=\sum_{\mathbf{a}\in\Omega^n}|W_{\sigma_{n-1}}(\mathbf{a})|=|W_{\sigma_{n-1}}(\mathbf{0})|+\sum_{\mathbf{a}\in\Omega^n\setminus\{\mathbf{0}^n\}}|W_{\sigma_{n-1}}(\mathbf{a})|=$$
$$=2^n-2n-1+(-1)^n+\sum_{\mathbf{a}\in\Omega^n\setminus\{\mathbf{0}^n\}}\big|\,4\|\mathbf{a}\|-2n-1+(-1)^n\big|.$$

Исходя из того, что в множестве Ω^n векторов веса k ровно $\binom{n}{k}$, получаем

$$\sigma(f)=2^n-2n-1+(-1)^n+\sum_{k=1}^{n}\binom{n}{k}|4k-2n-1+(-1)^n|,$$

что при чётных n принимает вид

$$\sigma(f)=2^n-4n+2n\binom{n}{n/2},$$

а при нечётных —

$$\sigma(f)=2^n-4n-4+4n\binom{n-1}{(n-1)/2}.$$

Утверждение 6 доказано. ■

Следствие 1. Пусть $f(x_1,\ldots,x_n)=\sigma_{n-1}(x_1,\ldots,x_n)$, $n\geqslant 4$. Тогда

$$\mathrm{nl}(f)=n+(1-(-1)^n)/2.$$

Доказательство. Покажем, что $\max\limits_{\mathbf{a}\in\Omega^n}|W_f(\mathbf{a})| = |W_f(\mathbf{0}^n)|$. Исходя из равенств (7), $|W_f(\mathbf{a})| = \big|\,4\|\mathbf{a}\| - 2n - 1 + (-1)^n\big|$ при $\mathbf{a} \neq \mathbf{0}^n$. Так как $1 \leqslant \|\mathbf{a}\| \leqslant n$ и $n \geqslant 4$, то

$$0 \leqslant \big|\,4\|\mathbf{a}\| - 2n - 1 + (-1)^n\big| \leqslant 2n - 1 + (-1)^n,$$

а значит, $\max\limits_{\mathbf{a}\in\Omega^n\setminus\{\mathbf{0}^n\}}|W_f(\mathbf{a})| \leqslant 2n - 1 + (-1)^n$. Но $|W_f(\mathbf{0}^n)| = 2^n - 2n - 1 + (-1)^n$ и при $n \geqslant 2$ справедливо неравенство

$$2^n - 2n - 1 + (-1)^n \geqslant 2n - 1 + (-1)^n,$$

следовательно, $|W_f(\mathbf{0}^n)| \geqslant \max\limits_{\mathbf{a}\in\Omega^n\setminus\{\mathbf{0}^n\}}|W_f(\mathbf{a})|$. Тогда

$$\mathrm{nl}(f) = 2^{n-1} - \frac{1}{2}\big(2^n - 2n - 1 + (-1)^n\big) = n + \frac{1-(-1)^n}{2}.$$

Следствие 1 доказано. ■

2. Булевы функции, полученные из бент-функций

Утверждение 7. Пусть n — чётное число, $n \geqslant 4$, $f(x_1,\ldots,x_n)$ — булева функция от n переменных, задаваемая равенством

$$f(x_1,\ldots,x_n) = \sigma_{n-1}(x_1,\ldots,x_n) \oplus \varphi(x_1,\ldots,x_n),$$

где $\varphi(x_1,\ldots,x_n)$ — бент-функция. Тогда

$$2^{3n/2} - n2^{n+1} \leqslant \sigma(f) \leqslant 2^{3n/2}. \tag{8}$$

Доказательство. Рассмотрим коэффициент Уолша — Адамара $W_f(\mathbf{a})$ функции $f(\mathbf{x})$. Используя обозначения из доказательства утверждения 6, запишем

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}\in\Omega_1}(-1)^{\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle} + \sum_{\mathbf{x}\in\Omega_2}(-1)^{1\oplus\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle} + \sum_{\mathbf{x}\in\Omega_3}(-1)^{\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle}.$$

Следовательно,

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}\in\Omega^n}(-1)^{\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle} - 2\sum_{\mathbf{x}\in\Omega_2}(-1)^{\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle} + (-1)^{\varphi(\mathbf{1})+\|\mathbf{a}\|} - (-1)^{\varphi(\mathbf{1})+\|\mathbf{a}\|},$$

а значит,

$$W_f(\mathbf{a}) = W_\varphi(\mathbf{a}) - 2\sum_{\mathbf{x}\in\Omega_2}(-1)^{\varphi(\mathbf{x})\oplus\langle\mathbf{a},\mathbf{x}\rangle}.$$

Так как $\varphi(\mathbf{x})$ — бент-функция, то $W_\varphi(\mathbf{a}) = \pm 2^{n/2}$. Тогда

$$2^{n/2} - 2n \leqslant |W_f(\mathbf{a})| \leqslant 2^{n/2} + 2n;$$

отсюда следует (8). ■

Теорема 1. Пусть n — нечетное число, $f(x_1,\ldots,x_n)$ — булева функция от n переменных, определяемая равенством

$$f(x_1,\ldots,x_n) = x_n\varphi_0(x_1,\ldots,x_{n-1}) \oplus \overline{x}_n\varphi_1(x_1,\ldots,x_{n-1}),$$

где $\varphi_0(x_1,\ldots,x_{n-1})$, $\varphi_1(x_1,\ldots,x_{n-1})$ — бент-функции. Тогда $\sigma(f) = 2^{(3n-1)/2}$.

Доказательство. Рассмотрим коэффициент Уолша — Адамара функции $f(\mathbf{x})$, соответствующий вектору $\mathbf{a}$:

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}\in\Omega^n} (-1)^{x_n\varphi_0(x_1,\ldots,x_{n-1})\oplus\overline{x}_n\varphi_1(x_1,\ldots,x_{n-1})\oplus\langle\mathbf{x},\mathbf{a}\rangle}.$$

Введём следующие обозначения: $\mathbf{x}' = (x_1,\ldots,x_{n-1})$; $\mathbf{a}' = (a_1,\ldots,a_{n-1})$. Тогда

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}\in\Omega^n, x_n=1} (-1)^{\varphi_0(\mathbf{x}')\oplus\langle\mathbf{x}',\mathbf{a}'\rangle\oplus a_n} + \sum_{\mathbf{x}\in\Omega^n, x_n=0} (-1)^{\varphi_1(\mathbf{x}')\oplus\langle\mathbf{x}',\mathbf{a}'\rangle}.$$

Следовательно,

$$W_f(\mathbf{a}) = (-1)^{a_n} W_{\varphi_0}(\mathbf{a}') + W_{\varphi_1}(\mathbf{a}').$$

Так как $\varphi_0(\mathbf{x}'), \varphi_1(\mathbf{x}')$ — бент-функции, то $W_f(\mathbf{a}) \in \{0, \pm 2^{(n+1)/2}\}$. Введём обозначения: $\mathbf{a}_1 = (\underbrace{a_1,\ldots,a_{n-1}}_{\mathbf{a}'}, 0)$; $\mathbf{a}_2 = (\underbrace{a_1,\ldots,a_{n-1}}_{\mathbf{a}'}, 1)$. Тогда

$$W_f(\mathbf{a}_1) = W_{\varphi_0}(\mathbf{a}') + W_{\varphi_1}(\mathbf{a}'), \quad W_f(\mathbf{a}_2) = -W_{\varphi_0}(\mathbf{a}') + W_{\varphi_1}(\mathbf{a}').$$

Возможны следующие варианты:

1) $W_f(\mathbf{a}_1) = 0$, $|W_f(\mathbf{a}_2)| = 2^{(n+1)/2}$;
2) $W_f(\mathbf{a}_2) = 0$, $|W_f(\mathbf{a}_1)| = 2^{(n+1)/2}$.

Таким образом, все векторы из Ω^n разбиваются на пары и справедливо соотношение

$$\sigma(f) = \sum_{\mathbf{a}\in\Omega^n} |W_f(\mathbf{a})| = 2^{n-1}(2^{(n+1)/2}) = 2^{(3n-1)/2}.$$

Теорема 1 доказана. ■

Из доказательства следует, что f сбалансирована тогда и только тогда, когда $W_{\varphi_0}(\mathbf{0}^{n-1}) + W_{\varphi_1}(\mathbf{0}^{n-1}) = 0$, то есть, учитывая равенства $\|\varphi_0\| = 2^{n-2} - W_{\varphi_0}(\mathbf{0}^{n-1})/2$ и $\|\varphi_1\| = 2^{n-2} - W_{\varphi_1}(\mathbf{0}^{n-1})/2$, φ_0 и φ_1 — бент-функции разного веса. При этом $\mathrm{nl}(f) = 2^{n-1} - 2^{(n-1)/2}$

Теорема 2. Пусть n — чётное число, $n \geqslant 6$, $f(x_1,\ldots,x_n)$ — булева функция от n переменных, определяемая равенством

$$f(x_1,\ldots,x_n) = x_n x_{n-1}\varphi(x_1,\ldots,x_{n-2}),$$

где $\varphi(x_1,\ldots,x_{n-2})$ — бент-функция. Тогда

$$\sigma(f) = (2^n - 4)\,2^{(n-2)/2} + 3\cdot 2^{n-1} - 2\,W_\varphi(\mathbf{0}^{n-2}).$$

Доказательство. Рассмотрим коэффициент Уолша — Адамара функции $f(\mathbf{x})$, соответствующий вектору $\mathbf{a} = (a_1,\ldots,a_{n-2},a_{n-1},a_n)$. Разобьём множество Ω^n на следующие подмножества: Ω_1 — множество векторов $\mathbf{x}$, для которых $x_{n-1} = 0$, $x_n = 0$; Ω_2 — множество векторов $\mathbf{x}$, для которых $x_{n-1} = 0$, $x_n = 1$; Ω_3 — множество векторов $\mathbf{x}$, для которых $x_{n-1} = 1$, $x_n = 0$; Ω_4 — множество векторов $\mathbf{x}$, для которых $x_{n-1} = 1$, $x_n = 1$. Введём обозначения: $\mathbf{x}' = (x_1,\ldots,x_{n-2})$; $\mathbf{a}' = (a_1,\ldots,a_{n-2})$. Тогда

$$W_f(\mathbf{a}) = \sum_{\mathbf{x}\in\Omega_1} (-1)^{\langle\mathbf{x}',\mathbf{a}'\rangle} + \sum_{\mathbf{x}\in\Omega_2} (-1)^{\langle\mathbf{x}',\mathbf{a}'\rangle\oplus a_n} + \sum_{\mathbf{x}\in\Omega_3} (-1)^{\langle\mathbf{x}',\mathbf{a}'\rangle\oplus a_{n-1}} + \sum_{\mathbf{x}\in\Omega_4} (-1)^{\varphi(\mathbf{x}')\oplus\langle\mathbf{x}',\mathbf{a}'\rangle\oplus a_n\oplus a_{n-1}}.$$

Если $\mathbf{a}' \neq \mathbf{0}^{n-2}$, то $\sum\limits_{\mathbf{x}' \in \Omega^{n-2}} (-1)^{\langle \mathbf{x}', \mathbf{a}' \rangle} = 0$. Тогда $W_f(\mathbf{a}) = (-1)^{a_n \oplus a_{n-1}} W_\varphi(\mathbf{a}')$, а значит, так как $\varphi(\mathbf{x}')$ — бент-функция, верно равенство $|W_f(\mathbf{a})| = |W_\varphi(\mathbf{a}')| = 2^{(n-2)/2}$. Если $\mathbf{a}' = \mathbf{0}^{n-2}$, то $\sum\limits_{\mathbf{x}' \in \Omega^{n-2}} (-1)^{\langle \mathbf{x}', \mathbf{a}' \rangle} = 2^{n-2}$ и справедливо равенство

$$W_f(\mathbf{a}) = 2^{n-2}((-1)^{a_{n-1}} + (-1)^{a_n} + 1) + (-1)^{a_{n-1} \oplus a_n} W_\varphi(\mathbf{a}').$$

Возможны следующие варианты:

1) если $\mathbf{a} = (0, \ldots, 0, 0)$, то $W_f(\mathbf{a}) = 3 \cdot 2^{n-2} + W_\varphi(\mathbf{0}^{n-2})$;
2) если $\mathbf{a} = (0, \ldots, 0, 1)$, то $W_f(\mathbf{a}) = 2^{n-2} - W_\varphi(\mathbf{0}^{n-2})$;
3) если $\mathbf{a} = (0, \ldots, 1, 0)$, то $W_f(\mathbf{a}) = 2^{n-2} - W_\varphi(\mathbf{0}^{n-2})$;
4) если $\mathbf{a} = (0, \ldots, 1, 1)$, то $W_f(\mathbf{a}) = -2^{n-2} + W_\varphi(\mathbf{0}^{n-2})$.

Если $W_\varphi(\mathbf{0}^{n-2}) = 2^{(n-2)/2}$, то

$$\sum_{\mathbf{a} \in \Omega^n, \mathbf{a}' = \mathbf{0}^{n-2}} |W_f(\mathbf{a})| = 6 \cdot 2^{n-2} - 2 \cdot 2^{(n-2)/2} = 3 \cdot 2^{n-1} - 2\, W_\varphi(\mathbf{0}^{n-2}).$$

Если $W_\varphi(\mathbf{0}^{n-2}) = -2^{(n-2)/2}$, то

$$\sum_{\mathbf{a} \in \Omega^n, \mathbf{a}' = \mathbf{0}^{n-2}} |W_f(\mathbf{a})| = 6 \cdot 2^{n-2} + 2 \cdot 2^{(n-2)/2} = 3 \cdot 2^{n-1} - 2\, W_\varphi(\mathbf{0}^{n-2}).$$

Таким образом, получаем

$$\sigma(f) = \sum_{\mathbf{a} \in \Omega^n} |W_f(\mathbf{a})| = \sum_{\mathbf{a} \in \Omega^n, \mathbf{a}' \neq \mathbf{0}^{n-2}} |W_f(\mathbf{a})| + \sum_{\mathbf{a} \in \Omega^n, \mathbf{a}' = \mathbf{0}^{n-2}} |W_f(\mathbf{a})| =$$
$$= (2^n - 4) 2^{(n-2)/2} + 3 \cdot 2^{n-1} - 2\, W_\varphi(\mathbf{0}^{n-2}).$$

Теорема 2 доказана. ■

В работе [2] изучена функция

$$f(x_1, \ldots, x_n) = x_1 x_2 \ldots x_{n-1} \oplus \varphi(x_1, \ldots, x_{n-2}) \oplus x_n,$$

где $\varphi(x_1, \ldots, x_{n-2})$ — бент-функция от $n-2$ переменных; n — чётное число. Рассмотрим некоторое изменение данной функции и исследуем полученный класс.

Утверждение 8. Пусть n — чётное число, $n \geqslant 4$, $f(x_1, \ldots, x_n)$ — булева функция от n переменных, определяемая равенством

$$f(x_1, \ldots, x_n) = x_1 x_2 \ldots x_{n-2} \oplus \varphi(x_1, \ldots, x_{n-2}) \oplus x_{n-1} \oplus x_n,$$

где $\varphi(x_1, \ldots, x_{n-2})$ — бент-функция. Тогда $\sigma(f) = 2^{(3n-2)/2} - 2^{(n+4)/2}$.

Доказательство. Рассмотрим коэффициент Уолша — Адамара функции $f(\mathbf{x})$, соответствующий вектору $\mathbf{a}$. Заметим, что если $a_n = 0$ или $a_{n-1} = 0$, то $W_f(\mathbf{a}) = 0$, поэтому пусть $\mathbf{a} = (a_1, \ldots, a_{n-2}, 1, 1)$. Введём обозначения: $\mathbf{a}' = (a_1, \ldots, a_{n-2})$; $\mathbf{x}' = (x_1, \ldots, x_{n-2})$. Тогда

$$W_f(\mathbf{a}', 1, 1) = \sum_{\mathbf{x} \in \Omega^n} (-1)^{x_1 x_2 \ldots x_{n-2} \oplus \varphi(x_1, \ldots, x_{n-2}) \oplus \langle \mathbf{a}', \mathbf{x}' \rangle} =$$

$$= 4 \left(\sum_{\mathbf{x}' \in \Omega^{n-2} \setminus \{\mathbf{1}^{n-2}\}} (-1)^{\varphi(\mathbf{x}') \oplus \langle \mathbf{a}', \mathbf{x}' \rangle} - (-1)^{\varphi(\mathbf{1}^{n-2}) \oplus \|\mathbf{a}'\|} \right) = 4\, W_\varphi(\mathbf{a}') - 8(-1)^{\varphi(\mathbf{1}^{n-2}) \oplus \|\mathbf{a}'\|}.$$

Рассмотрим бент-функцию $\tilde{\varphi}(\mathbf{x}')$, дуальную к $\varphi(\mathbf{x}')$; для неё справедливо равенство

$$W_\varphi(\mathbf{a}') = (-1)^{\tilde{\varphi}(\mathbf{a}')} 2^{(n-2).2}.$$

Рассмотрим булеву функцию $\psi(\mathbf{x}') = \tilde{\varphi}(\mathbf{x}') \oplus \|\mathbf{x}'\| \oplus \varphi(\mathbf{1}^{n-2})$. Коэффициент Уолша — Адамара функции $\psi(\mathbf{x}')$, соответствующий вектору $\mathbf{0}^{n-2}$, равен

$$W_\psi(\mathbf{0}^{n-2}) = \sum_{\mathbf{x}' \in \Omega^{n-2}} (-1)^{\tilde{\varphi}(\mathbf{x}') \oplus \|\mathbf{x}'\| \oplus \varphi(\mathbf{1}^{n-2})} = (-1)^{\varphi(\mathbf{1}^{n-2})} W_{\tilde{\varphi}}(\mathbf{1}^{n-2}) = 2^{(n-2)/2}.$$

Тогда

$$\|\psi\| = 2^{n-3} - \frac{1}{2} W_\psi(\mathbf{0}^{n-2}) = 2^{n-3} - 2^{(n-4)/2}.$$

Далее заметим, что выполняется соотношение

$$(-1)^{\|\mathbf{a}'\| \oplus \varphi(\mathbf{1}^{n-2})} W_\varphi(\mathbf{a}') = 2^{(n-2)/2} (-1)^{\psi(\mathbf{a}')},$$

следовательно,

$$W_\varphi(\mathbf{a}') = \begin{cases} -2^{(n-2)/2} (-1)^{\|\mathbf{a}'\| \oplus \varphi(\mathbf{1}^{n-2})}, & \text{если } \mathbf{a}' \in N_\psi, \\ 2^{(n-2)/2} (-1)^{\|\mathbf{a}'\| \oplus \varphi(\mathbf{1}^{n-2})}, & \text{если } \mathbf{a}' \notin N_\psi, \end{cases}$$

где $N_\psi = \{\mathbf{a}' \in \Omega^{n-2} : \psi(\mathbf{a}') = 1\}$ — носитель функции ψ. Таким образом, получаем равенство для кривизны

$$\sigma(f) = (2^{n-3} - 2^{(n-4)/2})(4 \cdot 2^{(n-2)/2} + 8) + (2^{n-3} + 2^{(n-4)/2})(4 \cdot 2^{(n-2)/2} - 8) = 2^{(3n-2)/2} - 2^{(n+4)/2}.$$

Утверждение 8 доказано. ■

3. Связь кривизны и нелинейности булевой функции

Утверждение 9. Для произвольной булевой функции $f(x_1, \ldots, x_n)$ от n переменных справедлива оценка

$$\sigma(f) \leqslant S(f)(2^n - 2\,\mathrm{nl}(f)),$$

где $S(f) = |\{\mathbf{a} \in \Omega^n : W_f(\mathbf{a}) \neq 0\}|$ — спектральная сложность функции f.

Доказательство. Заметим, что

$$\max_{\mathbf{a} \in \Omega^n} |W_f(\mathbf{a})| = 2^n - 2\,\mathrm{nl}(f).$$

Тогда

$$\sigma(f) = \sum_{\mathbf{a} \in \Omega^n} |W_f(\mathbf{a})| \leqslant \sum_{\mathbf{a} \in \Omega^n} \max_{\mathbf{b} \in \Omega^n} |W_f(\mathbf{b})| = S(f) \max_{\mathbf{a} \in \Omega^n} |W_f(\mathbf{a})| = S(f)(2^n - 2\,\mathrm{nl}(f)).$$

Утверждение 9 доказано. ■

Оценка из утверждения 9 достигается для бент-функций, аффинных и платовидных функций [1].

Заключение

В работе получены оценки и точные значения кривизны и нелинейности различных классов булевых функций, полученных с помощью суперпозиции булевых функций, симметрических многочленов и бент-функций. Доказано неравенство, связывающее кривизну булевой функции с ее нелинейностью.

ЛИТЕРАТУРА

1. *Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2012. 584 с.
2. *Де Ла Крус Хименес Р. А., Камловский О. В.* Суммы модулей коэффициентов Уолша — Адамара булевых функций // Дискретная математика. 2015. Т. 27. Вып. 4. С. 49–66.
3. *Dobbertin H.* Construction of bent functions and balanced Boolean functions with high nonlinearity // LNCS. 1995. V. 1008. P. 61–74.
4. *Камловский О. В.* Суммы модулей коэффициентов Уолша — Адамара некоторых сбалансированных булевых функций // Математические вопросы криптографии. 2017. Т. 8. Вып. 4. С. 75–98.
5. *Тиссин А. С.* Кривизна мажоритарной булевой функции // Дискретная математика. 2021. Т. 33. Вып. 2. С. 155–165.
6. *Fedorov S. N.* On a new classification of Boolean functions // Математические вопросы криптографии. 2019. Т. 10. Вып. 2. С. 159–168.
7. *Логачев О. А., Федоров С. Н., Ященко В. В.* Булевы функции как точки на гиперсфере в евклидовом пространстве //Дискретная математика. 2018. Т. 30. Вып. 1. С. 39–55.
8. *Камловский О. В.* Количество появлений элементов в выходных последовательностях фильтрующих генераторов // Прикладная дискретная математика. 2013. № 3(21). С. 11–25.
9. *Камловский О. В.* Количество появлений векторов на циклах выходных последовательностей двоичных комбинирующих генераторов // Проблемы передачи информации. 2017. Т. 53. Вып. 1. С.92–100.
10. *Тиссин А. С.* Число появлений элементов из заданного подмножества на отрезках усложнений линейных рекуррентных последовательностей // Прикладная дискретная математика. 2023. № 60. С. 30–39.
11. *De la Cruz Jiménez R. A.* On some properties of the curvature and nondegeneracy of Boolean functions // Математические вопросы криптографии. 2022. Т. 13. Вып. 2. С. 65–98.
12. *Камловский О. В.* Спектральный метод оценки числа решений систем нелинейных уравнений с линейными рекуррентными аргументами // Дискретная математика. 2016. Т. 28. Вып. 2. С. 27–43.

REFERENCES

1. *Logachev O. A., Sal'nikov A. A., Smyshlyaev S. V., and Yashchenko V. V.* Bulevy funktsii v teorii kodirovaniya i kriptologii. [Boolean Functions in Coding Theory and Cryptology]. Moscow, MCCME Publ., 2012. 584 p. (in Russian)
2. *De La Krus Khimenes R. A. and Kamlovskii O. V.* The sum of modules of Walsh coefficients of Boolean functions. Discrete Math. Appl., 2016, vol. 26, no. 5, pp. 259–272.
3. *Dobbertin H.* Construction of bent functions and balanced Boolean functions with high nonlinearity. LNCS, 1995, vol. 1008, pp. 61–74.
4. *Kamlovskiy O. V.* Summy moduley koeffitsientov Uolsha — Adamara nekotorykh sbalansirovannykh bulevykh funktsiy [The sum of modules of Walsh coefficients for some balanced Boolean functions]. Matematicheskie Voprosy Kriptografii, 2017, vol. 8, no. 4, pp. 75–98. (in Russian)
5. *Tissin A. S.* Curvature of the Boolean majority function. Discrete Math. Appl., 2022, vol. 32, no. 5, pp. 359–367.
6. *Fedorov S. N.* On a new classification of Boolean functions. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, no. 2, pp. 159–168.

7. *Logachev O. A., Fedorov S. N., and Yashchenko V. V.* Boolean functions as points on the hypersphere in the Euclidean space. Discrete Math. Appl., 2019, vol. 29, no. 2, pp. 89–101.
8. *Kamlovskiy O. V.* Kolichestvo poyavleniy elementov v vykhodnykh posledovatel'nostyakh fil'truyushchikh generatorov [Distribution properties of sequences produced by filtering generators]. Prikladnaya Diskretnaya Matematika, 2013, no. 3(21), pp. 11–25. (in Russian)
9. *Kamlovskii O. V.* Occurrence numbers for vectors in cycles of output sequences of binary combining generators. Problems Inform. Transmission, 2017, vol. 53, no. 1, pp. 84–91.
10. *Tissin A. S.* Chislo poyavleniy elementov iz zadannogo podmnozhestva na otrezkakh uslozhneniy lineynykh rekurrentnykh posledovatel'nostey [The number of occurrences of elements from a given subset on the complication segments of linear recurrence sequences]. Prikladnaya Diskretnaya Matematika, 2023, no. 60, pp. 30–39. (in Russian)
11. *De la Cruz Jiménez R. A.* On some properties of the curvature and nondegeneracy of Boolean functions. Matematicheskiye Voprosy Kriptografii, 2022, vol. 13, no. 2, pp. 65–98.
12. *Kamlovskii O. V.* Estimating the number of solutions of systems of nonlinear equations with linear recurring arguments by the spectral method. Discrete Math. Appl., 2017, vol. 27, no. 4, pp. 199–211.

УДК 519.719.2 DOI 10.17223/20710410/68/3

СЛОЖНОСТЬ ВЫЧИСЛЕНИЯ НЕКОТОРЫХ ПОДСТАНОВОК, ИМЕЮЩИХ *TU*-ПРЕДСТАВЛЕНИЕ

Д. Б. Фомин*, Д. И. Трифонов**

**Академия криптографии Российской Федерации, г. Москва, Россия*

***Технический комитет по стандартизации «Криптографическая защита информации», г. Москва, Россия*

E-mail: dbfomin@kryptonian.ru, d.arlekino@gmail.com

Рассматривается проблема оценки вычислительной сложности определённых классов подстановок, обладающих *TU*-представлением. В качестве метрик выбраны комбинационная сложность и глубина функции, задающей данную подстановку. Для получения оценок исследуется представление элементов поля в различных базисах: полиномиальном, нормальном, смешанном, а также с использованием PRR- и RRB-представлений элементов поля. Основное внимание уделяется анализу различных представлений элементов поля и их влиянию на вычислительную сложность. Комбинационная сложность оценивается на основе количества элементарных операций, необходимых для реализации подстановки; глубина функции определяется как максимальное количество логических уровней в схеме. Изучение различных базисов позволяет выявить наиболее эффективные способы представления, способствующие минимизации вычислительной сложности. В качестве примера приводится оценка указанных характеристик для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных стандартизированных симметричных криптографических алгоритмах. Получена минимальная из известных оценка комбинационной сложности, равная 169.

Ключевые слова: *подстановка, комбинационная сложность, глубина функции, конструкция типа «бабочка», TU-представление.*

COMPUTATIONAL WORK FOR SOME *TU*-BASED PERMUTATIONS

D. B. Fomin*, D. I. Trifonov**

**The Academy of Cryptography of the Russian Federation, Moscow, Russia*

***Technical Committee "Cryptography and Security Mechanism", Moscow, Russia*

The problem of evaluating the computational complexity of certain classes of substitutions with a *TU*-representation is considered. The metrics used include combinatorial complexity and the depth of the function that defines the substitution. To obtain these evaluations, the representation of field elements in various bases is investigated, including polynomial, normal, mixed, as well as PRR and RRB representations. The primary focus is on analyzing different representations of field elements and their impact on computational complexity. The combinatorial complexity is assessed based on the number of elementary operations required to implement the substitution, while the function depth is determined by the maximum number of logical levels in the circuit. The use of different bases allows us to identify the most effective representation methods that help minimize computational complexity. As an example, we

provide an evaluation of the specified characteristics for the substitution used in Russian standardized symmetric algorithms. The lowest known estimate of combinatorial complexity has been obtained, which equals 169.

Keywords: *permutation, combinational complexity, circuit depth, butterfly, TU-decomposition.*

Введение

Сложность вычислений является одной из ключевых проблем, актуальных в современных исследованиях в области вычислительной науки. Увеличение объёма обрабатываемых данных, широкое внедрение технологий Интернета вещей (IoT) и машинной связи (M2M) подчеркивает значимость разработки высокоэффективных и безопасных систем. Оценка сложности вычислений имеет большое значение как с теоретической, так и с практической точки зрения, позволяя, с одной стороны, понять фундаментальные ограничения и возможности алгоритмов, с другой — производить оптимизацию времени вычислений и необходимых ресурсов для реальных систем.

В настоящее время разработаны подходы к созданию стойких симметричных криптографических алгоритмов, однако вопросы, касающиеся сложности их реализации, часто остаются недостаточно изученными. Одним из ключевых примитивов современных криптографических алгоритмов являются нелинейные биективные преобразования — подстановки. В условиях ограничений современных вычислительных устройств, помимо естественных требований, связанных с их криптографическими характеристиками, возникают дополнительные ограничения, относящиеся к сложности их реализации. В данной работе представлены понятия сложности вычислений, приведён обзор методов построения нелинейных биективных преобразований с низкой вычислительной сложностью, а также рассмотрены подходы к минимизации сложности вычислений подстановок специального вида.

1. Способы построения низкоресурсных нелинейных биективных преобразований с заданными криптографическими свойствами

Существуют три основных подхода к построению подстановок с заданными эксплуатационными характеристиками: полный поиск с использованием обхода графа в глубину и метода встречи посередине [1–3], эвристические методы [3–7] и использование «простых алгебраических конструкций», например мономиальных подстановок (в частности, обращения ненулевых элементов поля) [8]. Однако практически все из этих подходов позволяют оценивать только количество операций. Для того чтобы оценить реальную физическую трудоёмкость реализации, предлагается реализовывать узлы на реальных физических устройствах [9] или использовать знание о трудоёмкости реализации каждой базисной функции для эвристического поиска оптимальной реализации [3].

Фундаментальной монографией в области оценки вычислительной сложности можно считать работу Д. Э. Сэвиджа [10]. Введённые им метрики сложности позволяют оценивать эффективность реализации на различных платформах при сохранении достаточного уровня математической строгости.

При описании представления реализации произвольной функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ зачастую используются логические схемы [11]. Логическая схема, которую мы, согласно [10], также будем называть комбинационной машиной, представляет собой соединение элементов некоторого базиса Ω, каждый из которых реализует некоторую логиче-

скую (булеву) функцию указанного базиса. Логическая схема может быть представлена в виде ориентированного графа, при этом вершины, имеющие полустепень захода равную нулю, обозначают аргументы функции f, а вершины, имеющие полустепень исхода равную нулю, — значение функции f.

Определение 1. Комбинационная сложность функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ в базисе Ω, обозначаемая $C_\Omega(f)$, есть минимальное число элементов базиса Ω, достаточное для реализации функции f логической схемой.

Определение 2. Глубина функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ в базисе Ω, обозначаемая $D_\Omega(f)$, есть число логических элементов, расположенных на самом длинном ориентированном пути графа, представляющего логическую схему.

В данной работе под мерой сложности функций f понимаются комбинационная сложность и глубина функции в некотором базисе.

Замечание 1. В качестве базиса Ω будем использовать $\Omega = \{\wedge, \vee, \oplus, \neg\}$; в случае, когда это понятно из контекста, будем опускать его и говорить просто о «комбинационной сложности функции f» и «глубине функции f».

Представляет интерес задача нахождения для заданной функции f логической схемы «минимального размера», то есть задача вычисления её комбинационной сложности. Хорошо известны разработанные для этих целей метод карт Карно и его обобщение — процедура Квайна — Мак-Класки [12]. Применение этого метода позволяет минимизировать размер схем в случае реализации функций формулами, имеющими вид суммы произведений (дизъюнктивной нормальной формы). О. Б. Лупановым показано [13], что функция сложения по модулю 2 (функция чётности, равная единице, если нечётно число единиц среди её аргументов $x_1, \ldots, x_n$, где $x_i \in \{0, 1\}$, $i = 1, \ldots, n$) реализуется при указанных ограничениях схемой экспоненциального (относительно n) размера, тогда как при отсутствии ограничений возможна реализация схемами линейного размера. Аналогичные результаты справедливы для некоторых других функций.

Определение 3 [14]. Пусть $F\colon \mathbb{F}_2^{n-t} \times \mathbb{F}_2^t \to \mathbb{F}_2^{n-t} \times \mathbb{F}_2^{m-n+t}$; $m - n + t \geqslant 1$; $x_1, y_1 \in \mathbb{F}_2^{n-t}$; $x_2 \in \mathbb{F}_2^t$; $y_2 \in \mathbb{F}_2^{m-n+t}$; $T\colon \mathbb{F}_2^{n-t} \times \mathbb{F}_2^t \to \mathbb{F}_2^{n-t}$; $U\colon \mathbb{F}_2^t \times \mathbb{F}_2^{n-t} \to \mathbb{F}_2^{m-n+t}$. Если функция F имеет представление

$$F(x_1, x_2) = (y_1, y_2) = \big(T(x_1, x_2), U(x_2, T(x_1, x_2))\big), \tag{1}$$

где $T(x_1, x_2)$ является биективным отображением по x_1 при фиксированном значении $x_2 \in \mathbb{F}_2^{n-t}$, то такое представление функции F в виде (1) называется TU-представлением (рис. 1).

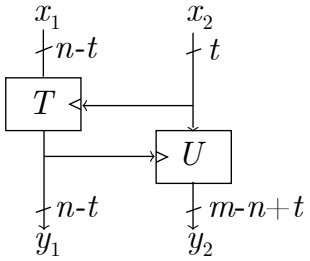


Рис. 1. Графическое изображение функции, имеющей TU-представление

2. Способы представления элементов поля

Пусть $\mathbb{F}_{2^n}$ — конечное поле из 2^n элементов. В нём есть подполе $\mathbb{F}_2$, что позволяет рассматривать поле $\mathbb{F}_{2^n}$ как векторное пространство над полем $\mathbb{F}_2$, имеющее некоторый базис $\alpha_1, \alpha_2, \ldots, \alpha_n$, состоящий из n элементов. Таким образом, для любого поля $\mathbb{F}_{2^n}$, зафиксировав базис, каждый его элемент естественным образом можно представить в виде вектора элементов поля $\mathbb{F}_2$ длины n, что позволяет задать однозначное отображение $\sigma\colon \mathbb{F}_{2^n} \to V_n$ из множества элементов поля в множество вектор-строк.

Для произвольного k, такого, что $k|n$, в поле $\mathbb{F}_{2^n}$ существует подполе из 2^k элементов, которое обозначим $\mathbb{F}_{2^k}$. При этом существует неприводимый многочлен $f(x)$ степени $m = n/k$ над $\mathbb{F}_{2^k}$, такой, что $\mathbb{F}_{2^n} \cong \mathbb{F}_{2^k}[x]/f(x)$ и $[x]_{f(x)}$ есть корень многочлена $f(x)$ в поле $\mathbb{F}_{2^k}[x]/f(x)$.

Определение 4. Пусть α — элемент поля $\mathbb{F}_{2^n}$, такой, что множество $\{\alpha^i : i = 0, \ldots, m-1\}$ является базисом $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$. Говорят, что $\{\alpha^i : i = 0, \ldots, m-1\}$ является полиномиальным базисом поля $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$; элемент α называется образующим полиномиального базиса.

Полиномиальный базис будем обозначать Poly. Помимо термина «полиномиальный базис», в литературе можно встретить эквивалентные названия: «стандартный базис» и «канонический базис». Очевидно, что $\left\{[x]^i_{f(x)} : i = 0, \ldots, m-1\right\}$ является полиномиальным базисом поля $\mathbb{F}_{2^k}[x]/f(x)$ над $\mathbb{F}_{2^k}$. Элемент α является образующим полиномиального базиса тогда и только тогда, когда α — корень многочлена $f(x)$ над $\mathbb{F}_{2^n}$. Если α — корень неприводимого над $\mathbb{F}_{2^k}$ многочлена степени m, то все корни в поле $\mathbb{F}_{2^n}$ есть элементы множества $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$, и они линейно независимы над $\mathbb{F}_{2^k}$.

Определение 5. Пусть α — элемент поля $\mathbb{F}_{2^n}$, такой, что множество $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$ является базисом $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$. Говорят, что $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$ является нормальным базисом поля $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$; элемент α называется образующим нормального базиса.

Нормальный базис будем обозначать Norm. Очевидно, что для поля $\mathbb{F}_{2^n}$ существует как минимум один нормальный базис и корень α неприводимого над $\mathbb{F}_{2^k}$ многочлена $f(x)$ в поле $\mathbb{F}_{2^n}$ является образующим как полиномиального, так и нормального базиса.

Говоря о подстановке на множестве элементов поля, будем подразумевать, что подстановка действует на векторное представление элементов поля.

Известно, что полиномиальный и нормальный базисы эффективны для реализации умножения и эндоморфизма Фробениуса соответственно [15]. При этом в настоящее время наиболее эффективный способ уменьшения комбинационной сложности и глубины функций, реализующих операции в поле, заключается в использовании теоремы о башне полей и реализации операций в подполе. Например, в работе [8] предлагается элементы поля $\mathbb{F}_{2^8}$ представлять в виде вектора $\mathbb{F}^2_{2^4}$, а элементы поля $\mathbb{F}_{2^4}$ рассматривать в виде вектора $\mathbb{F}^2_{2^2}$ и т. д. Таким образом, элементы поля $\mathbb{F}_{2^8}$ представляются векторами из множества $\left(\left(\mathbb{F}_2^2\right)^2\right)^2$.

Помимо описанных представлений элементов поля, существуют и другие способы представления, позволяющие достигать «низких» значений комбинационной сложности и глубины схем для функций, реализующих операции в поле. В работе [16] предлагается задавать элементы поля $\mathbb{F}_{2^n}$ с использованием $m \geqslant n$ бит следующим образом. Пусть $f(x)$ — неприводимый многочлен степени n над полем $\mathbb{F}_2$; многочлен $g(x)$ сте-

пени $m-n$ над полем $\mathbb{F}_2$ такой, что НОД$(f(x), g(x)) = 1$ и $g(0) \neq 0$. Тогда можно рассмотреть многочлен $p(x) = g(x)f(x)$ степени m.

Будем задавать элементы поля $\mathbb{F}_2^n$ многочленами степени не больше m таким образом, чтобы они образовывали подпространство размерности n в векторном пространстве размерности m (фактически задаётся CRC-(m,n)-код). Элементы поля $\mathbb{F}_{2^n}$ однозначно определяются элементами фактор-кольца многочленов по модулю $p(x)$, которые делятся на $g(x)$ без остатка. Заметим, что операция умножения таких многочленов корректно определена и задаётся через умножение по модулю многочлена $p(x)$. В зарубежной литературе такое представление называется Polynomial Ring Representation или PRR [16].

Согласно [17], сложность операций в поле, как правило, тем меньше, чем меньше коэффициентов в записи неприводимого многочлена, его задающего. Таким образом, в случае $p(x) = x^{n+1}+1$ сложность операций потенциально снижается. Очевидно, многочлен $x^{n+1}+1$ не является неприводимым, что не позволяет задавать с его помощью поле. Однако он может быть использован для реализации PRR-представления. В [18] предлагается рассмотреть следующий случай:

$$x^{n+1}+1 = (x+1)\left(x^n + x^{n-1} + \ldots + 1\right),$$

где $g(x) = x+1$; $f(x) = x^n + x^{n-1} + \ldots + 1$. Использование неприводимого многочлена $f(x)$ такого вида позволяет эффективно реализовывать операцию умножения в поле, а также операцию возведения в квадрат произвольное число раз, которая есть просто перестановка коэффициентов базисных векторов [16].

Заметим, что в PRR-представлении каждый элемент поля однозначно представляется одним элементом фактор-кольца $\mathbb{F}_2[x]/p(x)$. Однако с использованием всех элементов фактор-кольца $\mathbb{F}_2[x]/p(x)$ можно задать элементы кольца. В этом случае одному элементу поля можно поставить в соответствие несколько элементов фактор-кольца. Действительно, если $t_1, t_2 \in \mathbb{F}_2[x]/p(x)$ и $t_1(x) = t_2(x) \pmod{f(x)}$, то t_1 и t_2 задают один элемент поля $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/f(x)$. В случае, когда $p(x) = x^{n+1}+1 = (x+1)(x^n + x^{n-1} + \ldots + 1)$, такое представление в зарубежной литературе носит название RRB-представления [19]. Известно, что его использование уменьшает глубину операции умножения элементов поля [18].

Замечание 2. В данной работе оцениваются комбинационные сложности и глубина функций, реализующих некоторые функции, задаваемые над полем. Так как вид функции напрямую зависит от базиса и поля, над которым рассматривается преобразование, то введём дополнительные обозначения: через $C_\Omega(f; \mathbb{F}, \text{Basis})$ и $D_\Omega(f; \mathbb{F}, \text{Basis})$ будем обозначать соответственно комбинационную сложность и глубину функции f, определённой над полем $\mathbb{F}$ в базисе Basis. При использовании башни полей в качестве базиса будем указывать базис расширения.

3. Некоторые классы нелинейных биективных преобразований

В работе [20] впервые были предложены классы подстановок пространства $\mathbb{F}_2^8$, имеющих TU-представление и обладающие «высокими» показателями криптографических характеристик. В работах [20–22] данные подстановки были обобщены и предложены параметрические семейства нелинейных биективных преобразований, а также оценены их криптографические характеристики.

Определение 6. Пусть $x_1, x_2 \in \mathbb{F}_2^m$, $\pi_i, \widehat{\pi}_i \in S(\mathbb{F}_2^m)$, $\pi_i(0) = 0$, $\widehat{\pi}_i(0) = 0$, $i = 1, 2$. Тогда подстановку $F_{\text{А}}(x_1, x_2) = (y_1, y_2)$, определяемую равенствами

$$y_1 = \begin{cases} \pi_1(x_1)\, x_2, & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases}$$

$$y_2 = \begin{cases} \pi_2\left((x_2)^2\, \pi_1(x_1)\right), & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0, \end{cases}$$

будем называть подстановкой из параметрического семейства типа «А» или просто подстановкой типа «А» (рис. 2).

Определение 7. Пусть $x_1, x_2 \in \mathbb{F}_2^m$, $\pi_i, \widehat{\pi}_i \in S(\mathbb{F}_2^m)$, $\pi_i(0) = 0$, $\widehat{\pi}_i(0) = 0$, $i = 1, 2$. Тогда подстановку $F_{\text{Б}}(x_1, x_2) = (y_1, y_2)$, определяемую равенствами

$$y_1 = \begin{cases} x_1 \cdot \pi_1(x_2), & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases}$$

$$y_2 = \begin{cases} x_2 \cdot \pi_2\left(x_1 \cdot \pi_1(x_2)\right), & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0, \end{cases}$$

будем называть подстановкой из параметрического семейства типа «Б» или просто подстановкой типа «Б» (рис. 3).

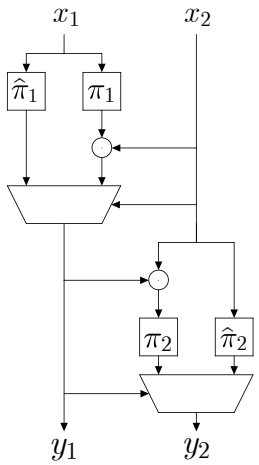


Рис. 2. Подстановка типа «А»

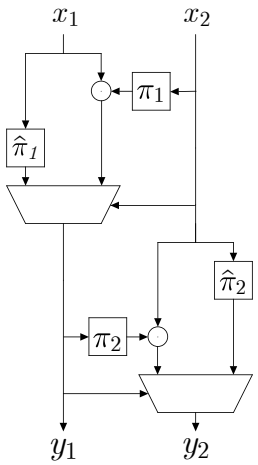


Рис. 3. Подстановка типа «Б»

Рассмотрим семейство подстановок, параметрами которого являются четвёрка степеней $(\alpha, \beta, \gamma, \delta)$ и подстановки $\widehat{\pi}_i \in S(\mathbb{F}_2^m)$, такие, что $\widehat{\pi}_i(0) = 0$, $i = 1, 2$:

$$\begin{aligned} G_1(x_1, x_2) = y_1 &= \begin{cases} x_1^{\alpha} \cdot x_2^{\beta}, & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases} \\ G_2(x_1, x_2) = y_2 &= \begin{cases} x_1^{\gamma} \cdot x_2^{\delta}, & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0. \end{cases} \end{aligned} \tag{2}$$

Чтобы (2) задавало биективное преобразование, достаточно, чтобы система уравнений

$$\begin{cases} G_1(x_1, x_2) = a_1, \\ G_2(x_1, x_2) = a_2 \end{cases}$$

имела решение для произвольных $a_1, a_2 \in \mathbb{F}_2^m$. Такое семейство подстановок будем называть подстановкой из параметрического семейства типа «Г», произвольную подстановку из которого будем обозначать F_Γ. Очевидно, что если в качестве параметров в параметрических семействах типов «А» и «Б» выбираются мономиальные подстановки, они представляются подстановками из параметрического семейства типа «Г». В [23] показано, что эта подстановка также имеет TU-представление. Однако такое задание подстановки потенциально позволяет уменьшить глубину схемы из функциональных элементов, реализующей её. Более того, среди всех известных подстановок из предложенных семейств подстановка $G_\Gamma(x_1, x_2) = (y_1, y_2)$ из параметрического семейства «Г» имеет минимальное количество нелинейных преобразований (две подстановки, два умножения и два мультиплексора) и задаётся следующим образом (здесь и далее под подстановкой обращения ненулевых элементов поля $\mathbb{F}_2^n$, $n \geqslant 2$, понимаем подстановку, задаваемую формулой x^{2^n-2}, и обозначаем её x^{-1}):

1) $x' = x_1^{-1}$;
2) $y' = x_2^{-1}$;
3) $x'' = x_1 \cdot y'$;
4) $y'' = x' \cdot y'$;
5) если $x_1 = 0$, то $y_2 = y'$, иначе $y_2 = y''$;
6) если $x_2 = 0$, то $y_1 = x'$, иначе $y_1 = x''$.

Результаты работы [24] показывают эффективность реализации определённых выше нелинейных биективных преобразований на аппаратных платформах с использованием программируемых логических интегральных схем (ПЛИС). Становится актуальной задача оценки комбинационной сложности и глубины функции для подстановок из представленных семейств, что важно при их программной реализации (bitslice implementation [1, 25, 26]) и аппаратной реализации на сверхбольших интегральных схемах (СБИС) и СБИС с программируемой архитектурой.

Для задания указанных подстановок используются следующие функции: операция умножения в поле; мультиплексор (условный выбор); подстановки.

Для параметрических семейств типов «А» и «Б» в [20, 22] в качестве подстановок π_1, π_2 рассматриваются мономиальные подстановки. В данной работе также будем рассматривать в качестве указанных параметров мономиальные подстановки.

Подстановки $\widehat{\pi}_1, \widehat{\pi}_2$ параметрических семейств «А», «Б» и «Г» в работе [27] предлагается выбирать с использованием эвристического алгоритма.

Замечание 3. Проведено экспериментальное исследование классов аффинной эквивалентности подстановок $\widehat{\pi}_1, \widehat{\pi}_2$ для рассматриваемых параметрических семейств в случае построения подстановок пространства $\mathbb{F}_2^8$ с помощью алгоритма из [27]. Это можно сделать, так как для подстановок пространства $\mathbb{F}_2^4$ имеется полная их аффинная классификация [28]. Результаты экспериментов показали, что в подавляющем большинстве случаев (более 97 %) указанные подстановки принадлежат двум семействам подстановок $\mathbb{F}_2^4$ с представителями x^{14} и $x^7 + x^4 + x$. Таким образом, представляет интерес нахождение сложности реализации этих подстановок.

Определение 8. Две подстановки $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ называются аффинно эквивалентными, если существует пара невырожденных аффинных преобразований $A_1, A_2 \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, таких, что $G(x) = A_2(F(A_1(x)))$ для всех $x \in \mathbb{F}_2^n$. Аффинное преобразование — это отображение вида $A(x) = Mx + b$, где $M \in \mathrm{GL}(n, 2)$ — невырожденная матрица; $b \in \mathbb{F}_2^n$ — вектор.

Замечание 4. Помимо рассматриваемых в данной работе подстановок, подход, изложенный далее, применим для ряда подстановок, также имеющих TU-представление, например, для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных стандартизированных симметричных алгоритмах [29] (рис. 4), а также подстановки из работы [30] (рис. 5).

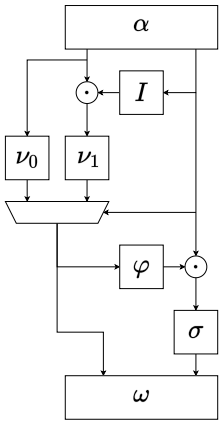


Рис. 4. Подстановка алгоритма «Кузнечик» [29]

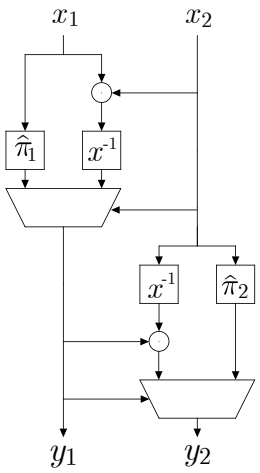


Рис. 5. Подстановка из работы [30]

4. Сложность реализации некоторых классов нелинейных биективных преобразований

4.1. Сложность задания функций над полем $\mathbb{F}_{2^2}$ в нормальном базисе

Поле $\mathbb{F}_{2^2}$ задается единственным неприводимым многочленом второй степени $x^2 + x + 1$ над полем $\mathbb{F}_2$. Для того чтобы построить поле $\mathbb{F}_{(2^2)^2}$, необходимо выбрать неприводимый многочлен степени 2 над полем $\mathbb{F}_{2^2}$.

Известно, что многочлен $x^2+x+\varepsilon$ неприводим над полем $\mathbb{F}_{2^n}$ тогда и только тогда, когда $\mathrm{tr}(\varepsilon) = 1$, где через $\mathrm{tr}_{\mathbb{F}_q}^{\mathbb{F}_{q^m}} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ (или просто tr) обозначен след из поля $\mathbb{F}_{q^m}$ в поле $\mathbb{F}_q$, ставящий в соответствие произвольному элементу $\alpha \in \mathbb{F}_{q^m}$ элемент $\mathrm{tr}_{\mathbb{F}_q}^{\mathbb{F}_{q^m}}(\alpha) = \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}$ [31]. Произвольный многочлен $f(x) = ax^2 + bx + c$, у которого $a, b \neq 0$, можно привести к этой форме, выполнив преобразование $(a/b^2)f(bx/a)$. Будем в дальнейшем рассматривать только многочлены такого вида.

Здесь и далее, если не сказано иное, элементы поля $\mathbb{F}_2$ будем обозначать курсивным шрифтом a, b, c, элементы поля $\mathbb{F}_{2^2}$ — прямым a, b, c, базисные векторы — греческими буквами α, β, γ, элементы поля $\mathbb{F}_{(2^2)^2}$ — жирным прямым шрифтом $\mathbf{a}, \mathbf{b}, \mathbf{c}$, а его базисные векторы — греческими жирными буквами $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$.

В работе [32] предлагается следующий способ реализации операций в поле $\mathbb{F}_{2^2}$. Пусть $e(x) = x^2 + x + 1$ и его корнем является элемент α. Тогда $\alpha^2 + \alpha = 1$ и $\alpha^3 = 1$, а также $\alpha^3 = \alpha^2 + \alpha$. Рассмотрим нормальный базис $\{\alpha, \alpha^2\}$. Следующие результаты напрямую следуют из работы [32].

Предложение 1. Пусть «$\cdot$» — операция умножения в $\mathbb{F}_{2^2}$, тогда для $x, y \in \mathbb{F}_{2^2}$

$$C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 7, \quad D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 3.$$

Действительно, согласно [32]: пусть $\mathrm{a} = a_0\alpha + a_1\alpha^2$, $\mathrm{b} = b_0\alpha + b_1\alpha^2$, $\mathrm{c} = c_0\alpha + c_1\alpha^2$, $\mathrm{a}\cdot\mathrm{b} = \mathrm{c}$. Тогда

$$\mathrm{a}\cdot\mathrm{b} = \left(a_0\alpha + a_1\alpha^2\right)\left(b_0\alpha + b_1\alpha^2\right) =$$
$$= \left((a_0+a_1)(b_0+b_1) + a_0b_0\right)\alpha + \left((a_0+a_1)(b_0+b_1) + a_1b_1\right)\alpha^2 = c_0\alpha + c_1\alpha^2.$$

Предложение 2. Пусть α, α^2 — элементы поля $\mathbb{F}_{2^2}$, задающие его нормальный базис, тогда для $x \in \mathbb{F}_{2^2}$ имеет место: $C_\Omega(x\cdot\alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = C_\Omega(x\cdot\alpha^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$, $D_\Omega(x\cdot\alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = D_\Omega(x\cdot\alpha^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$.

Доказательство следует из формул [32]

$$\alpha\mathrm{a} = a_1\alpha + (a_0+a_1)\alpha^2,\ \ \alpha^2\mathrm{a} = (a_0+a_1)\alpha + a_0\alpha^2.$$

Предложение 3. Для $x \in \mathbb{F}_{2^2}$ выполняется

$$C_\Omega(x^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 0, \quad D_\Omega(x^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 0.$$

Верность предложения 3 следует из двух ключевых наблюдений. Во-первых, операция возведения в квадрат является эндоморфизмом Фробениуса. Во-вторых, в $\mathbb{F}_{2^2}$ выполнено равенство $x^2 = x^{-1}$.

4.2. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ в полиномиальном базисе

В работе [33] поле $\mathbb{F}_{(2^2)^2}$ предлагается рассматривать в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$, который будем обозначать Poly. Поле $\mathbb{F}_{(2^2)^2}$ строится с использованием неприводимого многочлена $g(x) = x^2 + x + \alpha$, где α является базисным элементом поля $\mathbb{F}_{2^2}$ в нормальном базисе. Аналогично сказанному выше, неприводимость многочлена $g(x)$ следует из [31, следствие 3.79, с. 163] и того факта, что $\mathrm{tr}(\alpha) \neq 0$.

Приведём некоторые результаты, следующие из [33]. Пусть $\mathbf{a} = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}$, $\mathbf{b} = \mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}$, $\mathrm{a}_i, \mathrm{b}_i \in \mathbb{F}_{2^2}$, $i = 1, 2$.

Предложение 4. Пусть «$\cdot$» — операция умножения в поле $\mathbb{F}_{(2^2)^2}$, задаваемом неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x, y \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x\cdot y; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 30$, $D_\Omega\left(x\cdot y; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 5$.

Доказательство. Полное доказательство приведено для наглядности. Для последующих предложений справедлива такая же схема доказательства.

Рассмотрим два элемента $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{(2^2)^2}$, представимые в виде

$$\mathbf{a} = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}, \quad \mathbf{b} = \mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta},$$

где $\mathrm{a}_0, \mathrm{a}_1, \mathrm{b}_0, \mathrm{b}_1 \in \mathbb{F}_{2^2}$ и представлены в нормальном базисе; $\boldsymbol{\beta}$ — корень неприводимого многочлена $g(x)$, удовлетворяющий соотношению $\boldsymbol{\beta}^2 = \boldsymbol{\beta} + \alpha$ (так как $g(\boldsymbol{\beta}) = 0$).

Произведение $\mathbf{c} = \mathbf{a}\cdot\mathbf{b}$ раскрывается следующим образом:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})(\mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}) = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_0\mathrm{b}_1\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_0\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_1\boldsymbol{\beta}^2.$$

Подставляем $\boldsymbol{\beta}^2 = \boldsymbol{\beta} + \alpha$:

$$\mathbf{a}\cdot\mathbf{b} = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_0\mathrm{b}_1\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_0\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_1(\boldsymbol{\beta} + \alpha).$$

Группируем члены с $\boldsymbol{\beta}$ и свободные члены:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha) + (\mathrm{a}_0\mathrm{b}_1 + \mathrm{a}_1\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1)\boldsymbol{\beta}.$$

Обозначим коэффициенты при свободном члене и $\boldsymbol{\beta}$ соответственно:

$$\mathrm{c}_0 = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha, \quad \mathrm{c}_1 = \mathrm{a}_0\mathrm{b}_1 + \mathrm{a}_1\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1.$$

Коэффициент c_0 вычисляется по формуле $\mathrm{c}_0 = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha$, где

— $\mathrm{a}_0\mathrm{b}_0$ и $\mathrm{a}_1\mathrm{b}_1$ — операции умножения в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_1\mathrm{b}_1)\,\alpha$ — операция умножения на константу α в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_0\mathrm{b}_0) + (\mathrm{a}_1\mathrm{b}_1\alpha)$ — операция сложения в подполе $\mathbb{F}_{2^2}$.

Тогда, учитывая, что:

— умножение в $\mathbb{F}_{2^2}$ имеет комбинационную сложность $C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 7$ и глубину $D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 3$ (см. предложение 1);
— умножение на константу α в $\mathbb{F}_{2^2}$ имеет комбинационную сложность $C_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$ и глубину $D_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$ (см. предложение 2);
— сложение в $\mathbb{F}_{2^2}$ имеет сложность $C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 2$ и глубину $D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$,

получаем, что для c_0:

$$C_\Omega\left(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2 \cdot 7 + 1 + 2 = 17,$$

$$D_\Omega\left(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant \max\Big\{ D_\Omega\left(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}),$$

$$D_\Omega\left(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega\left(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) \Big\} \leqslant \max\{4, 5\} = 5.$$

Коэффициент c_1 после упрощений может быть вычислен по формуле

$$\mathrm{c}_1 = (\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1) + \mathrm{a}_0\mathrm{b}_0,$$

где

— $\mathrm{a}_0 + \mathrm{a}_1$, $\mathrm{b}_0 + \mathrm{b}_1$, $((\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1)) + (\mathrm{a}_0\mathrm{b}_0)$ — операции сложения в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_0 + \mathrm{a}_1)(\mathrm{b}_0 + \mathrm{b}_1)$ — операция умножения в подполе $\mathbb{F}_{2^2}$;
— $\mathrm{a}_0\mathrm{b}_0$ — операция умножения в подполе $\mathbb{F}_{2^2}$, уже выполненная при вычислении коэффициента c_0.

Таким образом, для c_1:

$$C_\Omega\big(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) \leqslant$$

$$\leqslant 2\,C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) + C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) + C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 13,$$

$$D_\Omega\big(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) \leqslant$$

$$\leqslant \max\Big\{ D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) + D_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}),$$

$$D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}), D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) \Big\} \leqslant 5.$$

Операция умножения $\mathbf{a} \cdot \mathbf{b}$ требует вычисления c_0 и c_1. Сложность и глубина оцениваются следующим образом:

$$C_\Omega\big(\mathbf{a} \cdot \mathbf{b}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) = C_\Omega(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}) + C_\Omega(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}) \leqslant 17 + 13 = 30,$$

$$D_\Omega\big(\mathbf{a} \cdot \mathbf{b}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) = \max(D_\Omega(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}), D_\Omega(\mathrm{c}_1); \mathbb{F}_{(2^2)^2}; \mathrm{Poly})) \leqslant \max(5, 5) = 5.$$

Предложение 4 доказано. ■

Заметим, что доказательство предложения 4 является чисто техническим, для него достаточно рассмотреть следующую реализацию операции умножения:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})\,(\mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}) = \\ = (\mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha) + ((\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1) + \mathrm{a}_0\mathrm{b}_0)\,\boldsymbol{\beta} = \mathrm{c}_0 + \mathrm{c}_1\boldsymbol{\beta} = \mathbf{c}.$$

Предложение 5. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^4; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$; $D_\Omega\left(x^4; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) = 1$.

Действительно, возведение в степень 4 является эндоморфизмом Фробениуса и вычисляется по формуле

$$\mathbf{a}^4 = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})^4 = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4 = \mathrm{a}_0 + \mathrm{a}_1(\boldsymbol{\beta} + \alpha^2 + \alpha) = \mathrm{a}_0 + \mathrm{a}_1(\boldsymbol{\beta} + 1) = (\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}.$$

Возведение в квадрат вычислется аналогично, откуда следует

Предложение 6. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^2; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 3$; $D_\Omega\left(x^2; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$.

Обратный элемент в поле можно вычислить с использованием алгоритма Ито — Цудзи [34], в котором используется следующее свойство: для любого ненулевого элемента $\mathbf{a} \in \mathbb{F}_{(2^2)^2}$ выполняется $\mathbf{a}^5 \in \mathbb{F}_{2^2}$. Это следует из того, что мультипликативная группа $\mathbb{F}_{2^2}^\times$ имеет порядок 3, а, поскольку $\mathbf{a}^{15} = 1$, то $(\mathbf{a}^5)^3 = 1$, что означает $\mathbf{a}^5 \in \mathbb{F}_{2^2}$. Таким образом, реализация операции обращения в $\mathbb{F}_{(2^2)^2}$ сводится к реализации операции обращения в подполе:

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left((\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})\left(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4\right) = \\ = \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}) = \mathrm{d}_0 + \mathrm{d}_1\boldsymbol{\beta}.$$

Вычисляя выражения для значений коэффициентов d_0 и d_1, получаем

Предложение 7. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8$.

Так как в рамках данной работы мы ограничиваемся выбором только мономиальных параметров в параметрических семействах типов «А» и «Б» для построения подстановок пространства $\mathbb{F}_2^8$, то найдём сложности реализации всех подстановок вида x^i, $i = 1, \ldots, 15$. Все такие подстановки разбиваются на два класса: линейные при $i \in \{1, 2, 4, 8\}$ и нелинейные при $i \in \{7, 11, 13, 14\}$.

Сначала рассмотрим подстановку x^8.

Предложение 8. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^8; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 3$; $D_\Omega\left(x^8; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$.

Доказательство следует из следующей цепочки равенств:

$$\mathbf{a}^8 = (\mathbf{a}^4)^2 = ((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta})^2 = (\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2\boldsymbol{\beta}^2 = (\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2(\boldsymbol{\beta} + \alpha) = (\mathrm{a}_0^2 + \mathrm{a}_1^2\alpha^2) + \mathrm{a}_1^2\boldsymbol{\beta}.$$

Цикломатический класс элемента α в мультипликативной группе конечного поля $\mathbb{F}_{2^n}$ образуют все элементы вида α^{2^k} для $k \geqslant 0$, соответствующие сопряжённым элементам относительно эндоморфизма Фробениуса $\phi(x) = x^2$. Для подстановок в поле $\mathbb{F}_{2^4}$ выполняются следующие соотношения внутри одного цикломатического класса:

$$x^{14} = x^{-1}, \quad x^{13} = (x^{-1})^2 = x^{-2}, \quad x^{11} = (x^{-1})^4 = x^{-4}, \quad x^7 = (x^{-1})^8 = x^{-8}.$$

Данные равенства следуют из того, что в мультипликативной группе поля $\mathbb{F}_{2^4}^{\times}$ порядка 15 выполняется $x^{15} = 1$ для всех $x \neq 0$, следовательно $x^{-k} \equiv x^{15-k}$, где $1 \leqslant k < 15$.

Предложение 9. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется

$$C_\Omega\left(x^{13}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 29, \quad D_\Omega\left(x^{13}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8;$$
$$C_\Omega\left(x^{11}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 26, \quad D_\Omega\left(x^{11}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8;$$
$$C_\Omega\left(x^{7}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 29, \quad D_\Omega\left(x^{7}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8.$$

Доказательство. Используя соотношение $x^{11} = (x^{-1})^4$ и предложения 3, 5 и 7, получаем

$$\mathbf{a}^{11} = (\mathbf{a}^{-1})^4 = \left[\left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}\right)\right]^4 =$$
$$= \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-4}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}^4\right) = \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}).$$

Действительно:

— элемент $(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha) = \mathbf{a}^5$ принадлежит подполю $\mathbb{F}_{2^2}$ (предложение 7), мультипликативная группа которого имеет порядок 3;

— $\boldsymbol{\beta}^4 = \boldsymbol{\beta} + 1$ (предложение 5).

Для x^{13} и x^7 доказательство аналогично с использованием соотношений $x^{13} = (x^{-1})^2$ и $x^7 = (x^{-1})^8$, при этом глубина вычислений не превышает установленных границ благодаря мультипликативной структуре подполя $\mathbb{F}_{2^2}$. В качестве примера рассмотрим x^{13}:

$$\mathbf{a}^{13} = \left(\mathbf{a}^{14}\right)^2 = \left[\left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\right)\boldsymbol{\beta}\right]^2 =$$
$$= (\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2)^{-1}((\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2\boldsymbol{\beta}^2) = (\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2)^{-1}((\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2(\boldsymbol{\beta} + \alpha)) =$$
$$= \left(\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2\right)^{-1}\left(\mathrm{a}_0^2 + \mathrm{a}_1^2\alpha^2 + \mathrm{a}_1^2\boldsymbol{\beta}\right).$$

Предложение 9 доказано. ■

Как указано ранее (замечание 3), нас также интересует функция $x^7 + x^4 + x$. Рассмотрим сначала функцию $x^4 + x$:

$$\mathbf{a}^4 + \mathbf{a} = (\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta} = \mathrm{a}_1.$$

Тогда очевидно

Предложение 10. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^7 + x^4 + x; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 31$; $D_\Omega\left(x^7 + x^4 + x; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 9$.

4.3. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ в нормальном и смешанном базисах

Пусть элементы поля $\mathbb{F}_{(2^2)^2}$ задаются в нормальном базисе. Известно, что с использованием нормального базиса эффективно реализуется эндоморфизм Фробениуса. Это позволяет предложить, что реализация операции обращения ненулевых элементов x^{-1} имеет меньшую сложность.

Как и в работе [8], рассмотрим нормальный базис $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$, где $\boldsymbol{\beta}$ — корень неприводимого над $\mathbb{F}_{2^2}$ многочлена $x^2+x+\alpha$. Здесь, как и выше, α — образующий нормального базиса $\mathbb{F}_{2^2}$. Пусть $\mathbf{a} = \mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4$ — произвольный ненулевой элемент поля $\mathbb{F}_{(2^2)^2}$. Обратный элемент в поле вычисляется по формуле

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left(\left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0\mathrm{a}_1 + (\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) = \mathrm{d}_0\boldsymbol{\beta} + \mathrm{d}_1\boldsymbol{\beta}^4.$$

Записав явные выражения для d_0 и d_1, получаем

Предложение 11. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в нормальном базисе $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Norm}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Norm}\right) \leqslant 7$.

Таким образом, использование нормального базиса позволяет сократить глубину функции, реализующей вычисление обратного элемента в поле, на 1.

В работе [18] предлагается рассматривать смешанные базисы для реализации операций в поле. Использование разных базисов для разных операций может привести к уменьшению глубины схемы, реализующей подстановку в целом. Например, следующая формула описывает способ реализации подстановки обращения ненулевых элементов, заданных в нормальном базисе, так, что результат представлен в полиномиальном базисе:

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left(\left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0\mathrm{a}_1 + (\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left((\mathrm{a}_1+\mathrm{a}_0) + \mathrm{a}_0\boldsymbol{\beta}\right) = \mathrm{d}_0 + \mathrm{d}_1\boldsymbol{\beta}.$$

Здесь d_0 и d_1 — коэффициенты в полниномиальном базисе. Такое представление не приводит к увеличению сложности формулы. Для смешанных базисов будем использовать обозначения PtN (Polynomialal to Normal) и NtP (Normal to Polynomial) для функций, задаваемых в одном базисе, результат которых представляется в другом базисе. В качестве полиномиального базиса везде далее будем рассматривать $\{1, \boldsymbol{\beta}\}$, а в качестве нормального — базис $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$.

Предложение 12 [18]. Для $x \in \mathbb{F}_{(2^2)^2}$ выполнено $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{NtP}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{NtP}\right) \leqslant 7$.

Найдём сложность реализации всех мономиальных подстановок. Возведение в степень 4 вычисляется по формуле

$$\mathbf{a}^4 = \left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4.$$

Если необходимо, чтобы результат был представлен в полиномиальном базисе, то получаем

$$\mathbf{a}^4 = \left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0(\boldsymbol{\beta}+1) = \mathrm{a}_0 + (\mathrm{a}_0+\mathrm{a}_1)\boldsymbol{\beta}.$$

Отсюда верно

Предложение 13. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=0,\quad D_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=0,$$
$$C_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 2,\quad D_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 1.$$

Возведение в квадрат и степень 8 вычисляются аналогично:

Предложение 14. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=C_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)\leqslant 4,$$
$$D_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=D_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)\leqslant 2.$$

Доказательство. Для элемента $\mathbf{a}=\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4\in\mathbb{F}_{(2^2)^2}$, используя равенства $\boldsymbol{\beta}^2=\boldsymbol{\beta}+\alpha$ и $\boldsymbol{\beta}^8=\boldsymbol{\beta}^4+\alpha$, получаем

$$\mathbf{a}^2=(\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4)^2=\mathrm{a}_0^2\boldsymbol{\beta}^2+\mathrm{a}_1^2\boldsymbol{\beta}^8=\mathrm{a}_0^2(\boldsymbol{\beta}+\alpha)+\mathrm{a}_1^2(\boldsymbol{\beta}^4+\alpha)=(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha)\boldsymbol{\beta}+(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha^2)\boldsymbol{\beta}^4.$$

Аналогично $\mathbf{a}^8=(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)\boldsymbol{\beta}+(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)\boldsymbol{\beta}^4$.

Для завершения доказательства заметим, что вычисление $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ можно произвести за три (а не четыре) операции, так как значение этой функции зависит от пересекающихся значений переменных. После этого значение $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$ вычисляется умножением $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ на α^2, которое возможно произвести за одну операцию.

Для получения оценки глубины необходимо рассмотреть граф, в котором $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ и $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$ — это два независимых пути. ■

Аналогично для смешанного базиса:

Предложение 15. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)=C_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 4,$$
$$D_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)=D_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 2.$$

Таким образом, использование нормального базиса для реализации линейных подстановок не повышает эффективности по сравнению с использованием полиномиального базиса.

Оценим сложность реализации подстановок x^7, x^{11}, x^{13}. Для получения итоговой формулы, как и ранее, необходимо явно получить выражения для коэффициентов d_0 и d_1. Рассмотрим сначала самый простой случай:

$$\mathbf{a}^{11}=\left(\mathbf{a}^{-1}\right)^4=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left(\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4\right)=\mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Если аргумент подстановки представляется в полиномиальном базисе, то получаем

$$\mathbf{a}^{11}=\left(\mathbf{a}^{-1}\right)^4=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left((\mathrm{a}_0+\mathrm{a}_1)+\mathrm{a}_0(\boldsymbol{\beta}+1)\right)=$$
$$=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}(\mathrm{a}_1+\mathrm{a}_0\boldsymbol{\beta})=\mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Замечание 5. Комбинационная сложность и глубина функции, реализующей подстановку x^{11} в нормальном и смешанном базисах, равны соответствующим значениям для подстановки x^{14}.

Рассмотрим подстановку x^{13} в нормальном и смешанном базисах:

$$\mathbf{a}^{13} = \left(\mathbf{a}^{-1}\right)^2 = \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\mathrm{a}_1^2\left(\alpha^2\boldsymbol{\beta}+\alpha\boldsymbol{\beta}^4\right)+\mathrm{a}_0^2\left(\alpha\boldsymbol{\beta}+\alpha^2\boldsymbol{\beta}^4\right)\right) =$$
$$= \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2\right)\boldsymbol{\beta}+\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)\boldsymbol{\beta}^4\right) = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Для подстановки $x^7 = (x^{13})^4$ формула останется такой же с точностью до перестановки значений коэффициентов при базисных векторах.

Предложение 16. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^{13};\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) = C_\Omega\left(x^7;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) \leqslant 29,$$
$$D_\Omega\left(x^{13};\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) = D_\Omega\left(x^7;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) \leqslant 7.$$

Доказательство. Достаточно показать, что значения $(\mathrm{a}_0+\mathrm{a}_1)\alpha^2$, $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$, $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ можно вычислить за шесть операций в базисе Ω. ■

Рассмотрим случай, когда значение подстановки представляется в полиномиальном базисе:

$$\mathbf{a}^{13} = \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2\right)\boldsymbol{\beta}+\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)(\boldsymbol{\beta}+1)\right) =$$
$$= \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)+\left(\mathrm{a}_0^2+\mathrm{a}_1^2\right)\boldsymbol{\beta}\right) = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Замечание 6. Комбинационная сложность и глубина функций, реализующих подстановки x^{13} и x^7 в нормальном базисе, равны соответствующим значениям в смешанном базисе.

Приведём формулу из [18], позволяющую получить значение в нормальном базисе для операции умножения в поле элементов, представленных в полиномиальном базисе:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0+\mathrm{a}_1\boldsymbol{\beta})(\mathrm{b}_0+\mathrm{b}_1\boldsymbol{\beta}) = \left[(\mathrm{a}_0+\mathrm{a}_1)(\mathrm{b}_0+\mathrm{b}_1)+\mathrm{a}_1\mathrm{b}_1\alpha\right]\boldsymbol{\beta}+(\mathrm{a}_0\mathrm{b}_0+\mathrm{a}_1\mathrm{b}_1\alpha)\boldsymbol{\beta}^4 = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Предложение 17. Для $x, y \in \mathbb{F}_{(2^2)^2}$ выполнено

$$C_\Omega\left(x\cdot y;\mathbb{F}_{(2^2)^2};\mathrm{PtN}\right) \leqslant 26; \quad D_\Omega\left(x\cdot y;\mathbb{F}_{(2^2)^2};\mathrm{PtN}\right) \leqslant 5.$$

Такое представление имеет комбинационную сложность на 4 меньше, чем в случае полиномиального базиса, однако обе реализации имеют равную глубину.

4.4. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ с использованием PRR- и RRB-представлений

Пусть $f(x) = x^4+x^3+x^2+x+1$ — неприводимый многочлен над полем $\mathbb{F}_2$ и $\boldsymbol{\beta}$ — его корень в минимальном поле разложения. Тогда $\{\boldsymbol{\beta}^0,\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,\boldsymbol{\beta}^3\}$ — полиномиальный базис. При использовании RRB-представления элементы поля представляются в виде линейной комбинации элементов множества $\{\boldsymbol{\beta}^0,\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,\boldsymbol{\beta}^3,\boldsymbol{\beta}^4\}$. Очевидно, что элементы поля представляются не единственным образом.

Для такого представления операция умножения реализуется следующим образом: пусть $\mathbf{a} = a_0+a_1\boldsymbol{\beta}+\ldots+a_4\boldsymbol{\beta}^4$, $\mathbf{b} = b_0+b_1\boldsymbol{\beta}+\ldots+b_4\boldsymbol{\beta}^4$. Тогда $\mathbf{d} = \mathbf{a}\cdot\mathbf{b}$, $\boldsymbol{d} = d_0+{}$ $+d_1\boldsymbol{\beta}+\ldots+d_4\boldsymbol{\beta}^4$ вычисляются по формулам [18]

$$d_0 = (a_1+a_3)(b_1+b_4)+(a_2+a_3)(b_2+b_3),$$
$$d_1 = (a_0+a_1)(b_0+b_1)+(a_2+a_4)(b_2+b_4),$$
$$d_2 = (a_0+a_2)(b_0+b_2)+(a_4+a_4)(b_3+b_4),$$
$$d_3 = (a_0+a_3)(b_0+b_3)+(a_2+a_2)(b_1+b_2),$$
$$d_4 = (a_0+a_4)(b_0+b_4)+(a_3+a_3)(b_1+b_3).$$

Комбинационная сложность такого представления равна 35, что больше аналогичных значений для других базисов. В то же время глубина функции, задающей такое представление, равна 3, что является наименьшим известным значением [18].

Использование PRR-представления позволяет реализовывать некоторые операции эффективнее, чем в полиномиальном, нормальном или смешанных базисах. Например, в работе [18] приводится способ вычисления обратного элемента в поле.

Пусть $\mathbf{a} = a_0 + a_1\boldsymbol{\beta} + \ldots + a_4\boldsymbol{\beta}^4$, $\mathbf{b} = b_0 + b_1\boldsymbol{\beta} + \ldots + b_4\boldsymbol{\beta}^4$, $\mathbf{a}^{-1} = \mathbf{b}$. Тогда

$$\begin{aligned} b_0 &= (a_1 \vee a_4)(a_2 \vee a_3), \\ b_1 &= ((a_4 + 1)(a_1 + a_2)) \vee (a_0 a_4 (a_2 \vee a_3)), \\ b_2 &= ((a_3 + 1)(a_2 + a_4)) \vee (a_0 a_3 (a_1 \vee a_4)), \\ b_3 &= ((a_2 + 1)(a_1 + a_3)) \vee (a_0 a_2 (a_1 \vee a_4)), \\ b_4 &= ((a_1 + 1)(a_3 + a_4)) \vee (a_0 a_1 (a_2 \vee a_3)). \end{aligned}$$

Такое представление возможно, так как в PRR-базисе аргументами булевой функции, задающей операцию обращения ненулевых элементов поля, являются только векторы, имеющие нулевой двоичный вес. Это позволяет определять остальные значения произвольным образом так, чтобы минимизировать сложность вычислений. Более того, так как значение, заданное в PRR-представлении, также является RRB-представлением, то (как и сделано авторами [18] для функции выше) можно отказаться от требования, что значение подстановки будет иметь PRR-представление. При использовании PRR-представления с $p(x) = (x + 1)(x^4 + x^3 + x^2 + x + 1)$ элемент $1 \in \mathbb{F}_{2^4}$ может быть представлен так:

— как многочлен $g(x) = x^4 + x^3 + x^2 + x + 1$ (каноническое PRR-представление);
— как константа 1 (вырожденное представление).

При последующих умножениях в поле оба представления ведут себя идентично относительно представлений элементов поля, что сохраняет корректность вычислений.

Комбинационная сложность операции обращения ненулевых элементов поля равна 31, что больше, чем в случае нормального, полиномиального и смешанного базисов, однако глубина функции равна 3.

Заметим, что возведение элемента в степени 2, 4, 8 имеет комбинационную сложность и глубину, равные нулю. Отсюда, в частности, следует, что вычисление значения мономиальной подстановки имеет сложность и глубину, равную аналогичным значениям операции вычисления подстановки x^{14}.

4.5. Сложность реализации мультиплексора MUX

Рассмотрим трудоёмкость реализации мультиплексора, аналогично [6]. Согласно определению параметрических семейств типов «А», «Б» и «Г», происходят вычисления, аналогичные следующему:

«Если $x_1 = 0$, то $y = \widehat{\pi}(x_0)$, иначе $y = \pi_2(\pi_0(x_0) \cdot \pi_1(x_1))$»,

где $\pi_0, \pi_1, \pi_2, \widehat{\pi}$ — нелинейные биективные преобразования пространства $\mathbb{F}_2^4$.

Рассмотрим функцию-индикатор, принимающую значение 1 в точке $x_1 = 0$ и нулевое значение во всех остальных точках:

$$\mathrm{Ind}_0(x_1) = \overline{x_1^{(1)}} \cdot \overline{x_1^{(2)}} \cdot \overline{x_1^{(3)}} \cdot \overline{x_1^{(4)}} = \overline{x_1^{(1)} \vee x_1^{(2)} \vee x_1^{(3)} \vee x_1^{(4)}}.$$

Здесь $x_1^{(j)}$, $j = 1, \ldots, 4$, — значение j-й координаты вектора $x_1 \in \mathbb{F}_2^4$. Комбинационная сложность вычисления функции индикатора равна 4, глубина равна 3 (за счёт вычисления операции отрицания).

Значение рассматриваемой функции может быть вычислено по формуле

$$\mathrm{Ind}_0(x_1) \cdot \widehat{\pi}(x_0) + \overline{\mathrm{Ind}_0(x_1)} \cdot \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Вычисление можно упростить следующим образом:

$$\mathrm{Ind}_0(x_1) \cdot (\widehat{\pi}(x_0) + \pi_2(\pi_0(x_0) \cdot \pi_1(0))) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Если $\pi_1(0) = 0$, последнее выражение упрощается:

$$\mathrm{Ind}_0(x_1) \cdot (\widehat{\pi}(x_0) + \pi_2(0)) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Если $\pi_2(0) = 0$, то выражение принимает следующий вид:

$$\mathrm{Ind}_0(x_1) \cdot \widehat{\pi}(x_0) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)). \tag{3}$$

Предложение 18. Комбинационная сложность вычисления значения (3) оценивается сверху величиной

$$C_\Omega\left(\widehat{\pi}\right) + C_\Omega\left(\pi_2(\pi_0(x_0) \cdot \pi_1(x_1))\right) + 12.$$

Глубина функции, реализующей (3), равна

$$\max\left\{4, D_\Omega\left(\widehat{\pi}\right) + 2, D_\Omega\left(\pi_2(\pi_0(x_0) \cdot \pi_1(x_1))\right) + 1\right\}.$$

Замечание 7. При использовании PRR-представления сложность и глубина формулы, вычисляющей $\mathrm{Ind}_0(x_1)$, не изменится, так как нулевое значение в этом представлении задается вектором из пяти нулей [18], при этом ни один из других векторов $\mathbb{F}_2^5$, задающих элементы поля $\mathbb{F}_2^4$, не имеет в своей записи четыре нуля ни на каких позициях.

В случае RRB-представления комбинационная сложность вычисления $\mathrm{Ind}_0(x_1)$ увеличится на 1, а глубина не изменится. Более того, для подстановок, сохраняющих 0 (мономиальные подстановки сохраняют 0), можно вычислять $\mathrm{Ind}_0(x_1)$ от входных значений, что даже при использовании PRR-представления не изменит комбинационную сложность (относительно значения, полученного в предложении 18) и позволит сократить глубину всей схемы целиком.

Таким образом, для нормального и полиномиального базисов комбинационная сложность вычисления y на 12 больше сложности вычисления оставшихся функций; для PRR- или RRB-представлений она равна 14.

4.6. Сложность реализации $\widehat{\pi}_i$

Как сказано ранее, в результате экспериментальных исследований алгоритма из работы [27] выяснилось, что подстановки $\widehat{\pi}_i$, $i \in \{1, 2\}$, в подавляющем большинстве аффинно эквивалентны подстановкам с представителями x^{14}, $x^7 + x^4 + x$, сложность реализации которых уже оценена.

При реализации аффинно эквивалентных подстановок помимо формул x^{14} и $x^7 + x^4 + x$ необходимо вычислить не более двух умножений на обратимые матрицы из $\mathrm{GL}(4, 2)$ и не более двух сложений с векторами длины 4.

Глубина умножения на матрицу зависит от максимального веса $\mathrm{w}_{\max}$ строки матрицы и равна $\lceil \log_2 \mathrm{w}_{\max} \rceil$. Это значение всегда меньше либо равно 2.

Комбинационную сложность можно оценить с использованием количества единиц во всей матрице. Можно легко показать, что комбинационная сложность умножения

на произвольную (но фиксированную) обратимую матрицу не превосходит 9. Действительно, при умножении вектора из $\mathbb{F}_2^4$ на произвольную обратимую (4×4)-матрицу максимальная комбинационная сложность не превышает девяти операций «$\oplus$». Это достигается за счёт оптимального выбора предвычисляемых парных сумм входных переменных.

Для четырёх переменных существует $\mathrm{C}_4^2 = 6$ возможных попарных сумм, однако в любом случае достаточно вычислить только пять из них. Выбор конкретных пар определяется структурой матрицы. Такое предвычисление требует ровно пять операций «$\oplus$».

Для строк, содержащих менее трёх единиц, дополнительных вычислений не потребуется. При вычислении выходных значений для строк матрицы, содержащих три единицы, понадобится одна дополнительная операция «$\oplus$», если использовать предвычисленные пары. Для строк с четырьмя единицами выходное значение получается суммированием двух предвычисленных пар, что добавляет ещё одну операцию.

В наиболее требовательном случае, когда матрица содержит одну строку с четырьмя единицами и три строки с тремя единицами, общее количество операций составляет

$$5 \text{ (предвычисление)} + 1 \text{ (4 единицы)} + 3 \times 1 \text{ (3 единицы)} = 9 \text{ операций «}\oplus\text{».}$$

Для конкретных матриц сложность может быть ниже, но представленный метод гарантирует, что в любом случае девяти операций «$\oplus$» достаточно для выполнения матричного умножения над $\mathbb{F}_2^4$.

Таким образом, комбинационная сложность вычислений $\widehat{\pi}_i$, $i \in \{1, 2\}$, не превосходит $C_\Omega(\pi')+26$; глубина формулы не превосходит $D_\Omega(\pi')+6$, где $\pi' \in \{x^{14}, x^7 + x^4 + x\}$.

Замечание 8. Для заданной подстановки $\widehat{\pi}_i$ могут существовать различные аффинные представления вида

$$\widehat{\pi}_i(x) = \mathcal{A}_1 \circ \pi' \circ \mathcal{A}_2(x) = \mathcal{B}_1 \circ \pi' \circ \mathcal{B}_2(x),$$

где $\pi' \in \{x^{-1}, x^7+x^4+x\}$; $\mathcal{A}_k, \mathcal{B}_k$, $k = 1, 2,$ — аффинные преобразования. Это позволяет выбирать между представлениями, оптимизированными по разным критериям: одно может минимизировать комбинационную сложность, другое — глубину схемы.

4.7. Сложность реализации подстановок из рассматриваемых параметрических семейств

Рассмотрим сложность реализации подстановок из параметрического семейства типа «Г», которые в том числе обобщают параметрические подстановки типов «А» и «Б» в случае мономиального выбора параметров π_1, π_2.

Будем считать, что входные векторы заданы в нужном базисе, так как умножение каждой из координат x_1, x_2 на обратимую матрицу не изменяет её класс эквивалентности, при этом представление операций при их задании не конкретизируется.

Проведём рассуждения аналогично работе [33]. Для этого вычисление всех подстановок и операции умножения необходимо проводить в смешанных базисах, а вычисление $\widehat{\pi}_i$, $i = 1, 2,$ — в нормальном базисе. Для подстановки из параметрического семейства типа «Г» необходимо реализовать две функции, каждая из которых состоит из трёх подстановок (две из которых мономиальные), операции умножения и мультиплексора. Комбинационная сложность задания такой подстановки оценивается следующей величиной:

$$C_\Omega(x^\alpha) + C_\Omega(x^\beta) + C_\Omega(x^\gamma) + C_\Omega(x^\delta) + 2C_\Omega(\cdot) + C_\Omega\left(\widehat{\pi}_1\right) + C_\Omega\left(\widehat{\pi}_2\right) + 2C_\Omega(\mathrm{MUX}) \leqslant$$
$$\leqslant 4 \cdot C_\Omega(x^7) + 2C_\Omega(\cdot) + 2C_\Omega(x^7 + x^4 + x) + 2 \cdot 26 + 2 \cdot 12 = 314.$$

Более того, каждая из пар подстановок (x^α, x^γ), $\left(x^\beta, x^\delta\right)$ либо содержит одну линейную подстановку и реализация пары не превышает $29+3$ операций, либо содержит две нелинейные подстановки, формулы вычисления которых во многом совпадают и их комбинационная сложность также не превосходит $29+3$ (сначала вычисляем нелинейную, затем возводим в степень 2^i при некотором i и опять получаем другую нелинейную). Этот факт позволяет уменьшить максимальное значение комбинационной сложности до 262.

Эта величина может быть сильно завышенной. Рассмотрим следующую подстановку $S(x_1, x_2) = (y_1, y_2)$ [22, 23]:

$$\begin{aligned} y_1 &= \begin{cases} x_1 \cdot x_2^2, & x_2 \neq 0, \\ x_1^{-1}, & x_2 = 0, \end{cases} \\ y_2 &= \begin{cases} x_1^{-1} \cdot x_2^{-1}, & x_1 \neq 0, \\ x_2^{-1}, & x_1 = 0. \end{cases} \end{aligned} \tag{4}$$

Её комбинационная сложность в рассматриваемом базисе не превосходит 139.

Оценим глубину формулы, задающей подстановку из параметрического семейства типа «Г»:

$$\begin{gathered} \max\left\{4, D_\Omega\left(\widehat{\pi}_1\right)+2, D_\Omega\left(\widehat{\pi}_2\right)+2, D_\Omega\left(x_1^\alpha \cdot x_2^\beta\right)+1, D_\Omega\left(x_1^\gamma \cdot x_2^\delta\right)+1\right\} \leqslant \\ \leqslant \max\left\{4, 8+8+2, \max\left\{D_\Omega\left(x_1^\alpha\right), D_\Omega\left(x_2^\beta\right), D_\Omega\left(x_1^\gamma\right), D_\Omega\left(x_2^\delta\right)\right\}+6\right\} \leqslant \\ \leqslant \max\{4, 8+8+2, 8+6\} \leqslant 18. \end{gathered}$$

Глубина формулы, задающей подстановку (4), очевидно, не превышает 14. Из этого, в частности, следует, что для реализации подстановок на программно-аппаратных платформах существенным является выбор именно $\widehat{\pi}_i$, $i = \{1, 2\}$. При той же глубине подстановка $G_\Gamma(x_1, x_2)$, определённая на с. 35, имеет комбинационную сложность 136.

В случае, когда все $i \in \{\alpha, \beta, \gamma, \delta\}$, задающие мономиальные подстановки, принадлежат множеству $\{4, 7, 11, 13, 14\}$, можно использовать представление подстановок в нормальном базисе для уменьшения глубины соответствующей цепочки на 1. При этом комбинационная сложность реализации конкретных подстановок может не измениться $(1, 7, 11, 13, 14)$, уменьшиться (4) или увеличиться $(2, 8)$. Использование смешанных базисов позволяет сократить комбинационную сложность операции умножения. Например, комбинационная сложность подстановки S (с использованием смешанных базисов) в случае задания аргументов в нормальном базисе равна 134 (две дополнительные операции на вычисление единичной подстановки в смешанном базисе). Глубина формулы, задающей подстановку, равна 13. При той же глубине подстановка $G_\Gamma(x_1, x_2)$ имеет комбинационную сложность 130.

Получается, что применение смешанных базисов потенциально позволяет снизить как комбинационную сложность, так и глубину формулы, задающей подстановку.

Использование PRR- и RRB-представлений оправдано только при необходимости минимизации глубины формулы, задающей подстановку. Однако могут они быть полезны при реализации большого количества линейных мономиальных подстановок. Дополнительно необходимо учитывать трудоёмкость преобразования из полиномиального/нормального базисов в эти представления и обратно.

5. Пример получения оценки комбинационной сложности и глубины функции для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных симметричных алгоритмах

В работе [6] получена оценка комбинационной сложности подстановки π_R, используемой в отечественном алгоритме шифрования с длиной блока 128 бит «Кузнечик» и в отечественном алгоритме хеширования «Стрибог». Как известно [29], эта подстановка имеет TU-представление и для неё применимы результаты, представленные выше. В [6] использован эвристический алгоритм поиска представления преобразований, определяемых TU-представлением, подстановки π_R. Согласно [29], происходят следующие вычисления:

1) $(l \parallel r) := \alpha(l \parallel r)$;
2) если $r = 0$, то $l := \nu_0(l)$, иначе $l := \nu_1(l \cdot I(r))$;
3) $r := \sigma(r \cdot \varphi(l))$;
4) $(l \parallel r) := \omega(l \parallel r)$.

При этом ν_0, ν_1, I, σ — нелинейные биективные функции пространства $\mathbb{F}_2^4$; $\alpha, \omega \in \mathrm{GL}(8,2)$; φ — нелинейное преобразование (подробнее см. в [29]).

В [6] для реализации умножения в поле использован полиномиальный базис $\mathbb{F}_{2^4}$, комбинационная сложность которого составляет 31. Выше показано, что в поле $\mathbb{F}_{(2^2)^2}$ комбинационная сложность умножения равна 30. Воспользуемся этим фактом. Для этого определим преобразования $si, sp \in \mathrm{GL}(4,2)$: si — подстановка, ставящая в соответствие каждому элементу $x \in \mathbb{F}_{2^4}$ его представление в $\mathbb{F}_{(2^2)^2}$; $sp = si^{-1}$ — обратное преобразование. Стоит отметить, что преобразование si (и, как следствие, sp) определены неоднозначно — всего существует 8 таких подстановок, однако следующие рассуждения верны для любой фиксации подстановки si.

Определим вспомогательное преобразование $\nu_0'(l) = \nu_0(l) + \nu_1(0)$ и рассмотрим вычисление подстановки π_R с использованием умножения в полиномиальном базисе $\mathbb{F}_{(2^2)^2}$:

1) $(l \parallel r) := \alpha(l \parallel r)$;
2) $(l \parallel r) := (si(l) \parallel si(r))$;
3) $l := \mathrm{Ind}(r = 0) \cdot \nu_0'(sp(l)) + \nu_1\left(sp\left((l \cdot si(I(sp(r))))\right)\right)$;
4) $r := \sigma\left(sp(r \cdot si(\varphi(l)))\right)$;
5) $(l \parallel r) := \omega(l \parallel r)$.

При этом умножение уже происходит в $\mathbb{F}_{(2^2)^2}$. Преобразования, задаваемые первыми двумя шагами, — некоторое новое линейное преобразование $\widehat{\alpha}$. Введём также следующие обозначения:

— $\widehat{I}(x) = si(I(sp(x)))$;
— $\widehat{\nu_0}(x) = \nu_0'(sp(x))$;
— $\widehat{\nu_1}(x) = \nu_1(sp(x))$;
— $\widehat{\varphi}(x) = si(\varphi(x))$;
— $\widehat{\sigma}(x) = \sigma(sp(x))$.

Тогда алгоритм вычисления подстановки π_R можно переписать так:

1) $(l \parallel r) := \widehat{\alpha}(l \parallel r)$;
2) $l := \mathrm{Ind}(r = 0) \cdot \widehat{\nu_0}(l) + \widehat{\nu_1}(l \cdot \widehat{I}(r))$;
3) $r := \widehat{\sigma}\left(r \cdot \widehat{\varphi}(l)\right)$;
4) $(l \parallel r) := \omega(l \parallel r)$.

Для получения оценок комбинационной сложности и глубины функции, задающей подстановку π_R, необходимо найти соответствующее величины для функций $\widehat{\alpha}$,

$\widehat{I}$, $\widehat{\nu_0}$, $\widehat{\nu_1}$, $\widehat{\varphi}$, $\widehat{\sigma}$. При этом уже корректно говорить, что все преобразования выполняются в полиномиальном базисе (как элементы представляются в памяти компьютера). С использованием эвристического алгоритма [7] построены указанные характеристики сложности (см. приложение):

$$
\begin{aligned}
C_\Omega(\widehat{\alpha};\mathbb{F}_{2^8};\mathrm{Poly}) &= 11, & D_\Omega(\widehat{\alpha};\mathbb{F}_{2^8};\mathrm{Poly}) &= 5;\\
C_\Omega(\widehat{I};\mathbb{F}_{2^4};\mathrm{Poly}) &= 16, & D_\Omega(\widehat{I};\mathbb{F}_{2^4};\mathrm{Poly}) &= 10;\\
C_\Omega(\widehat{\nu_0};\mathbb{F}_{2^4};\mathrm{Poly}) &= 19, & D_\Omega(\widehat{\nu_0};\mathbb{F}_{2^4};\mathrm{Poly}) &= 11;\\
C_\Omega(\widehat{\nu_1};\mathbb{F}_{2^4};\mathrm{Poly}) &= 11, & D_\Omega(\widehat{\nu_1};\mathbb{F}_{2^4};\mathrm{Poly}) &= 7;\\
C_\Omega(\widehat{\varphi};\mathbb{F}_{2^4};\mathrm{Poly}) &= 16, & D_\Omega(\widehat{\varphi};\mathbb{F}_{2^4};\mathrm{Poly}) &= 9;\\
C_\Omega(\widehat{\sigma};\mathbb{F}_{2^4};\mathrm{Poly}) &= 19, & D_\Omega(\widehat{\sigma};\mathbb{F}_{2^4};\mathrm{Poly}) &= 12;\\
C_\Omega(\omega;\mathbb{F}_{2^8};\mathrm{Poly}) &= 5, & D_\Omega(\omega;\mathbb{F}_{2^8};\mathrm{Poly}) &= 2.
\end{aligned}
$$

Отсюда следует, что $C_\Omega(\pi_R;\mathbb{F}_{2^8};\mathrm{Poly}) = 169$, $D_\Omega(\pi_R;\mathbb{F}_{2^8};\mathrm{Poly}) = 55$. Тем же способом можно получить оценку комбинационной сложности и глубины функции, реализующей подстановку $G_\Gamma(x_1, x_2)$, равные $2 \cdot 30 + 2 \cdot 12 + 2 \cdot 16 = 116$ и $7 + 2 + 5 = 14$ соответственно.

Аналогичные результаты, полученные с помощью эвристического алгоритма [7] (однако без перехода в $\mathbb{F}_{(2^2)^2}$), были представлены коллективом авторов на конференции CTCrypt'23 [35], где оценка комбинационной сложности составила 179. Позже авторы улучшили результат и опубликовали новую оценку, равную 176 [36].

Заметим, что полученную оценку можно улучшить следующим образом: пусть $sl, sk \in \mathrm{GL}(4,2)$. Тогда алгоритм вычисления подстановки π_R можно переписать так:

1) $(l \parallel r) := \widehat{\alpha}(l \parallel r)$;
2) $l := \mathrm{Ind}(r = 0) \cdot sl(\widehat{\nu_0})(l) + sl(\widehat{\nu_1})(l \cdot \widehat{I}(r))$;
3) $r := sk(\widehat{\sigma})\,(r \cdot sl^{-1}(\widehat{\varphi})(l))$;
4) $l := sl^{-1}(l)$
5) $r := sk^{-1}(r)$
6) $(l \parallel r) := \omega(l \parallel r)$.

Опробуя значения различных $sl, sk \in \mathrm{GL}(4,2)$, можно попытаться уменьшить оценку комбинационной сложности. Также возможно использовать умножение в смешанном базисе. Всё это является направлением дальнейших исследований.

Выводы

В работе рассмотрены оценки комбинационной сложности и глубины функций, реализующих подстановки из параметрического семейства типа «Г», которые в том числе обобщают параметрические подстановки типов «А» и «Б» в случае мономиального выбора параметров π_1, π_2.

Эти результаты могут быть использованы при реализации указанных подстановок на различных программных и аппаратных платформах. Получены также оценки комбинационной сложности и глубины функции подстановки пространства $\mathbb{F}_2^8$ из отечественных стандартизированных алгоритмах, что может быть полезно при их программной и аппаратной реализации для высокопроизводительных систем.

Авторы выражают искреннюю благодарность анонимному рецензенту за исключительно глубокий и содержательный анализ. Столь внимательное рецензирование существенно повысило научную ценность и строгость нашей работы.

ЛИТЕРАТУРА

1. *Ullrich M., De Cannière, C., Indesteege S., et al.* Finding Optimal Bitsliced Implementations of 4 x 4-bit S-boxes. Ecrypt II. 2011. http://skew2011.mat.dtu.dk/proceedings/Finding Optimal Bitsliced Implementations of 4 to 04-bit S-boxes.pdf.
2. *Чичаева А. А.* Поиск эффективно реализуемых подстановок с оптимальными криптографическими характеристиками. Рускрипто'21. Солнечногорск, 2021. https://ruscrypto.ru/resource/archive/rc2021/files/02_chichayeva.pdf.
3. *Jean J., Peyrin T., Sim S. M., and Tourteaux J.* Optimizing implementations of lightweight building blocks // IACR Trans. Symmetric Cryptol. 2017. V. 4. P. 130–168.
4. *Rudell R. L.* Multiple-Valued Logic Minimization for PLA Synthesis. Technical Report. EECS Department, University of California, 1986. https://www2.eecs.berkeley.edu/Pubs/TechRpts/1986/ERL-86-65.pdf.
5. *Hlavička J. and Fičer P.* A heuristic Boolean minimizer // Proc ICCAD'01. San Jose, California, 2001. P. 439–442.
6. *Avraamova O. D., Fomin D. B., Serov V. A., et al.* A compact bit-sliced representation of Kuznyechik S-box // Матем. вопр. криптогр. 2021. Т. 12. № 2. С. 21–38.
7. *Dansarie M.* sboxgates: A program for finding low gate count implementations of s-boxes // J. Open Source Software. 2021. No. 6(62). https://joss.theoj.org/papers/10.21105/joss.02946.
8. *Nogami Y., Nekado K., Toyota T., et al.* Mixed bases for efficient inversion in $\mathrm{GF}((2^2)^2)^2$ and conversion matrices of subbytes of AES // LNCS. 2010. V. 6225. P. 234–247.
9. *Turan M. S., McKay K., Chang D., et al.* Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. NIST, 2021. https://doi.org/10.6028/NIST.IR.8369.
10. *Сэвидж Д. Э.* Сложность вычислений. М.: Факториал, 1998. 368 с.
11. *Mano M. M. and Kime C.* Logic and Computer Design Fundamentals. Prentice Hall Press, 2007. 672 p.
12. *Boot T. L.* Digital Networks and Computer Systems. N.Y.: Wiley, 1971. 451 p.
13. *Лупанов О. Б.* О реализации функции алгебры логики формулами из конечных классов (формулами ограниченной глубины) в базисе $\&, \vee, {}^-$ // Проблемы кибернетики. 1961. Т. 6. С. 5–14.
14. *Canteaut A. and Perrin L.* On CCZ-equivalence, extended-affine equivalence, and function twisting // Finite Fields Appl. 2018. V. 56. P. 209–246.
15. *Olofsson M.* VLSI Aspects on Inversion in Finite Fields. PhD Thesis. Linköping, Sweden, 2002.
16. *Drolet G.* A new representation of elements of finite fields $\mathrm{GF}(2^m)$ yielding small complexity arithmetic circuits // IEEE Trans. Computers. 1998. V. 47. No. 9. P. 938–946.
17. *Wu H.* Low complexity bit-parallel finite field arithmetic using polynomial basis // LNCS. 1999. V. 1717. P. 280–291.
18. *Ueno R., Homma N., Nogami Y., et al.* Highly efficient $\mathrm{GF}(2^8)$ inversion circuit based on hybrid GF representations // J. Cryptogr. Engin. 2019. V. 9. No. 2. P. 101–113.
19. *Nekado K., Nogami Y., and Iokibe K.* Very short critical path implementation of AES with direct logic gates // LNCS. 2012. V. 7631. P. 51–68.
20. *Fomin D. B.* New classes of 8-bit permutations based on a butterfly structure // Матем. вопр. криптогр. 2019. Т. 10. № 2. С. 169–180.
21. *Фомин Д. Б.* Построение подстановок пространства V_{2m} с использованием $(2m, m)$-функций // Матем. вопр. криптогр. 2020. Т. 11. № 3. С. 121–138.

22. *Фомин Д. Б.* Об алгебраической степени и дифференциальной равномерности подстановок пространства V_{2m}, построенных с использованием $(2m, m)$-функций // Матем. вопр. криптогр. 2020. Т. 11. № 4. С. 133–149.
23. *Fomin D. B. and Kovrizhnykh M. A.* On differential uniformity of permutations derived using a generalized construction // Матем. вопр. криптогр. 2022. Т. 13. № 2. С. 37–52.
24. *Фомин Д. Б., Трифонов Д. И.* Об аппаратной реализации одного класса байтовых подстановок // Прикладная дискретная математика. Приложение. 2019. № 12. С. 134–137.
25. *Rebeiro C., Selvakumar A. D., and Devi A. S. L.* Bitslice implementation of AES // LNCS. 2006. V. 4301. P. 203–212.
26. *Grosso V., Leurent G., Standaert F., and Varici K.* LS-designs: Bitslice encryption for efficient masked software implementations // LNCS. 2014. V. 8540. P. 18–37.
27. *Ковржных М. А., Фомин Д. Б.* Об эвристическом алгоритме построения подстановок с заданными криптографическими характеристиками с использованием обобщённой конструкции // Прикладная дискретная математика. 2022. Т. 57. С. 5–21.
28. *Saarinen M-J. O.* Cryptographic analysis of all 4 x 4-bit S-Boxes // LNCS. 2011. V. 7118. P. 118–133.
29. *Biryukov A., Perrin L., and Udovenko A.* Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 // LNCS. 2016. V. 9665. P. 372–402.
30. *De la Cruz Jiménez R. A.* Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication // LNCS. 2017. V. 11368. P. 191–206.
31. *Lidl R. and Niederreiter H.* Finite Fields. Cambridge: Cambridge University Press, 1997. 755 p.
32. *Canright D.* A very compact s-box for AES // LNCS. 2005. V. 3659. P. 441–455.
33. *Morioka S. and Satoh A* An optimized s-box circuit architecture for low power AES design // LNCS. 2002. V. 2523. P. 172–186.
34. *Itoh T. and Tsujii S.* A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases // Information and Computation. 1988. V. 78. No. 3. P. 171–177.
35. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice Representations of Some Nonlinear Bijective Transformations. CTCrypt'23. Волгоград, 2023. `https://ctcrypt.ru/files/files/2023/08/03.pdf`.
36. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice representations of some nonlinear bijective transformations // Матем. вопр. криптогр. 2024. Т. 15. № 1. С. 97–125.

REFERENCES

1. *Ullrich M., De Cannière, C., Indesteege S., et al.* Finding Optimal Bitsliced Implementations of 4 x 4-bit S-boxes. Ecrypt II. 2011. `http://skew2011.mat.dtu.dk/proceedings/Finding Optimal Bitsliced Implementations of 4 to 04-bit S-boxes.pdf`.
2. *Chichaeva A. A.* Poisk effektivno realizuemykh podstanovok s optimal'nymi kriptograficheskimi kharakteristikami [Search for Efficiently Implementable Substitutions with Optimal Cryptographic Characteristics]. Ruskripto'21, Solnechnogorsk, 2021. `https://ruscrypto.ru/resource/archive/rc2021/files/02_chichayeva.pdf`. (in Russian)
3. *Jean J., Peyrin T., Sim S. M., and Tourteaux J.* Optimizing implementations of lightweight building blocks. IACR Trans. Symmetric Cryptol., 2017, vol. 4, pp. 130–168.
4. *Rudell R. L.* Multiple-Valued Logic Minimization for PLA Synthesis. Technical Report, EECS Department, University of California, 1986. `https://www2.eecs.berkeley.edu/Pubs/TechRpts/1986/ERL-86-65.pdf`.

5. *Hlavička J. and Fičer P.* A heuristic Boolean minimizer. Proc ICCAD'01, San Jose, California, 2001, pp. 439–442.
6. *Avraamova O. D., Fomin D. B., Serov V. A., et al.* A compact bit-sliced representation of Kuznyechik S-box. Matematicheskie Voprosy Kriptografii, 2021, vol. 12, no. 2, pp. 21–38.
7. *Dansarie M.* sboxgates: A program for finding low gate count implementations of s-boxes. J. Open Source Software, 2021, no. 6(62). `https://joss.theoj.org/papers/10.21105/joss.02946`.
8. *Nogami Y., Nekado K., Toyota T., et al.* Mixed bases for efficient inversion in $\mathrm{GF}((2^2)^2)^2$ and conversion matrices of subbytes of AES. LNCS, 2010, vol. 6225, pp. 234–247.
9. *Turan M. S., McKay K., Chang D., et al.* Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. NIST, 2021. `https://doi.org/10.6028/NIST.IR.8369`.
10. *Savage J. E.* The Complexity of Computing. N.Y., Wiley, 1976. 424 p.
11. *Mano M. M. and Kime C.* Logic and Computer Design Fundamentals. Prentice Hall Press, 2007. 672 p.
12. *Boot T. L.* Digital Networks and Computer Systems. N.Y., Wiley, 1971. 451 p.
13. *Lupanov O. B.* O realizatsii funktsii algebry logiki formulami iz konechnykh klassov (formulami ogranichennoy glubiny) v bazise $\&, \vee, {}^-$ [On implementation of a function of logic algebra by formulas from finite classes (formulas of bounded depth) in a basis $\&, \vee, {}^-$]. Problemy Kibernetiki, 1961, vol. 6, pp. 5–14. (in Russian)
14. *Canteaut A. and Perrin L.* On CCZ-equivalence, extended-affine equivalence, and function twisting. Finite Fields Appl., 2018, vol. 56, pp. 209–246.
15. *Olofsson M.* VLSI Aspects on Inversion in Finite Fields. PhD Thesis, Linköping, Sweden, 2002.
16. *Drolet G.* A new representation of elements of finite fields $\mathrm{GF}(2^m)$ yielding small complexity arithmetic circuits. IEEE Trans. Computers, 1998, vol. 47, no. 9, pp. 938–946.
17. *Wu H.* Low complexity bit-parallel finite field arithmetic using polynomial basis. LNCS, 1999, vol. 1717, pp. 280–291.
18. *Ueno R., Homma N., Nogami Y., et al.* Highly efficient $\mathrm{GF}(2^8)$ inversion circuit based on hybrid GF representations. J. Cryptogr. Engin., 2019, vol. 9, no. 2, pp. 101–113.
19. *Nekado K., Nogami Y., and Iokibe K.* Very short critical path implementation of AES with direct logic gates. LNCS, 2012, vol. 7631, pp. 51–68.
20. *Fomin D. B.* New classes of 8-bit permutations based on a butterfly structure. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, no. 2, pp. 169–180.
21. *Fomin D. B.* Postroenie podstanovok prostranstva V_{2m} s ispol'zovaniem $(2m, m)$-funktsiy [Construction of permutations on the space V_{2m} by means of $(2m, m)$-functions]. Matematicheskie Voprosy Kriptografii, 2020, vol. 11, no. 3, pp. 121–138. (in Russian)
22. *Fomin D. B.* Ob algebraicheskoy stepeni i differentsial'noy ravnomernosti podstanovok prostranstva V_{2m}, postroennykh s ispol'zovaniem $(2m, m)$-funktsiy [On the algebraic degree and differential uniformity of permutations on the space V_{2m} constructed via $(2m, m)$-functions]. Matematicheskie Voprosy Kriptografii, 2020, vol. 11, no. 4, pp. 133–149. (in Russian)
23. *Fomin D. B. and Kovrizhnyh M. A.* On differential uniformity of permutations derived using a generalized construction. Matematicheskie Voprosy Kriptografii, 2022, vol. 13, no. 2, pp. 37–52.
24. *Fomin D. B. and Trifonov D. I.* Ob apparatnoy realizatsii odnogo klassa baytovykh podstanovok [Hardware implementation of one class of 8-bit permutations]. Prikladnaya diskretnaya matematika. Prilozhenie, 2019. vol. 12, pp. 134–137. (in Russian)

25. *Rebeiro C., Selvakumar A. D., and Devi A. S. L.* Bitslice implementation of AES. LNCS, 2006, vol. 4301, pp. 203–212.
26. *Grosso V., Leurent G., Standaert F., and Varici K.* LS-designs: Bitslice encryption for efficient masked software implementations. LNCS, 2014, vol. 8540, pp. 18–37.
27. *Kovrizhnykh M. A. and Fomin D. B.* Ob evristicheskom algoritme postroeniya podstanovok s zadannymi kriptograficheskimi kharakteristikami s ispol'zovaniem obobshchennoy konstruktsii [Heuristic algorithm for obtaining permutations with given cryptographic properties using a generalized construction]. Prikladnaya Diskretnaya Matematika, 2022, vol. 57, pp. 5–21. (in Russian)
28. *Saarinen M-J. O.* Cryptographic analysis of all 4 x 4-bit S-Boxes. LNCS, 2011, vol. 7118, pp. 118–133.
29. *Biryukov A., Perrin L., and Udovenko A.* Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1. LNCS, 2016, vol. 9665, pp. 372–402.
30. *De la Cruz Jiménez R. A.* Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication. LNCS, 2017, vol. 11368, pp. 191–206.
31. *Lidl R. and Niederreiter H.* Finite Fields. Cambridge, Cambridge University Press, 1997. 755 p.
32. *Canright D.* A very compact s-box for AES. LNCS, 2005, vol. 3659, pp. 441–455.
33. *Morioka S. and Satoh A* An optimized s-box circuit architecture for low power AES design. LNCS, 2002, vol. 2523, pp. 172–186.
34. *Itoh T. and Tsujii S.* A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. Information and Computation, 1988, vol. 78, no. 3, pp. 171–177.
35. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice Representations of Some Nonlinear Bijective Transformations. CTCrypt'23, Volgograd, 2023. `https://ctcrypt.ru/files/files/2023/08/03.pdf`.
36. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the bit-slice representations of some nonlinear bijective transformations. Matematicheskie Voprosy Kriptografii, 2024, vol. 15, no. 1, pp. 97–125.

Приложение. Реализация подстановки π_R

```
def bitAlpha(inp):
    out2 = inp[4] ^ inp[7]
    out3 = inp[6] ^ inp[2]
    out0 = out3 ^ inp[5]
    out7 = inp[6] ^ inp[0]
    tmp1   = inp[3] ^ out7
    out6 = tmp1 ^ inp[1]
    out1 = inp[7] ^ inp[5] ^ inp[3] ^ inp[1]
    out4 = out1 ^ inp[2]
    out5 = tmp1 ^ out4
    return [out0, out1, out2, out3, out4, out5, out6, out7]

def bitOmega(inp):
    out1 = inp[1] ^ inp[7]
    out2 = inp[2] ^ inp[6]
    out7 = inp[3] ^ inp[1]
    out4 = inp[4] ^ inp[7] ^ out7
    out0 = inp[0]
    out3 = inp[3]
    out5 = inp[5]
    out6 = inp[2]
```

```
    return [out0, out1, out2, out3, out4, out5, out6, out7]

def bitInv(inp):
    tmp1   = inp[1] & inp[2]
    tmp2   = (tmp1 ^ inp[0]) & inp[3]
    out2 = inp[2] ^ tmp2
    tmp3   = tmp1 | inp[3]
    out3 = tmp3 ^ tmp2 ^ (tmp2 | inp[2])
    out1 = ((inp[1] & inp[3]) ^ out3) ^ (tmp1 | inp[0])
    out0 = inp[1] ^ out1 ^ tmp3 ^ (out1 & inp[0])
    return [out0, out1, out2, out3]

def bitNu0(inp):
    tmp1   = inp[1] & inp[2]
    tmp2   = tmp1 ^ inp[0]
    tmp3   = inp[1] ^ tmp2
    out1 = tmp2 ^ ((tmp3 | inp[2]) & inp[3])
    tmp4   = (tmp1 | tmp2) ^ 1
    tmp5   = inp[1] & tmp3
    out0 = tmp4 ^ ((tmp5 ^ inp[2]) | inp[3])
    tmp6   = out0 & out1
    out3 = tmp5 ^ ((tmp6 ^ inp[2]) & inp[3])
    out2 = ((tmp6 | tmp4) ^ tmp3) ^ inp[3]
    return [out0, out1, out2, out3]

def bitNu1(inp):
    out3 = (inp[3] | inp[1]) ^ inp[2] ^ inp[0]
    tmp1   = inp[1] ^ 1
    out0 = tmp1 ^ (out3 | inp[3])
    tmp2   = out3 ^ inp[1]
    out1 = (tmp2 | out0) ^ inp[2]
    out2 = (tmp2 | tmp1) ^ inp[3]
    return [out0, out1, out2, out3]

def bitPhi(inp):
    tmp1   = inp[1] ^ inp[3]
    tmp2   = inp[1] | inp[0]
    tmp3   = (tmp2 & tmp1) | inp[2]
    out2 = tmp1 ^ inp[0] ^ tmp3
    tmp4   = inp[2] ^ tmp2
    tmp5   = (tmp4 & out2) | inp[3]
    out3 = inp[0] ^ tmp5
    tmp6   = (inp[1] ^ tmp5) & tmp3
    out1 = tmp6 ^ (tmp4 & inp[1])
    out0 = (tmp4 ^ 1) | tmp6
    return [out0, out1, out2, out3]

def bitSigma(inp):
    tmp1   = inp[3] ^ (inp[2] | inp[1])
    out1 = tmp1 ^ (((inp[2] & tmp1) ^ inp[1]) | inp[0])
    tmp2   = inp[0] ^ tmp1
    tmp3   = tmp2 ^ inp[1]
    tmp4   = tmp1 ^ tmp3
    tmp5   = tmp4 & tmp2
    tmp6   = tmp5 | inp[2]
    out0 = tmp3 ^ tmp6
    out2 = tmp4 ^ ((tmp5 ^ 1 ^ inp[2]) | inp[0])
    out3 = (tmp2 | out2) ^ tmp6 ^ inp[3]
    return [out0, out1, out2, out3]
```

```
def mulf_2_2(a,b):
    tmp1 = a[0] ^ a[1]
    tmp2 = b[0] ^ b[1]
    tmp3 = a[0] & b[0]
    tmp4 = tmp1 & tmp2
    tmp5 = b[1] & a[1]
    out0 = tmp3 ^ tmp4
    out1 = tmp5 ^ tmp4
    return [out0, out1]

def bitAlpha4(a):
    out0 = a[0] ^ a[1]
    out1 = a[0]
    return [out0, out1]

def mul2_2_2(a, b):
    tmp10 = a[0] ^ a[2]
    tmp11 = a[1] ^ a[3]
    tmp20 = b[0] ^ b[2]
    tmp21 = b[1] ^ b[3]
    tmp3    = mulf_2_2([a[0], a[1]], [b[0], b[1]])
    tmp4    = mulf_2_2([tmp10, tmp11], [tmp20,tmp21])
    tmp5    = mulf_2_2([a[2], a[3]], [b[2], b[3]])
    tmp6    = bitAlpha4(tmp5)
    out0 = tmp3[0] ^ tmp6[0]
    out1 = tmp3[1] ^ tmp6[1]
    out2 = tmp3[0] ^ tmp4[0]
    out3 = tmp3[1] ^ tmp4[1]
    return [out0, out1, out2, out3]

def bitInd(inp):
    return (inp[0] | inp[1] | inp[2] | inp[3]) ^ 1

def pi_r(x):
    tmp1 = bitAlpha(x)
    r = [tmp1[0], tmp1[1], tmp1[2], tmp1[3]]
    l = [tmp1[4], tmp1[5], tmp1[6], tmp1[7]]
    tmp2 = bitInd(r)
    tmp3 = bitNu0(l)
    tmp4 = bitNu1(mul2_2_2(l, bitInv(r)))
    l[0] = (tmp2 & tmp3[0]) ^ tmp4[0]
    l[1] = (tmp2 & tmp3[1]) ^ tmp4[1]
    l[2] = (tmp2 & tmp3[2]) ^ tmp4[2]
    l[3] = (tmp2 & tmp3[3]) ^ tmp4[3]
    r = bitSigma(mul2_2_2(r,bitPhi(l)))
    out = bitOmega(r+l)
    return out

pi = []
for x in range(256):
    inp = [int(t) for t in bin(x)[2:].zfill(8)][::-1]
    tmp = pi_r(inp)
    res = 0
    for i in range(len(tmp)):
        res = res + (tmp[i] << i)
    pi.append(res)
print(pi)
```

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

УДК 519.725 DOI 10.17223/20710410/68/4

НЕАСИМПТОТИЧЕСКАЯ ОЦЕНКА ВЕРОЯТНОСТИ ТОГО, ЧТО КВАДРАТ ШУРА — АДАМАРА СЛУЧАЙНОГО ДЛИННОГО ЛИНЕЙНОГО КОДА ИМЕЕТ МАКСИМАЛЬНУЮ РАЗМЕРНОСТЬ

И. В. Чижов

МГУ имени М. В. Ломоносова,
Федеральный исследовательский центр «Информатика и управление» РАН,
АО «НПК „Криптонит“», г. Москва, Россия

E-mail: ichizhov@cs.msu.ru

Установлена оценка вероятности того, что квадрат Адамара (Шура — Адамара) случайного линейного кода размерности k и длины $n > k(k+1)/2$ имеет максимально возможную размерность. Оценка носит неасимптотический характер и поэтому может быть использована для обоснования сложности методов криптографического анализа постквантовых криптосистем, построенных на основе теории помехоустойчивого кодирования.

Ключевые слова: *произведение Шура линейных кодов, произведение Адамара линейных кодов, случайный код, квадрат Шура, квадрат Адамара, криптосистема Мак-Элиса.*

A NON-ASYMPTOTIC ESTIMATE OF THE PROBABILITY THAT A SHUR — HADAMARD SQUARE OF LONG RANDOM LINEAR CODE HAS A MAXIMUM DIMENSION

I. V. Chizhov

Lomonosov Moscow State University,
Federal Research Center “Computer Science and Control” of the RAS,
Joint Stock “Research and production company Kryptonite”, Moscow, Russia

Let $\mathbb{F}_q$ be a finite field of q elements. Let $\mathcal{V}_n(q)$ denote the vector space of length n over $\mathbb{F}_q$. Define a linear $[n,k]_q$-code $\mathcal{C}$ as any linear subspace of dimension k of the space $\mathcal{V}_n(q)$. This paper focuses on a special operation defined on the set of linear codes of the same length: the Schur — Hadamard product, also known as the Schur product or Hadamard product. The Schur — Hadamard product of two vectors $x = (x_1, \ldots, x_n) \in \mathcal{V}_n(q)$ and $y = (y_1, \ldots, y_n) \in \mathcal{V}_n(q)$ is defined as the vector $x \circ y = (x_1 \cdot y_1, \ldots, x_n \cdot y_n) \in \mathcal{V}_n(q)$, where $\cdot$ is the field $\mathbb{F}_q$ multiplication. Define the Schur — Hadamard square $\mathcal{C}^{\circ 2}$ of $[n]_q$-code $\mathcal{C}$ as the linear span of the set of vectors $\{c \circ b : c, b \in \mathcal{C}\}$. It is known that for any $[n,k]_q$-code $\mathcal{C}$ the inequality $\dim \mathcal{C}^{\circ 2} \leqslant \min\left(k(k+1)/2, n\right)$ holds. For a random linear code, the probability that its Schur — Hadamard square has the maximum possible dimension tends to 1 as $n, k \to \infty$. This fact is used in the analysis of code-based cryptosystems. However, in

practice researchers deal with fixed values of k and n. Therefore, the non-asymptotic estimation of the probability that the Schur — Hadamard square of a random $[n,k]_q$-code has the maximum possible dimension is of interest. In the case $n < k(k+1)/2$ such an estimate was obtained earlier. We provide a non-asymptotic estimate for the case $n > k(k+1)/2$. Two theorems are proven: the first gives an estimate for very long codes, while the second applies to relatively short codes. Let $k, n \in \mathbb{N}$ be such that $k \geqslant 5$ and $n > k(k+1)/2$. Then the following inequality holds:

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - q^{k(k+1)/2+\log_q 2-(2-\log_q(2q-1))n}.$$

If $k \geqslant 6$ and $n < (k^2-4k)/2(\log_q(2q-1)-1)$, then

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - q^{k(k+1)/2+\log_q 2-\left(1-\log_q\left(1+(q-1)q^{-\delta_q(n,k)k}\right)\right)n},$$

where $\delta_q(n,k) = \dfrac{1}{2} + \dfrac{1}{2k} - \dfrac{1}{2k}\sqrt{2k+1+2(\log_q(2q-1)-1)n}$. Finally, examples of estimates are given for different values of n, k and q.

Keywords: *Shur product of linear codes, Hadamard product of linear codes, random codes, Shur square of linear code, Hadamard square of linear code, McEliece public key cryptosystem.*

Введение

Значительный прогресс в области анализа кодовых постквантовых криптографических систем с открытым ключом не в последнюю очередь обязан применению такой операции над линейными кодами, как их покоординатное произведение, или произведение Шура — Адамара.

В алгебраической теории помехоустойчивого кодирования произведение Шура — Адамара появилось в 1992 г. В работе [1] операция покоординатного произведения линейных кодов использовалась для построения так называемых пар локаторов ошибок.

В области анализа кодовых криптосистем операция произведения Шура — Адамара использована впервые в [2], где построена первая полиномиальная атака на криптосистему Бергера — Луадро [3].

В 2013 г. в работе [4] был построен первый эффективный алгоритм, который позволяет отличать коды Гоппы с высокой скоростью передачи от случайных кодов. Выяснилось, что иногда произведение Шура — Адамара кодов, дуальных к кодам Гоппы, не заполняет собой всё пространство; при этом случайные коды длины n при возведении в квадрат Адамара должны совпадать со всем пространством векторов длины n.

В дальнейшем тот факт, что код, на основе которого построена кодовая криптосистема, ведёт себя относительно произведения Адамара не как случайный, был использован для построения атак на различные модификации криптосистемы Мак-Элиса [5–10].

В 2015 г. в работе [11] установлена асимптотическая оценка того, что квадрат Адамара случайного линейного кода имеет максимально возможную размерность. Таким образом, было показано, что при росте параметров кода найти случайно код, квадрат Адамара которого имеет не максимальную размерность, практически невозможно.

Однако в криптографических приложениях криптоаналитики имеют дело с фиксированными значениями параметров линейных кодов. И возникает вопрос, насколько эффективен тот или иной отличитель кода от случайного при этих параметрах, а не в бесконечности. В 2023 г. в работе [12] была применена техника, связанная с обобщённым расстоянием Хемминга кодов Рида — Маллера второго порядка, для получения

неасимптотической оценки вероятности того, что квадрат Адамара линейного кода заполняет собой всё пространство. Однако эта оценка становится тривиальной, если рассматриваются длинные линейные коды, т. е. такие коды, у которых длина больше половины квадрата размерности.

В настоящей работе изучается случай длинных кодов. Доказана неасимптотическая оценка вероятности того, что случайный линейный длинный код имеет максимально возможную размерность. Заметим, что в этом случае квадрат Адамара такого кода не будет заполнять всё пространство, так как в силу большой длины кода в квадрате Адамара не хватит кодовых слов.

Для получения неасимптотической оценки применена более простая техника, чем в работе [11], но новая оценка вполне может быть использована для изучения конкретных вариантов кодовых криптосистем с заранее фиксированным набором значений их параметров.

1. Основные термины и определения

Рассмотрим конечное поле $\mathbb{F}_q$, состоящее из q элементов. Обозначим через $\mathcal{V}_n(q)$ пространство векторов длины n над $\mathbb{F}_q$. Будем считать, что элементами $\mathcal{V}_n(q)$ являются векторы-строки с координатами из поля $\mathbb{F}_q$.

Следуя классической монографии [13], дадим основные определения из теории кодов, исправляющих ошибки.

Линейным блоковым кодом, исправляющим ошибки, или *линейным кодом*, или просто *кодом* над полем $\mathbb{F}_q$ будем называть произвольное подпространство $\mathcal{C}$ пространства $\mathcal{V}_n(q)$. Число n в этом случае называется *длиной* кода. Векторы c, принадлежащие $\mathcal{C}$, называются *кодовыми словами* (или просто *словами*) кода $\mathcal{C}$. Линейный код $\mathcal{C}$ длины n над полем $\mathbb{F}_q$ называется $[n]_q$-кодом.

Любой $[n]_q$-код $\mathcal{C}$ как линейное пространство имеет размерность k. Тогда k называется *размерностью* $\mathcal{C}$ и обозначается $\dim\mathcal{C}$. Кроме того, $[n]_q$-код $\mathcal{C}$ размерности k называется $[n,k]_q$-кодом.

Так как $[n,k]_q$-код $\mathcal{C}$ является линейным пространством, он может быть задан своим базисом. Матрица, строками которой являются базисные векторы кода $\mathcal{C}$, называется *порождающей матрицей*. Порождающая матрица $[n,k]_q$-кода $\mathcal{C}$ является $(k\times n)$-матрицей и имеет полный ранг, равный k. Получается, что произвольный $[n,k]_q$-код $\mathcal{C}$ с порождающей матрицей G может быть задан как множество всех линейных комбинаций строк матрицы G, т. е. $\mathcal{C}=\{a\cdot G:\ a\in\mathcal{V}_k(q)\}$.

Иногда удобно задавать $[n,k]_q$-код $\mathcal{C}$ некоторой $(\ell\times n)$-матрицей D, которая обладает таким же свойством, как и порождающая матрица: линейная комбинация строк D совпадает с кодом $\mathcal{C}$, т. е. $\mathcal{C}=\{a\cdot D:\ a\in\mathcal{V}_\ell(q)\}$. Матрица D называется *охватывающей матрицей* кода $\mathcal{C}$ [14]. Порождающая матрица G кода $\mathcal{C}$ является и охватывающей. Фактически порождающая матрица является частным случаем охватывающей, а именно: это охватывающая матрица, имеющая полный ранг. В общем случае ни порождающая, ни охватывающая матрицы не заданы однозначно. Однако каждая порождающая и каждая охватывающая матрица задаёт ровно один линейный код.

Для каждого вектора $v\in\mathcal{V}_n(q)$ обозначим через $\mathrm{wt}(v)$ его вес Хемминга, т. е. число ненулевых координат этого вектора. С $[n,k]_q$-кодом $\mathcal{C}$ свяжем характеристику $d_{\mathcal{C}}$, которая равна минимальному весу Хемминга ненулевых кодовых слов $\mathcal{C}$: $d_{\mathcal{C}}=\min\limits_{c\in\mathcal{C},\ c\neq 0}\mathrm{wt}(c)$. Число $d_{\mathcal{C}}$ называется *минимальным расстоянием* кода $\mathcal{C}$. В дальнейшем $[n,k]_q$-код $\mathcal{C}$ с минимальным расстоянием d будем называть $[n,k,d]_q$-кодом.

Объектом исследований настоящей работы является специальная операция, заданная на множестве линейных кодов одной длины: произведение Шура — Адамара (произведение Шура или произведение Адамара).

Произведением Адамара векторов $x = (x_1, x_2, \ldots, x_n) \in \mathcal{V}_n(q)$ и $y = (y_1, y_2, \ldots, y_n) \in \in \mathcal{V}_n(q)$ называется вектор $x \circ y \in \mathcal{V}_n(q)$, равный покомпонентному произведению этих векторов:

$$x \circ y = (x_1 \cdot y_1, x_2 \cdot y_2, \ldots, x_n \cdot y_n).$$

Здесь «$\cdot$» — произведение элементов поля $\mathbb{F}_q$.

Операцию произведения Адамара двух $[n]_q$-кодов $\mathcal{C}$ и $\mathcal{B}$ можно задать двумя эквивалентными способами [15].

П е р в ы й с п о с о б. *Произведением Адамара* двух $[n]_q$-кодов $\mathcal{C}$ и $\mathcal{B}$ называется $[n]_q$-код $\mathcal{C} \circ \mathcal{B}$, равный линейной оболочке множества векторов $\{c \circ b : c \in \mathcal{C}, b \in \mathcal{B}\}$.

В т о р о й с п о с о б — через базисы кодов $\mathcal{C}$ и $\mathcal{B}$. Пусть $c_1, c_2, \ldots, c_{k_\mathcal{C}}$ — базис $[n, k_\mathcal{C}]_q$-кода $\mathcal{C}$, а $\{b_1, b_2, \ldots, b_{k_\mathcal{B}}\}$ — базис $[n, k_\mathcal{B}]_q$-кода $\mathcal{B}$. Тогда произведением Адамара называется $[n, k]_q$-код $\mathcal{C} \circ \mathcal{B}$, который охватывается матрицей D, составленной из $k_\mathcal{C} \cdot k_\mathcal{B}$ строк $c_i \circ b_j$, $i = 1, \ldots, k_\mathcal{C}$ и $j = 1, \ldots, k_\mathcal{B}$.

Квадратом Шура — Адамара (или *квадратом Адамара*) $[n]_q$-кода $\mathcal{C}$ будем называть $[n]_q$-код $\mathcal{C}^{\circ 2}$, равный произведению Адамара кода $\mathcal{C}$ на себя: $\mathcal{C}^{\circ 2} = \mathcal{C} \circ \mathcal{C}$.

Утверждение 1 [11]. Для любого $[n, k]_q$-кода $\mathcal{C}$ выполняется неравенство

$$\dim \mathcal{C}^{\circ 2} \leqslant \min\left(k(k+1)/2, n\right). \tag{1}$$

В работе [11] установлено, что для случайного $[n, k]_q$-кода неравенство (1) обращается в равенство с вероятностью, которая стремится к 1 при $n, k \to \infty$. Однако в практических приложениях, особенно в задачах криптографического анализа, исследователи имеют дело с фиксированными значениями k и n, поэтому интерес представляет получение неасимптотической оценки вероятности того, что для случайного $[n, k]_q$-кода неравенство (1) становится равенством. В случае, когда $n < k(k+1)/2$, такая оценка вероятности получена ранее в [12]. Далее доказана неасимптотическая оценка в случае, когда $n > k(k+1)/2$.

2. Произведение Шура — Адамара линейного кода и квадратичные формы над конечным полем

Впервые связь между произведением Адамара линейных кодов и пространством квадратичных форм была установлена в работе [16]. Эта связь была использована авторами работы [11] для описания асимптотического поведения вероятности того, что квадрат Адамара случайного линейного кода достигает максимальной размерности. Этот же аппарат применён далее для получения неасимптотических оценок вероятности этого события.

Начнём с некоторых определений.

Квадратичной формой $Q(x_1, \ldots, x_k)$ над полем $\mathbb{F}_q$ называется однородный многочлен степени 2 от переменных $x_1, x_2, \ldots, x_k$ над тем же полем, т.е.

$$Q(x_1, \ldots, x_k) = \sum_{1 \leqslant i < j \leqslant k} a_{i,j} x_i x_j + \sum_{i=1}^{k} b_i x_i^2,$$

где $a_{i,j} \in \mathbb{F}_q$, $1 \leqslant i < j \leqslant k$; $b_i \in \mathbb{F}_q$, $1 \leqslant i \leqslant k$.

Множество всех квадратичных форм над полем $\mathbb{F}_q$ от k переменных будем обозначать как $\mathcal{Q}_k(q)$. Очевидно, что $\mathcal{Q}_k(q)$ является линейным пространством над полем $\mathbb{F}_q$. Базис этого пространства состоит из элементов

$$x_1^2,\ x_2^2,\ \ldots,\ x_k^2,\ x_1x_2,\ \ldots, x_{k-1}x_k.$$

Таким образом, $\mathcal{Q}_k(q)$ имеет размерность $k + \binom{k}{2} = \dfrac{k(k+1)}{2}$.

Зададим на множестве $\mathcal{V}_k(q)$ лексикографический порядок, упорядочив элементы поля $\mathbb{F}_q$ любым удобным способом. Этот порядок будем обозначать через $(\mathcal{V}_k(q), \leqslant) = \{x^{(0)} < x^{(1)} < \cdots < x^{(q^k-1)}\}$.

Для квадратичной формы $Q(x_1, \ldots, x_k) \in \mathcal{Q}_k(q)$ рассмотрим её вектор значений

$$\operatorname{eval}(Q) = \Big(Q(x^{(0)}), Q(x^{(1)}), \ldots, Q(x^{(q^k-1)})\Big) \in \mathcal{V}_{q^k}(q).$$

Матрица R, составленная из строк

$$\operatorname{eval}(x_1^2),\ \operatorname{eval}(x_2^2),\ \ldots,\ \operatorname{eval}(x_k^2),\ \operatorname{eval}(x_1x_2),\ \ldots,\ \operatorname{eval}(x_{k-1}x_k),$$

порождает некоторый $[q^k, k(k+1)/2, (q-1)^2q^{k-2}]_q$-код, который называется *однородным кодом Рида — Маллера* [17] и обозначается как $\mathcal{HRM}_q(2,k)$.

Каждый столбец матрицы R, как и каждая координата слова кода $\mathcal{HRM}_q(2,k)$, однозначно соответствует вектору $x^{(i)} \in (\mathcal{V}_q(n), \leqslant)$, поэтому столбцы этой матрицы и координаты кодовых слов можно занумеровать элементами частичного порядка $(\mathcal{V}_q(k), \leqslant)$; через $R_{x^{(i)}}$ будем обозначать столбец матрицы R с номером $x^{(i)}$.

Теперь, если $G = (g_1^\top g_2^\top \ldots g_n^\top)$ — $(k \times n)$-матрица над полем $\mathbb{F}_q$, составленная из столбцов $g_1^\top, g_2^\top, \ldots, g_n^\top$, то каждому g_i соответствует столбец R_{g_i} матрицы R, а всей матрице — подматрица R_G, составленная из столбцов R_{g_i}, $i = 1, 2, \ldots, n$.

Из [12, следствие 1] несложно вывести следующее

Утверждение 2. Для произвольных $k, n \in \mathbb{N}$, $k > 1$ и $n > k(k+1)/2$, рассмотрим некоторый $[n,k]_q$-код $\mathcal{C}$. Пусть G — его порождающая матрица. Тогда равенство $\dim \mathcal{C}^{\circ 2} = k(k+1)/2$ выполняется, если и только если $\operatorname{rank} R_G = k(k+1)/2$.

Утверждение 3. Пусть $G = (g_1^\top g_2^\top \ldots g_n^\top)$ — произвольная $(k \times n)$-матрица над полем $\mathbb{F}_q$ и $n > k(k+1)/2$. Тогда ранг матрицы R_G равен $k(k+1)/2$, если и только если для любой квадратичной формы $Q(x_1, \ldots, x_k) \in \mathcal{Q}_q(k)$, $Q \neq 0$, хотя бы для одного $i \in \{1, \ldots, k\}$ выполняется неравенство $Q(g_i) \neq 0$.

Из утверждений 2 и 3 прямо следует

Утверждение 4. Для произвольных $k, n \in \mathbb{N}$, $k > 1$ и $n > k(k+1)/2$, рассмотрим некоторый $[n,k]_q$-код $\mathcal{C}$. Пусть G — его порождающая матрица. Тогда равенство $\dim \mathcal{C}^{\circ 2} = k(k+1)/2$ выполняется, если и только если для любой квадратичной формы $Q(x_1, \ldots, x_k) \in \mathcal{Q}_k(q)$, $Q \neq 0$, хотя бы для одного $i \in \{1, \ldots, k\}$ верно неравенство $Q(g_i) \neq 0$.

В дальнейшем нам потребуется утверждение, которое задаёт значение спектра весов кода $\mathcal{HRM}_q(2,m)$ [17].

Утверждение 5. Пусть A_w — количество кодовых слов веса w в коде $\mathcal{HRM}_q(2,m)$. Тогда справедливы следующие равенства:

1) $A_0 = 1$;
2) $A_{q^m - q^{m-1}} = q^m - 1 + \sum\limits_{j=1}^{\lfloor (m-1)/2 \rfloor} q^{j^2+j} \prod\limits_{i=m-2j}^{m} (q^i - 1) \Big/ \prod\limits_{i=1}^{j} (q^{2i} - 1)$;
3) $A_{q^m - q^{m-1} - \tau q^{m-j-1}(q-1)} = \dfrac{q^{j^2+j} + \tau q^{j^2}}{2} \prod\limits_{i=m-2j+1}^{m} (q^i - 1) \Big/ \prod\limits_{i=1}^{j} (q^{2i} - 1)$, где $\tau \in \{-1, 1\}$ и $1 \leqslant j \leqslant \lfloor m/2 \rfloor$;
4) $A_w = 0$ для остальных w.

3. Оценка вероятности того, что случайный линейный код имеет максимальную размерность

Определим вероятностную схему $A_{n,k}(q)$ выбора случайного линейного $[n, k]_q$-кода. Будем последовательно случайно, равновероятно и независимо друг от друга выбирать векторы $g_1, g_2, \ldots, g_n$ из $\mathcal{V}_k(q)$. Сформируем из выбранных векторов матрицу $G = (g_1^\top, g_2^\top, \ldots, g_n^\top)$. Эта матрица охватывает некоторый $[n]_q$-код $\mathcal{C}$, который имеет размерность не более k. Тот факт, что код $\mathcal{C}$ построен описанным способом, будем обозначать следующим образом: $\mathcal{C} \xleftarrow{\$} A_{n,k}(q)$.

Другие вероятностные схемы обсуждаются в заключении.

Далее, используя утверждение 5, установим вероятность того, что случайно выбранный набор элементов из $\mathcal{V}_k(q)$ состоит из корней хотя бы одной квадратичной формы $Q \in \mathcal{Q}_k(q)$. На основании утверждения 3 эта вероятность определяет вероятность того, что случайный линейный код $\mathcal{C}$, выбранный по схеме $A_{n,k}(q)$, имеет максимально возможную размерность.

Утверждение 6. Пусть $k, n \in \mathbb{N}$ и $k > 1$. Зафиксируем произвольное $\delta \in \mathbb{R}$ таким образом, чтобы выполнялось неравенство $1 < \delta \cdot k \leqslant k/2$. Выберем $\mathcal{C} \xleftarrow{\$} A_{n,k}(q)$. Тогда справедливо неравенство

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - q^{2\delta(1-\delta)k^2 + 2\delta k + \log_q(\delta k) - a_q n} - q^{k(k+1)/2 - n(1 - \log_q(1 + (q-1)q^{-\delta k}))},$$

где $a_q = 2 - \log_q(2q-1)$.

Доказательство. Будем оценивать сверху вероятность противоположного события

$$\Gamma = \left[\dim \mathcal{C}^{\circ 2} \neq k(k+1)/2\right].$$

Если $G = (g_1^\top, g_2^\top, \ldots, g_n^\top)$ — порождающая матрица кода $\mathcal{C}$, то на основании утверждения 4 событие Γ возникает, если и только если существует квадратичная форма $Q \in \mathcal{Q}_k(q)$, $Q \neq 0$, которая обращается в нуль на всех столбцах матрицы G:

$$Q(g_1) = Q(g_2) = \ldots = Q(g_n) = 0.$$

Поэтому справедливо неравенство

$$\Pr[\Gamma] \leqslant \sum_{Q \in \mathcal{Q}_k(q), Q \neq 0} \Pr\left[Q(g_1) = Q(g_2) = \ldots = Q(g_n) = 0\right].$$

Теперь, так как в соответствии с распределением $A_{n,k}(q)$ каждый вектор g_i, $i = 1, \ldots, n$, выбирается случайно и равновероятно из $\mathcal{V}_k(q)$ независимо от других векторов, то

$$\Pr\left[Q(g_1) = Q(g_2) = \ldots = Q(g_n) = 0\right] = \prod_{i=1}^{n} \Pr\left[Q(g_i) = 0\right].$$

Но так как $\Pr\left[Q(g_i)=0\right]=1-\Pr\left[Q(g_i)\neq 0\right]$, $i=1,\ldots,n$, то выполнено равенство

$$\Pr\left[Q(g_1)=Q(g_2)=\ldots=Q(g_n)=0\right]=\prod_{i=1}^{n}\left(1-\Pr\left[Q(g_i)\neq 0\right]\right).$$

В силу выбора g_i равновероятно из всех векторов длины k над полем $\mathbb{F}_q$ верно равенство

$$\Pr\left[Q(g_i)\neq 0\right]=\mathrm{wt}(\mathrm{eval}(Q))/q^k,\quad i=1,\ldots,n.$$

Значит,

$$\Pr\left[Q(g_1)=Q(g_2)=\ldots=Q(g_n)=0\right]=\left(1-\mathrm{wt}(\mathrm{eval}(Q))/q^k\right)^n.$$

Отсюда получаем оценку

$$\Pr\left[\Gamma\right]\leqslant\sum_{Q\in\mathcal{Q}_k(q),Q\neq 0}\left(1-\frac{\mathrm{wt}(\mathrm{eval}(Q))}{q^k}\right)^n.$$

Зафиксируем некоторое $\delta\in\mathbb{R}$, $1/k<\delta\leqslant 1/2$. Множество всех квадратичных форм разобьём на два непересекающихся подмножества $\mathcal{Q}^0$ и $\mathcal{Q}^1$. Подмножество $\mathcal{Q}^0$ состоит из всех квадратичных форм $Q\in\mathcal{Q}_k(q)$, для которых вес вектора $\mathrm{eval}(Q)$ не больше $q^k-q^{k-1}-(q-1)q^{k-1-\delta k}$:

$$\mathcal{Q}^0=\{Q\in\mathcal{Q}_k(q):\mathrm{wt}(\mathrm{eval}(Q))\leqslant q^k-q^{k-1}-(q-1)q^{k-1-\delta k}\}.$$

В подмножество $\mathcal{Q}^1$ поместим оставшиеся $Q\in\mathcal{Q}_k(q)$:

$$\mathcal{Q}^1=\{Q\in\mathcal{Q}_k(q):\mathrm{wt}(\mathrm{eval}(Q))>q^k-q^{k-1}-(q-1)q^{k-1-\delta k}\}.$$

Заметив, что $0\in\mathcal{Q}^0$, можно записать

$$\sum_{Q\in\mathcal{Q},Q\neq 0}\left(1-\frac{\mathrm{wt}(\mathrm{eval}(Q)}{q^k}\right)^n=\sum_{Q\in\mathcal{Q}^0,Q\neq 0}\left(1-\frac{\mathrm{wt}(\mathrm{eval}(Q)}{q^k}\right)^n+\sum_{Q\in\mathcal{Q}^1}\left(1-\frac{\mathrm{wt}(\mathrm{eval}(Q))}{q^k}\right)^n.$$

Минимальное расстояние кода $\mathcal{HRM}_q(2,k)$ равно $d_{\mathcal{HRM}_q(2,k)}=(q-1)^2q^{k-2}$. Поэтому по определению числа $d_{\mathcal{HRM}_q(2,k)}$, если $Q\in\mathcal{Q}^0$ и $Q\neq 0$, то $\mathrm{wt}(\mathrm{eval}(Q))\geqslant d_{\mathcal{HRM}_q(2,k)}=$ $=(q-1)^2q^{k-2}$, а значит,

$$1-\frac{\mathrm{wt}(\mathrm{eval}(Q))}{q^k}\leqslant 1-\frac{(q-1)^2}{q^2}=\frac{2q-1}{q^2}.$$

Для $Q\in\mathcal{Q}^1$ по определению верно неравенство

$$\mathrm{wt}(\mathrm{eval}(Q))>(q-1)q^{k-1}-(q-1)q^{k-1-\delta k},$$

поэтому в этом случае

$$1-\frac{\mathrm{wt}(\mathrm{eval}(Q))}{q^k}<1-\frac{q-1}{q}(1-q^{-\delta k})=\frac{1}{q}(1+(q-1)q^{-\delta k}).$$

Следовательно, справедливо неравенство

$$\Pr\left[\Gamma\right]<\left(\frac{2q-1}{q^2}\right)^n|\mathcal{Q}^0\setminus\{0\}|+\frac{(1+(q-1)q^{-\delta k})^n}{q^n}|\mathcal{Q}^1|.$$

По построению $\mathcal{Q}^1 \subseteq \mathcal{Q}_k(q)$, поэтому верна тривиальная оценка $|\mathcal{Q}^1| \leqslant |\mathcal{Q}_k(q)| = q^{(k+1)k/2}$. Получаем неравенство

$$\Pr\left[\Gamma\right] < \left(\frac{2q-1}{q^2}\right)^n |\mathcal{Q}^0 \setminus \{0\}| + (1+(q-1)q^{-\delta k})^n\, q^{k(k+1)/2-n}.$$

Введя константу $a_q = 2 - \log_q(2q-1)$, будем иметь

$$\Pr\left[\Gamma\right] < q^{-a_q n}|\mathcal{Q}^0 \setminus \{0\}| + q^{k(k+1)/2-n(1-\log_q(1+(q-1)q^{-\delta k}))}. \tag{2}$$

Оценим сверху величину $|\mathcal{Q}^0\setminus\{0\}|$. В множестве $\mathcal{Q}^0$ лежат квадратичные формы Q, у которых вес вектора $\mathrm{eval}(Q)$ не больше, чем $q^k - q^{k-1} - (q-1)q^{k-1-\delta k}$. Так как $q > 1$, для любого δ справедливо неравенство $q^k - q^{k-1} - (q-1)q^{k-1-\delta k} < q^k - q^{k-1}$ и, кроме того, для любого j верно, что $q^k - q^{k-1} < q^k - q^{k-1} + (q-1)q^{k-1-j}$. Значит, согласно утверждению 5, если $Q \in \mathcal{Q}^0$, то либо $\mathrm{wt}(\mathrm{eval}(Q)) = 0$, либо $\mathrm{wt}(\mathrm{eval}(Q)) = q^k - q^{k-1} - (q-1)q^{k-1-j}$ для некоторого j, $1 \leqslant j \leqslant \lfloor k/2 \rfloor$. Из неравенства

$$q^k - q^{k-1} - (q-1)q^{k-1-j} \leqslant q^k - q^{k-1} - (q-1)q^{k-1-\delta k}$$

следует, что если $Q \in \mathcal{Q}^0$ и $\mathrm{wt}(\mathrm{eval}(Q)) \neq 0$, то $\mathrm{wt}(\mathrm{eval}(Q)) = q^k - q^{k-1} - (q-1)q^{k-1-j}$ для $j \in \mathbb{Z}$, $1 \leqslant j \leqslant \delta k$. Следовательно,

$$|\mathcal{Q}^0 \setminus \{0\}| = \sum_{j=1}^{\lfloor \delta k \rfloor} A_{q^k - q^{k-1} - (q-1)q^{k-1-j}}.$$

С учётом утверждения 5 получим следующее выражение:

$$|\mathcal{Q}^0 \setminus \{0\}| = \sum_{j=1}^{\lfloor \delta k \rfloor} \frac{q^{j^2+j} + q^{j^2}}{2} \prod_{i=k-2j+1}^{k} (q^i - 1) \Big/ \prod_{i=1}^{j} (q^{2i} - 1). \tag{3}$$

Так как по условию $\delta k > 1$, то $\lfloor \delta k \rfloor \geqslant 1$, поэтому сумма (3) содержит хотя бы одно ненулевое слагаемое.

Оценим сверху каждое слагаемое суммы (3) для $j = 1, \ldots, \lfloor \delta k \rfloor$.

При $q \geqslant 2$ и $i \geqslant 1$ верно неравенство $(q-1)q^{i-1} \geqslant 1$, поэтому $q^i - 1 \geqslant q^{i-1}$ для всех $i \geqslant 1$ и $q \geqslant 2$, а значит,

$$\prod_{i=1}^{j} (q^{2i} - 1) \geqslant \prod_{i=1}^{j} q^{2i-1} = q^{2\sum\limits_{i=1}^{j} i - j} = q^{j(j+1)-j} = q^{j^2}.$$

Тогда

$$q^{j^2}(q^j + 1) \Big/ \prod_{i=1}^{j} (q^{2i} - 1) \leqslant \frac{q^{j^2}(q^j+1)}{q^{j^2}} = q^j + 1.$$

Далее,

$$\prod_{i=k-2j+1}^{k} (q^i - 1) < \prod_{i=k-2j+1}^{k} q^i = q^{\sum\limits_{i=k-2j+1}^{k} i} = q^{j(2k-2j+1)}.$$

Таким образом, для $j \geqslant 1$ верно неравенство

$$\frac{q^{j^2+j} + q^{j^2}}{2} \frac{\prod\limits_{i=k-2j+1}^{k} (q^i - 1)}{\prod\limits_{i=1}^{j} (q^{2i} - 1)} = \frac{1}{2} \frac{q^{j^2}(q^j+1)}{\prod\limits_{i=1}^{j} (q^{2i} - 1)} \prod_{i=k-2j+1}^{k} (q^i - 1) < \frac{1}{2}(q^j + 1)q^{j(2k-2j+1)}.$$

Так как $q^j > 1$ при $q > 1$ и $j \geqslant 1$, то в этом случае $q^j > (q^j + 1)/2$. Получим оценку мощности $\mathcal{Q}^0 \setminus \{0\}$:

$$|\mathcal{Q}^0 \setminus \{0\}| < \sum_{j=1}^{\lfloor \delta k \rfloor} q^{2j(k-j+1)}.$$

Функция $f(x) = x(k - x + 1)$ на отрезке $0 \leqslant x \leqslant (k + 1)/2$ не убывает; δ выбрано таким образом, чтобы $\delta k \leqslant k/2 < (k + 1)/2$, поэтому для любого $1 \leqslant j \leqslant \lfloor \delta k \rfloor \leqslant \delta k$ выполняется неравенство $2j(k - j + 1) \leqslant 2\delta k(k - \delta k + 1)$. Значит,

$$|\mathcal{Q}^0 \setminus \{0\}| < \delta k q^{2\delta(1-\delta)k^2+2\delta k} = q^{2\delta(1-\delta)k^2+2\delta k+\log_q \delta k}.$$

Из неравенства (2) окончательно получим оценку вероятности события Γ:

$$\Pr\left[\Gamma\right] < q^{2\delta(1-\delta)k^2+2\delta k+\log_q(\delta k)-a_q n} + q^{k(k+1)/2-n(1-\log_q(1+(q-1)q^{-\delta k}))}.$$

С учётом $\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] = 1 - \Pr\left[\Gamma\right]$ получим требуемое неравенство. ■

Проанализируем оценку из утверждения 6.

Теорема 1. Зафиксируем числа $k, n \in \mathbb{N}$ таким образом, что $k \geqslant 5$ и $n > k(k+1)/2$. Выберем $\mathcal{C} \xleftarrow{\$} A_{n,k}(q)$. Тогда справедливо неравенство

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - q^{k(k+1)/2+\log_q 2-(2-\log_q(2q-1))n}.$$

Доказательство. Докажем, что существует δ, $1 < \delta k \leqslant k/2$, для которого выполняется неравенство

$$2\delta(1-\delta)k^2 + 2\delta k \leqslant k(k+1)/2. \tag{4}$$

Действительно, $2\delta(1-\delta)$ достигает максимального значения при $\delta = 1/2$, поэтому для любого $\delta \leqslant 1/2$ имеет место

$$2\delta(1-\delta)k^2 \leqslant k^2/2.$$

Если $k \geqslant 5$, то $1/k \leqslant 1/5 < \delta = 1/4 < 1/2$, поэтому, например, при $\delta = 1/4$ верно (4).

Далее докажем, что для любых k, $k > 0$, δ, $1/k < \delta \leqslant 1/2$, и любого q, $q > 1$, выполняется неравенство

$$1 - \log_q(1 + (q-1)q^{-\delta k}) > a_q = 2 - \log_q(2q-1).$$

Это неравенство эквивалентно следующему:

$$\log_q \frac{2q-1}{1+(q-1)q^{-\delta k}} > 1,$$

которое, в свою очередь, эквивалентно неравенству

$$\frac{2q-1}{1+(q-1)q^{-\delta k}} > q \quad \Leftrightarrow \quad 2q - 1 > q + (q-1)q^{1-\delta k}.$$

Упрощая и учитывая условие $q > 1$, придём к равносильному неравенству $1 > q^{1-\delta k}$, которое выполняется, если и только если $\delta k > 1$.

Итак, для выбранных k, δ и q выполняется условие

$$k(k+1)/2 - (1 - \log_q(1 + (q-1)q^{-\delta k}))n < k(k+1)/2 - a_q n.$$

Следовательно, при $k \geqslant 5$ существует такое δ, $1/k < \delta \leqslant 1/2$, что

$$q^{2\delta(1-\delta)k^2+2\delta k-a_q n} + q^{k(k+1)/2-(1-\log_q(1+(q-1)q^{-\delta k}))n} \leqslant 2q^{k(k+1)/2-a_q n}.$$

Из этого неравенства и утверждения 6 получим требуемую оценку вероятности. ■

В силу того, что при маленьких q константа $2-\log_q(2q-1)$ мала, оценка теоремы 1 имеет смысл только либо в случае достаточно больших q, либо для очень длинных кодов, у которых длина больше, чем $k(k+1)$. Для относительно коротких кодов оценку можно улучшить.

Теорема 2. Зафиксируем числа $k, n \in \mathbb{N}$ таким образом, что $k \geqslant 6$ и $n < \dfrac{k^2-4k}{2(\log_q(2q-1)-1)}$. Выберем $\mathcal{C} \xleftarrow{\$} A_{n,k}(q)$. Тогда справедливо неравенство

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - q^{k(k+1)/2+\log_q 2-(1-\log_q(1+(q-1)q^{-\delta_q(n,k)k}))n},$$

где

$$\delta_q(n,k) = \frac{1}{2} + \frac{1}{2k} - \frac{1}{2k}\sqrt{2k+1+2(\log_q(2q-1)-1)n}.$$

Доказательство. Найдём условия на n, k и q, при которых существует такое δ из полуинтервала $(k^{-1}, 2^{-1}]$, что выполнено неравенство

$$2\delta(1-\delta)k^2 + 2\delta k + \log_q(\delta k) - a_q n \leqslant k(k+1)/2 - n(1-\log_q(1+(q-1)q^{-\delta k})). \qquad (5)$$

Сначала заметим, что при $\delta \leqslant 1/2$, $k > 1$ и $q > 1$ верно $\log_q \delta k \leqslant k/2$, поэтому добьёмся выполнения условия

$$2\delta(1-\delta)k^2 + 2\delta k + k/2 - a_q n \leqslant k(k+1)/2 - n(1-\log_q(1+(q-1)q^{-\delta k})),$$

которое равносильно неравенству

$$2\delta(1-\delta)k^2 + 2\delta k - a_q n \leqslant k^2/2 - n(1-\log_q(1+(q-1)q^{-\delta k})).$$

При $q > 1$ имеет место $\log_q(1+(q-1)q^{-\delta k}) > 0$, поэтому

$$k^2/2 - n < k^2/2 - n(1-\log_q(1+(q-1)q^{-\delta k})).$$

Значит, достаточно найти условия, при которых существует $\delta \in (k^{-1}, 2^{-1}]$, гарантирующее выполнение равенства

$$2\delta(1-\delta)k^2 + 2\delta k - a_q n = k^2/2 - n.$$

Раскрывая скобки и приводя подобные слагаемые, получим квадратное уравнение относительно переменной δ:

$$2k^2\delta^2 - 2\delta(k^2+k) + k^2/2 - (1-a_q)n = 0.$$

Разделив уравнение на $2k^2 \neq 0$, получим

$$\delta^2 - \delta\left(1+\frac{1}{k}\right) + \frac{1}{4} - (1-a_q)\frac{n}{2k^2} = 0.$$

Решениями этого уравнения являются действительные числа δ_1 и δ_2:

$$\begin{aligned}\delta_1 &= \frac{1}{2} + \frac{1}{2k} - \frac{1}{2k}\sqrt{2k+1+2(1-a_q)n},\\ \delta_2 &= \frac{1}{2} + \frac{1}{2k} + \frac{1}{2k}\sqrt{2k+1+2(1-a_q)n}.\end{aligned} \tag{6}$$

Нетрудно понять, что $\delta_2 > 1/2$, так как $k > 0$. Значит, нужно искать такое δ_1, что $k^{-1} < \delta_1 \leqslant 2^{-1}$.

Условие $\delta_1 \leqslant 2^{-1}$ эквивалентно следующему: $1 \leqslant 2k + 1 + 2(1-a_q)n$. Так как при $q > 1$ выполняется неравенство $\log_q(2q-1) > 1$, то $1 - a_q = \log_q(2q-1) - 1 > 0$; значит, $2k + 2(1-a_q)n > 0$ и условие $\delta_1 \leqslant 2^{-1}$ верно при любых $n, k, q, q > 1$.

Осталось рассмотреть неравенство $k^{-1} < \delta_1$. Оно эквивалентно следующему:

$$\frac{1}{2k}\sqrt{2k+1+2(1-a_q)n} < \frac{1}{2} - \frac{1}{2k}.$$

Упрощая его, получим

$$2k + 1 + 2(1-a_q)n < (k-1)^2 = k^2 - 2k + 1 \Leftrightarrow 2(1-a_q)n < k^2 - 4k.$$

Таким образом, если $n < \dfrac{k^2-4k}{2(\log_q(2q-1)-1)}$, то найдётся такое $\delta_1 \in (k^{-1}, 2^{-1}]$, при котором выполнено (5).

Из неравенства (5) и утверждения 6 при $\delta_q(n,k) = \delta_1$ следует оценка вероятности

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - 2q^{k(k+1)/2 - (1-\log_q(1+(q-1)q^{-\delta_q(n,k)k}))n},$$

где через $\delta_q(n,k)$ обозначено δ_1 из (6). ■

4. Примеры

Рассмотрим несколько примеров.

Начнём с двоичных кодов. Пусть $n = 1600$ и $k = 50$. Из теоремы 1 следует оценка вероятности

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = k(k+1)/2\right] > 1 - 2^{50\cdot 51/2 + 1 - (2-\log_2 3)1600} \approx 1 - 2^{611},$$

т. е. в этом случае она тривиальная. Однако при выбранных параметрах верно неравенство

$$\frac{50^2 - 4\cdot 50}{2(\log_2(3)-1)} > 1983,$$

поэтому можно применить теорему 2. Вычислим $\delta_2(1600, 50)$:

$$\delta_2(1600,50) = \frac{1}{2} + \frac{1}{2\cdot 50} - \frac{1}{2\cdot 50}\sqrt{2\cdot 50 + 1 + 2\left(\log_2(3)-1\right)1600} > 0{,}0658.$$

Далее

$$1 - \log_2(1 + 2^{-0{,}0658\cdot 50}) < 0{,}85956.$$

Следовательно, из теоремы 2 следует, что для любого случайного $[1600, 50]$-кода $\mathcal{C}$ выполняется неравенство

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = 1275\right] > 1 - 2^{1275+1-0{,}85965\cdot 1600} > 1 - 2^{-99{,}44}.$$

Рассмотрим линейные коды размерности $k = 50$ и длины $n = 1600$, но уже над полем $\mathbb{F}_3$. Верно неравенство

$$\frac{50^2 - 4 \cdot 50}{2(\log_3(5) - 1)} > 2473.$$

Можно применить теорему 2. В этом случае

$$\delta_3(1600, 50) = \frac{1}{2} + \frac{1}{2 \cdot 50} - \frac{1}{2 \cdot 50}\sqrt{2 \cdot 50 + 1 + 2\,(\log_3(5) - 1)\,1600} > 0{,}111.$$

Далее

$$1 - \log_3(1 + 2 \cdot 3^{-0{,}111 \cdot 50}) < 0{,}996.$$

Значит, из теоремы 2 получим, что для любого случайного $[1600, 50]_3$-кода $\mathcal{C}$ выполняется неравенство

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = 1275\right] > 1 - 3^{1275 + \log_3(2) - 0{,}996 \cdot 1600} > 1 - 3^{-317{,}9} > 1 - 2^{-500}.$$

Нетрудно установить, для каких значений k и n оценка теоремы 2 является нетривиальной (строго больше нуля). Для этого нужно рассмотреть неравенство

$$k(k+1)/2 + \log_q 2 < (1 - \log_q(1 + (q-1)q^{-\delta_q(n,k)k}))n.$$

Оно эквивалентно неравенству

$$\delta_q(n,k)\,k > \log_q\left(\frac{q-1}{q^{1-k(k+1)/(2n)-(\log_q 2)/n} - 1}\right).$$

Подставляя значение $\delta_q(n,k)$, получим условие

$$\frac{k}{2} + \frac{1}{2} - \frac{1}{2}\sqrt{2k + 1 + 2(\log_q(2q-1) - 1)n} > \log_q\left(\frac{q-1}{q^{1-k(k+1)/(2n)-(\log_q 2)/n} - 1}\right),$$

которое должно выполняться, чтобы оценка имела смысл. С учётом неравенства $k \leqslant n$ получим

$$\frac{k}{2} + \frac{1}{2} - \frac{1}{\sqrt{2}}\sqrt{\log_q(2q-1) + 1} > \log_q\left(\frac{q-1}{q^{1-k(k+1)/(2n)-(\log_q 2)/n} - 1}\right).$$

Из этого неравенства видно, что при относительно больших k и n при условии, что отношение $k(k+1)/(2n)$ достаточно мало, правая часть неравенства стремится к нулю, а левая при этом идёт к бесконечности. Таким образом, при достаточно больших k оценка теоремы 2 имеет смысл для почти всех допустимых n, т. е. при $k(k+1)/2 < n < \dfrac{k^2 - 4k}{2(\log_q(2q-1) - 1)n}$.

В завершении рассмотрим пример применения оценки теоремы 1.

Выберем $k = 50$ и $n = 3200$. Тогда квадрат Адамара случайного двоичного $[3200, 50]$-кода $\mathcal{C}$ имеет максимально возможную размерность 1275 с вероятностью

$$\Pr\left[\dim \mathcal{C}^{\circ 2} = 1275\right] > 1 - 2^{1276 - (2 - \log_2 3) \cdot 3200} > 1 - 2^{-52{,}11}.$$

Если рассматриваются троичные коды, то квадрат Адамара случайного $[3200, 50]_3$-кода $\mathcal{C}$ будет иметь размерность 1275 с вероятностью

$$\Pr\left[\mathcal{C}^{\circ 2} = 1275\right] > 1 - 3^{1275 + \log_3(2) - (2 - \log_3 5) \cdot 3200} > 1 - 3^{-436{,}45} > 1 - 2^{-691{,}76}.$$

Таким образом, найти случайно код, у которого квадрат Адамара не имеет максимальную размерность, оказывается трудной задачей.

Заключение

В работе с использованием известных результатов о спектре весов однородных кодов Рида — Маллера второго порядка установлена оценка вероятности того, что случайный линейный код длины n и размерности не более k над полем из q элементов имеет максимальную размерность. При этом полученная оценка не является асимптотической и может быть использована в обосновании результатов криптографического анализа постквантовых криптографических систем, построенных на основе кодов, исправляющих ошибки.

Сделаем несколько замечаний.

Первое касается вероятностной модели. Все результаты получены в модели, в которой фактически случайный линейный код отождествляется со случайной $(k \times n)$-матрицей, которая охватывает его. Однако эта модель не учитывает, что на практике используются коды с фиксированной размерностью, а не просто ограниченной сверху некоторым числом.

Рассмотрим другую модель. Для построения случайного $[n, k]_q$-кода $\mathcal{C}$ выбирается случайно и равновероятно $(k \times n)$-матрица G максимального ранга из множества всех $(k \times n)$-матриц ранга k и в качестве $\mathcal{C}$ выбирается код, порождаемый матрицей G. Этот факт будем записывать как $\mathcal{C} \xleftarrow{\$} U_{n,k}(q)$.

Известно [11], что если для некоторого $a \in \mathbb{R}$ выполняется

$$\Pr_{\mathcal{C} \xleftarrow{\$} A_{n,k}(q)} [\dim \mathcal{C} = k(k+1)/2] > 1 - q^{-a(n,k,q)},$$

то верно неравенство

$$\Pr_{\mathcal{C} \xleftarrow{\$} U_{n,k}(q)} [\dim \mathcal{C} = k(k+1)/2] > 1 - q^{-a(n,k,q)} - q^{-(n-k)}.$$

Используя этот факт, можно скорректировать оценки теорем 1 и 2. Отметим, что в конечном итоге добавленное в оценку слагаемое является несоизмеримо малой величиной по сравнению с основным слагаемым.

Второе замечание касается случая таких параметров n, k и q, что оценки обеих теорем 1 и 2 являются тривиальными. Такое возможно при относительно малых k и n. Так, например, если $k = 40$ и $n = 1024$, то для двоичных кодов оценка теоремы 1 становится равной примерно $1 - 2^{396} < 0$, а оценка теоремы 2 — примерно $1 - 2^{19{,}89} < 0$. Таким образом, остаётся открытым вопрос установления более точной оценки вероятности для кодов, параметры которых достаточно малы.

ЛИТЕРАТУРА

1. *Pellikaan R.* On decoding by error location and dependent sets of error positions // Discrete Math. 1992. V. 106–107. P. 369–381.
2. *Wieschebrink C.* Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes // LNCS. 2010. V. 6061. P. 61–72.
3. *Berger T. and Loidreau P.* How to mask the structure of codes for a cryptographic use // Des. Codes Cryptogr. 2005. V. 35. P. 63–79.
4. *Faugère J., Gauthier-Umanã V., Otmani A., et al.* A distinguisher for high-rate McEliece cryptosystems // IEEE Trans. Inform. Theory. 2013. V. 59. No. 10. P. 6830–6844.
5. *Couvreur A., Gaborit P., Gauthier-Umanã V., et al.* Distinguisher-based attacks on public-key cryptosystems using Reed — Solomon codes // Des. Codes Cryptogr. 2014. V. 73. No. 2. P. 641–666.

6. *Otmani A. and Kalachi H.* Square code attack on a modified Sidelnikov cryptosystem // LNCS. 2015. V. 9084. P. 173–183.
7. *Couvreur A., Otmani A., Tillich J.-P., and Gauthier-Umanã V.* A polynomial-time attack on the BBCRS scheme // LNCS. 2015. V. 9020. P. 175–193.
8. *Бородин М. А., Чижов И. В.* Эффективная атака на криптосистему Мак-Элиса, построенную на основе кодов Рида — Маллера // Дискретная математика. 2014. Т. 26. № 1. С. 10–20.
9. *Чижов И. В., Попова Е. А.* Структурная атака на криптосистемы типа Мак-Элиса — Сидельникова, построенные на основе комбинирования случайных кодов с кодами Рида — Маллера // Intern. J. Open Inform. Technol. 2020. V. 8. No. 6. P. 24–33.
10. *Бородин М. А., Чижов И. В.* Классификация произведений Адамара подкодов коразмерности 1 кодов Рида — Маллера // Дискретная математика. 2020. Т. 32. № 1. С. 115–134.
11. *Cascudo I., Cramer R., Mirandola D., and Zemor G.* Squares of random linear codes // IEEE Trans. Inform. Theory. 2015. V. 61. No. 3. P. 1159–1173.
12. *Чижов И. В.* Квадрат Адамара и обобщённое минимальное расстояние кода Рида — Маллера порядка 2 // Дискретная математика. 2023. Т. 35. № 1. С. 128–152.
13. *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979. 744 с.
14. *Hall J. I.* Notes on Coding Theory. `https://users.math.msu.edu/users/halljo/classes/CODENOTES/CODING-NOTES.HTML`. 2010.
15. *Чижов И. В.* Полная классификация произведений Адамара подкодов коразмерности 1 кодов Рида — Маллера // Вестник Московского университета. Сер. 15: Вычислительная математика и кибернетика. 2024. № 1. С. 67–80.
16. *Randriambololona H.* On products and powers of linear codes under componentwise multiplication // Algorithmic Arithmetic, Geometry, and Coding Theory. 2015. V. 637. P. 3–78.
17. *Shuxing L.* On the weight distribution of second order Reed — Muller codes and their relatives // Des. Codes Cryptogr. 2019. V. 87. No. 10. P. 2447–2460.

REFERENCES

1. *Pellikaan R.* On decoding by error location and dependent sets of error positions. Discrete Math., 1992, vol. 106–107, pp. 369–381.
2. *Wieschebrink C.* Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. LNCS, 2010, vol. 6061, pp. 61–72.
3. *Berger T. and Loidreau P.* How to mask the structure of codes for a cryptographic use. Des. Codes Cryptogr., 2005, vol. 35, pp. 63–79.
4. *Faugère J., Gauthier-Umanã V., Otmani A., et al.* A distinguisher for high-rate McEliece cryptosystems. IEEE Trans. Inform. Theory, 2013, vol. 59, no. 10, pp. 6830–6844.
5. *Couvreur A., Gaborit P., Gauthier-Umanã V., et al.* Distinguisher-based attacks on public-key cryptosystems using Reed — Solomon codes. Des. Codes Cryptogr., 2014, vol. 73, no. 2, pp. 641–666.
6. *Otmani A. and Kalachi H.* Square code attack on a modified Sidelnikov cryptosystem. LNCS, 2015, vol. 9084, pp. 173–183.
7. *Couvreur A., Otmani A., Tillich J.-P., and Gauthier-Umanã V.* A polynomial-time attack on the BBCRS scheme. LNCS, 2015, vol. 9020, pp. 175–193.
8. *Borodin M. A. and Chizhov I. V.* Effective attack on the McEliece cryptosystem based on Reed — Muller codes. Discrete Math. Appl., 2014, vol. 24, no. 5, pp. 273–280.

9. *Chizhov I. V. and Popova E. A.* Strukturnaya ataka na kriptosistemy tipa Mak-Elisa — Sidel'nikova, postroennye na osnove kombinirovaniya sluchaynykh kodov s kodami Rida — Mallera [Structural attack on McEliece-Sidelnikov cryptosystems built on the combining of random codes with Reed-Muller codes]. Intern. J. Open Inform. Technol., 2020, vol. 8, no. 6, pp. 24–33. (in Russian)
10. *Borodin M. A. and Chizhov I. V.* Classification of Hadamard products of one-codimensional subcodes of Reed — Muller codes. Discrete Math. Appl., 2022, vol. 32, no. 5, pp. 297–311.
11. *Cascudo I., Cramer R., Mirandola D., and Zemor G.* Squares of random linear codes. IEEE Trans. Inform. Theory, 2015, vol. 61, no. 3, pp. 1159–1173.
12. *Chizhov I. V.* Hadamard square of linear codes and the generalized minimal distance of Reed — Muller code of order 2. Discrete Math. Appl., 2025, vol. 35, no. 1, pp. 15–34. 1
13. *McWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. Parts I and II. Amsterdam, North-Holland Publ., 1977. 762 p.
14. *Hall J. I.* Notes on Coding Theory. `https://users.math.msu.edu/users/halljo/classes/CODENOTES/CODING-NOTES.HTML`, 2010.
15. *Chizhov I. V.* Polnaya klassifikatsiya proizvedeniy Adamara podkodov korazmernosti 1 kodov Rida—Mallera [Complete classification of Hadamard products of codimension 1 subcodes of Reed-Muller codes]. Bulletin of Moscow University. Ser. 15: Comput. Math. and Cybernetics, 2024, no. 1, pp. 67–80. (in Russian)
16. *Randriambololona H.* On products and powers of linear codes under componentwise multiplication. Algorithmic Arithmetic, Geometry, and Coding Theory, 2015, vol. 637, pp. 3–78.
17. *Shuxing L.* On the weight distribution of second order Reed — Muller codes and their relatives. Des. Codes Cryptogr., 2019, vol. 87, no. 10, pp. 2447–2460.

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.174.2 DOI 10.17223/20710410/68/5

ОБХОДЫ ГРАФОВ, РЕАЛИЗУЕМЫЕ ИТЕРАЦИОННЫМИ МЕТОДАМИ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

А. В. Пролубников

Новосибирский государственный университет, г. Новосибирск, Россия

E-mail: a.v.prolubnikov@mail.ru

Обходы графов используются для решения многих задач. Обычные варианты обхода графа — это поиск в глубину и в ширину. При обходе связного графа последовательно достигаются все его вершины в результате переходов по рёбрам. Поиск в ширину — обычный выбор при построении эффективных алгоритмов нахождения компонент связности графа. Методы простой итерации для решения систем линейных алгебраических уравнений с модифицированными матрицами смежности графов и заданной правой частью могут быть рассмотрены как алгоритмы обхода графа. Эти алгоритмы дают обходы, вообще говоря, отличные от обходов графа в глубину и в ширину. Примером такого алгоритма является алгоритм обхода графа, который даёт метод Гаусса — Зейделя. Для произвольного связного графа этому алгоритму требуется количество итераций не большее, чем для обхода в ширину. Для большого количества индивидуальных задач достаточно меньшего числа итераций.

Ключевые слова: *обходы графов, задачи о связности на графах.*

GRAPH TRAVERSALS IMPLEMENTED BY ITERATIVE METHODS FOR SOLVING SYSTEMS OF LINEAR EQUATIONS

A. V. Prolubnikov

Novosibirsk State University, Novosibirsk, Russia

Graph traversals, such as depth-first search and breadth-first search, are commonly used to solve many problems on graphs. By implementing a graph traversal, we consequently reach all graph vertices that belong to a connected component. The breadth-first search is the usual choice when constructing efficient algorithms for finding connected components of a graph. Methods of simple iteration for solving systems of linear equations with modified graph adjacency matrices can be considered as graph traversal algorithms if we use a properly specified right-hand side. These traversal algorithms, generally speaking, turn out to be neither equivalent to depth-first search nor to breadth-first search. An example of such a traversal algorithm is the one associated with the Gauss — Seidel method. For an arbitrary connected graph, the algorithm requires no more iterations to visit all its vertices than it takes for breadth-first search. For a large number of instances of the problem, fewer iterations will be required.

Keywords: *graph traversals, connectivity problems on graphs.*

Введение

Многие задачи, связанные с надёжностью транспортных сетей, сетей передачи данных, больших интегральных схем и др. формулируются как задачи о связности на графах. Примеры таких задач — задачи нахождения компонент связности, точек сочленения, мостов и др.

Пусть G — обыкновенный граф, то есть неориентированный невзвешенный граф без кратных рёбер и петель; V — множество его вершин, E — множество рёбер, имеющие соответственно мощности n и m. Вершины графа помечены (пронумерованы) в некотором произвольном порядке от 1 до n. *Компонента связности* графа — это максимальный по включению его связный подграф. Компонента связности определяется множеством вершин такого подграфа.

Обход графа представляет собой итеративный процесс, в ходе которого, начиная с некоторой *стартовой вершины*, производятся переходы по рёбрам графа. Обход графа завершается, когда посещены все его вершины. Под реализацией обхода понимается процесс получения последовательности вершин, которые посещаются в результате переходов по рёбрам.

Обычными вариантами реализации обхода графов для решения задач о связности являются поиск в глубину и поиск в ширину. *Поиск в глубину* (Depth-First Search, DFS) [1] представляет собой рекурсивную процедуру, в ходе которой производятся переходы через все вершины, смежные текущей достигнутой вершине. Если через ребро достижима ещё не посещённая вершина, то через него совершается переход в эту вершину, из которой рекурсивно запускается алгоритм обхода. Возврат из рекурсии происходит, если для текущей вершины среди смежных ей нет ещё не посещённых.

Для вычислительно эффективного нахождения компонент связности больших графов обычно используется *поиск в ширину* (Breadth-First Search, BFS) [1]. При проведении такого обхода, начиная со стартовой вершины, принадлежащей компоненте связности графа, производятся переходы в ещё не посещённые вершины, смежные вершинам, посещённым на предыдущей итерации. В ходе таких итераций строится *дерево достижимости* графа; $(k+1)$-му уровню этого дерева принадлежат вершины, достигаемые в результате последовательных переходов по k рёбрам некоторой простой цепи в графе. Построение дерева достижимости для компоненты связности производится за $\ell_{\max}$ итераций, где $\ell_{\max}$ — длина кратчайшей простой цепи, соединяющей стартовую вершину с наиболее удалённой от неё вершиной.

BFS был впервые использован К. Цузе в 1945 г., но не был опубликован с указанием автора [2] до 1972 г. Впервые BFS опубликован Э. Муром в 1959 г. в контексте поиска пути в лабиринте [3]. Позже его независимо от Мура предложил Ч. Ли в контексте разводки проводников на печатных платах [4].

BFS и его обобщения для взвешенных графов лежат в основе многих алгоритмов решения задач дискретного анализа и дискретной оптимизации. Обходы используются для нахождения дерева кратчайших путей в графе, проверки графа на двудольность, нахождения максимального потока в сети и др.

Пусть $s \in V$ — стартовая вершина. *Алгебраический BFS* представляет собой реализацию BFS через последовательное умножение вектора на матрицу смежности A графа:

$$x^{(k+1)} = Ax^{(k)}. \tag{1}$$

Здесь $x^{(0)} = e_s$; e_s — s-й единичный вектор стандартного базиса в $\mathbb{R}^n$, то есть все его компоненты нулевые, за исключением s-й. На каждой итерации некоторые компонен-

ты этого вектора становятся ненулевыми. Индексы таких компонент соответствуют посещённым на этой итерации вершинам.

Алгебраический BFS реализован в библиотеках программ, наиболее эффективно использующих возможности современных компьютерных архитектур для параллельных вычислений и оптимизации работы с памятью. Они предназначены для работы с разреженными графами, то есть с теми, для которых $m = O(n)$ и которые наиболее часто встречаются в приложениях. Такие низкоуровневые реализации алгебраического BFS при решении практических задач требуют меньше времени для обхода графа, чем реализации теоретически оптимального комбинаторного BFS [5–9]. Поскольку вычислительная сложность BFS составляет $O(m + n)$, для разреженных графов алгебраический BFS позволяет получать реализации с наименьшей возможной для задачи линейной вычислительной сложностью [10].

Однако несмотря на то, что вычисления (1) допускают эффективные параллельные реализации при нахождении текущего уровня дерева достижимости, не существует реализаций BFS с вычислительной сложностью меньше линейной, то есть имеющих сложность $O(n^c)$, где $0 < c < 1$.

Таким образом, имеется два принципиально отличных друг от друга подхода к численной реализации обхода графа — комбинаторный и алгебраический (линейно-алгебраический).

При реализации *комбинаторного* подхода, начиная с некоторой стартовой вершины, последовательно рассматриваются возможные варианты переходов по рёбрам, инцидентным текущей достигнутой вершине. После этого совершаются переходы в ещё не посещённые вершины по некоторым выбранным рёбрам.

При реализации *алгебраического* подхода на каждой итерации производится линейное преобразование *вектора состояния* $x^{(k)}$. Анализ компонент этого вектора позволяет определить вершины, посещённые в ходе итерации. Посещение вершины $i \in V$ регистрируется в случае, если значение i-й компоненты вектора состояния перестаёт быть нулевым. Примером реализации алгебраического подхода является алгебраический BFS, который в качестве линейного преобразования использует умножение вектора на матрицу смежности графа.

Рассматриваемые нами алгоритмы обхода графа используют другие линейные преобразования. Эти преобразования реализуются методами простой итерации решения систем линейных алгебраических уравнений (СЛАУ). Варианты метода простой итерации решения СЛАУ с модифицированными матрицами смежности графов и заданной правой частью могут быть рассмотрены как обходы графов. Эти методы дают два варианта обхода графа, первый из которых реализуется методом Якоби и эквивалентен BFS, второй — методом Гаусса — Зейделя, и он не эквивалентен BFS.

Назовём простую цепь в графе *правильной*, если номера её вершин образуют возрастающую последовательность. В отличие от DFS и BFS, при получении обхода графа с помощью итераций метода Гаусса — Зейделя учитывется нумерация вершин в графе, которая является обязательным параметром задачи. Такой алгоритм обхода не эквивалентен ни BFS, ни DFS. На каждой его итерации сначала производится итерация BFS и если при этом были достигнуты вершины, из которых начинаются правильные цепи, то производятся переходы по всем рёбрам этих правильных цепей.

Для обхода произвольного связного графа с помощью такого алгоритма нужно произвести не больше итераций, чем требуется для BFS. При этом для большого количества графов понадобится меньше итераций. Если в графе имеется достаточно много

правильных цепей, то даже последовательная реализация такого алгоритма может давать обход графа быстрее, чем параллельная версия BFS.

1. Итеративные численные методы решения СЛАУ и обходы графов, ассоциированные с ними

Рассмотрим итерации методов Якоби и Гаусса — Зейделя для решения СЛАУ

$$Ax = b. \tag{2}$$

Оба метода являются вариантами метода простой итерации. Итерации метода Якоби имеют следующий вид:

$$x^{(k+1)} = b - D^{-1}Ax^{(k)},$$

где D — диагональная матрица с диагональными элементами матрицы A. То есть для i-й компоненты приближённого решения, получаемого после $(k+1)$-й итерации, имеем

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right). \tag{3}$$

Итерации метода Гаусса — Зейделя имеют следующий вид:

$$(L + D)x^{(k+1)} = -Ux^{(k)} + b,$$

где L и U — матрицы с элементами матрицы A, находящимися соответственно под её диагональю и над ней, то есть

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right). \tag{4}$$

Для того чтобы СЛАУ с матрицами графов имели решение, а используемые численные методы их решения сходились, мы можем модифицировать матрицы смежности, заменяя нулевые элементы на их диагонали на значения $d > 0$, как это делается в [11, 12]. Если матрица A имеет диагональное преобладание, то есть для $i = 1, \ldots, n$ имеем

$$|a_{ii}| = d \geqslant \sum_{i \neq j} |a_{ij}|,$$

где суммирование ведётся по j от 1 до n, и если хотя бы одно из этих неравенств строгое, то приближённые решения, получаемые на итерациях (3) и (4), сходятся к точному решению, которое существует и единственно для любой правой части b.

Рассмотрим СЛАУ (2), где A — модифицированная матрица смежности с диагональным преобладанием и $b = e_s$. Алгоритмы обхода графа, рассматриваемые далее, производят итерации (3) и (4). После проведения не более чем n итераций (3) или (4) с начальным вектором $x^{(0)} = d \cdot e_s$, возможно, без достижения сходимости к точному решению СЛАУ (2), мы получаем последовательность приближённых решений $x^{(k)}$. Далее будем рассматривать их как текущее значение вектора состояния $x^{(k)}$ на k-й итерации, нежели как приближённые решения СЛАУ. Для проведения обходов графов с помощью методов решения СЛАУ диагональное преобладание не обязательно. Достаточно того, чтобы на диагонали модифицированной матрицы смежности графа были ненулевые значения, равные некоторому заданному d.

Переходы между вершинами графа в ходе этих итераций регистрируются следующим образом. Переход из вершины, достигнутой на предыдущей итерации, в вершину $i \in V$ происходит на $(k+1)$-й итерации, если выполнено следующее условие:

$$(x_i^{(k)} = 0) \quad \text{и} \quad (x_i^{(k+1)} \neq 0). \tag{5}$$

То есть вершина i не была достигнута на итерациях, предшествующих $(k+1)$-й итерации, и была достигнута после её выполнения. Вершина j, из которой был совершён переход в вершину i по ребру графа, может быть определена в ходе выполнения итерации (3) или (4). *Фронтир* $\mathcal{F}^{(k+1)}$ для $(k+1)$-й итерации — это множество вершин, которые были достигнуты в результате её выполнения; $\mathcal{F}^{(0)} = \{s\}$.

Производя итерации метода Якоби или метода Гаусса — Зейделя с заданными начальным значением вектора состояния и правой частью СЛАУ (2), последовательно получаем фронтиры для соответствующих им обходов графа, в результате посещая все вершины связного графа.

Методы простой итерации решения СЛАУ дают два отличных друг от друга алгоритма обхода графа. Один из них может быть получен проведением итераций (3) и, как мы покажем далее, он эквивалентен BFS, тогда как другой может быть получен проведением итераций (4), и он не эквивалентен BFS.

2. Использование операции умножения вместо операции деления в итерациях методов Якоби и Гаусса — Зейделя

Поскольку для проведения обхода графа с помощью итераций (3) и (4) нет необходимости добиваться сходимости к точному решению СЛАУ (2), вместо операции деления в (3) и (4) может быть использовано умножение. Не влияя на обходы, производимые при выполнении итераций, и на доказательства, которые проводятся далее, это позволяет сделать текст более компактным. После такой модификации значения $x_i^{(k+1)}$ становятся полиномами от d, а не от $1/d$, что имеет место при использовании деления.

Более того, такая модификация имеет и практический смысл, поскольку реализация операции деления машинных чисел с плавающей точкой в несколько раз медленней, чем реализация операции умножения. В результате уменьшается время, требуемое для проведения обхода.

Таким образом, далее, сопоставляя графу матрицу смежности, в которой нулевые диагональные элементы заменены некоторым $d > 0$, мы рассматриваем итерации двух методов как соответственно

$$x_i^{(k+1)} = \left(-b_i + \sum_{j \neq i} a_{ij} x_j^{(k)} \right) (-d); \tag{6}$$

$$x_i^{(k+1)} = \left(-b_i + \sum_{l=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) (-d). \tag{7}$$

Здесь d может быть произвольным положительным значением.

3. Комбинаторные варианты алгоритмов обхода графа

Цепью в графе называется конечная последовательность вершин, в которой каждая вершина соединена с последующей ребром. Цепь может пониматься и как набор этих рёбер. *Простые* цепи — это цепи без повторяющихся вершин. *Длина* цепи c, которую мы обозначаем как $\ell(c)$, — количество рёбер в ней. Цепь с совпадающими начальной и последней вершиной называется *циклом*. Мы называем цепь *правильной*, если

номера вершин в ней образуют возрастающую последовательность, то есть это цепь $c = \{i_0, i_1, \ldots, i_k\}$, $i_j \in V$, такая, что $(i_{j-1}, i_j) \in E$, $i_{j-1} < i_j$, $j = 1, \ldots, k$. Правильные цепи — это простые цепи. *Маршрут* — это такая цепь, в которой могут повторяться как вершины, так и рёбра.

Назовём *поиском правильных цепей* (Correct Chain Search, CCS) алгоритм обхода графа, получаемый в ходе итераций (7). Мы рассматриваем и BFS, и CCS как алгоритмы двух типов: комбинаторные (описание приведено в алгоритмах 1 и 2) и алгебраические (общая схема даётся в п. 4 алгоритмом 3).

Сходство BFS и CCS состоит в том, что, производя итерации (7), как и при выполнении итерации BFS, мы производим переходы по рёбрам, инцидентным вершинам, достигнутым на предыдущей итерации. Разница между ними в том, что если после этих переходов мы оказываемся в вершинах, из которых исходят правильные цепи, то в случае CCS на той же итерации будут произведены переходы по всем рёбрам, принадлежащим этим цепям. Это происходит потому, что, производя итерации (7), мы производим вычисление компонент текущего вектора состояния, используя значения компонент, которые уже рассчитаны ранее на текущей итерации. Так, если вершина j смежна вершине i, $j < i$, и j-я компонента вектора состояния перестала быть нулевой на текущей итерации, то на этой же итерации её значение будет использовано при вычислении i-й компоненты вектора состояния и она также перестанет быть нулевой, если она была таковой до этого.

При выполнении итераций BFS вектор состояния обновляется только исходя из результатов отдельных вычислений для каждой его компоненты. Это позволяет эффективно распараллеливать вычисления при умножении вектора на матрицу, но при этом в новый фронтир попадают только вершины, смежные вершинам фронтира, полученного на предыдущей итерации.

Пусть $\mathcal{N}$ обозначает множество вершин, смежных вершинам из $\mathcal{F}^{(k)}$ и не посещённых после k итераций алгоритма; $\mathcal{C}^{(k)}$ — множество вершин из V, посещённых после k итераций.

Алгоритм 1. Комбинаторный BFS

Вход: граф G, стартовая вершина $s \in V$.
Выход: компонента связности $\mathcal{C}$.

1: $k := 0$, $x^{(0)} := e_s$, $\mathcal{F}^{(0)} := \{s\}$, $\mathcal{C}^{(0)} := \{s\}$, $\mathcal{C}^{(1)} := \varnothing$.
2: **Пока** $\mathcal{C}^{(k)} \neq \mathcal{C}^{(k+1)}$:
3: $\quad \mathcal{N} := \big\{i \in V \setminus \mathcal{C}^{(k)} : \exists j \in \mathcal{F}^{(k)}\big((i,j) \in E\big)\big\}$;
4: $\quad \mathcal{F}^{(k+1)} := \mathcal{N}$;
5: $\quad \mathcal{C}^{(k+1)} := \mathcal{C}^{(k)} \cup \mathcal{F}^{(k+1)}$;
6: $\quad k := k + 1$.
7: $\mathcal{C} := \mathcal{C}^{(k+1)}$.

На итерации CCS в дополнение к переходам по рёбрам, инцидентным вершинам из $\mathcal{F}^{(k)}$, в вершины $\mathcal{F}^{(k+1)}$ производятся переходы по рёбрам правильных цепей, исходящим из уже достигнутых на этой итерации вершин $\mathcal{F}^{(k+1)}$. Пусть $C(\tilde{s}, i)$ — множество правильных цепей, исходящих из вершин $\tilde{s} \in \mathcal{N}$ и заканчивающихся в вершине i.

Таким образом, на шаге 3 обоих алгоритмов производятся переходы по рёбрам, инцидентным вершинам фронтира $\mathcal{F}^{(k)}$, полученного на предыдущей итерации. Дополнительные переходы по рёбрам правильных цепей из $C(\tilde{s}, i)$ выполняются на шаге 4 алгоритма 2.

Алгоритм 2. Комбинаторный CCS

Вход: граф G, стартовая вершина $s \in V$.
Выход: компонента связности $\mathcal{C}$.
1: $k := 0$, $x^{(0)} := e_s$, $\mathcal{F}^{(0)} := \{s\}$, $\mathcal{C}^{(0)} := \{s\}$, $\mathcal{C}^{(1)} := \varnothing$.
2: **Пока** $\mathcal{C}^{(k)} \neq \mathcal{C}^{(k+1)}$:
3: $\quad \mathcal{N} := \big\{i \in V \setminus \mathcal{C}^{(k)} : \exists j \in \mathcal{F}^{(k)}\big((i,j) \in E\big)\big\}$;
4: $\quad \mathcal{F}^{(k+1)} := \mathcal{N} \cup \big\{i \in V \setminus (\mathcal{C}^{(k)} \cup \mathcal{N}) : \exists \tilde{s} \in \mathcal{N}\big(C(\tilde{s}, i) \neq \varnothing\big)\big\}$;
5: $\quad \mathcal{C}^{(k+1)} := \mathcal{C}^{(k)} \cup \mathcal{F}^{(k+1)}$;
6: $\quad k := k + 1$.
7: $\mathcal{C} := \mathcal{C}^{(k+1)}$.

4. Алгебраические версии комбинаторных BFS и CCS, реализуемые как итерации методов Якоби и Гаусса — Зейделя

Обозначим через $\mathbf{F}\big(x^{(k)}\big)$ преобразование вида (6) или (7). Для заданной стартовой вершины s алгоритм 3 даёт компоненту связности $\mathcal{C}$, которой принадлежит s, производя обход этой компоненты связности. В случае, когда $\mathbf{F}$ — преобразование вида (6), этот алгоритм, как показано далее, является вариантом алгебраического BFS; при использовании $\mathbf{F}$ вида (7) получаем алгебраический CCS.

Алгоритм 3. Обход компоненты связности

Вход: граф G, стартовая вершина $s \in V$.
Выход: компонента связности $\mathcal{C}$.
1: $k := 0$, $x^{(0)} := e_s$, $\mathcal{F}^{(0)} := \{s\}$, $\mathcal{C}^{(0)} := \{s\}$, $\mathcal{C}^{(1)} := \varnothing$.
2: **Пока** $\mathcal{C}^{(k)} \neq \mathcal{C}^{(k+1)}$:
3: $\quad x^{(k+1)} := \mathbf{F}\big(x^{(k)}\big)$;
4: $\quad \mathcal{F}^{(k+1)} := \big\{i \in V(G) : (x_i^{(k)} = 0) \wedge (x_i^{(k+1)} \neq 0)\big\}$;
5: $\quad \mathcal{C}^{(k+1)} := \mathcal{C}^{(k)} \cup \mathcal{F}^{(k+1)}$;
6: $\quad k := k + 1$.
7: $\mathcal{C} := \mathcal{C}^{(k)}$.

Алгоритм 4 находит все компонент связности графа G. Компоненты связности обозначаются $\mathcal{C}_i$, $i = 1, \ldots, K$, где K — их количество.

Алгоритм 4. Нахождение всех компонент связности

Вход: граф G.
Выход: все компоненты связности $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$.
1: $V' := \varnothing$, $K := 1$.
2: **Пока** $V' \neq V$:
3: $\quad$ выбрать $s \in V \setminus V'$;
4: $\quad \mathcal{C}_K :=$ Обход компоненты связности (G, s);
5: $\quad V' := V' \cup \mathcal{C}_K$;
6: $\quad K := K + 1$.

Последовательно выбирая стартовые вершины для обходов на шаге 3 алгоритма 4, мы находим все компоненты связности графа.

Покажем, что итерации (6) дают алгоритм обхода графа, эквивалентный комбинаторному BFS, тогда как итерации (7) — комбинаторному CCS.

5. Итерации метода Якоби как реализация комбинаторного BFS

Пусть $C(s,i)$ обозначает множество простых цепей, соединяющих стартовую вершину s и вершину $i \in \mathcal{F}^{(k+1)}$. Если $C(s,i) \neq \varnothing$, то все они имеют длину k: $\ell(c) = k$ для всех $c \in C(s,i)$.

Лемма 1. Для итераций (6)

$$x_i^{(k+1)} = \sum_{c \in C(s,i)} (-1)^{\ell(c)+1} d^{\ell(c)+2},$$

в соответствии с чем для i-й компоненты вектора состояния имеем

$$\left(x_i^{(t)} = 0,\ t = 1, \ldots, k\right) \ \text{ и } \ \left(x_i^{(k+1)} \neq 0\right)$$

тогда и только тогда, когда $i \in \mathcal{F}^{(k+1)}$ для комбинаторного BFS.

Доказательство. Индукция по количеству k выполненных итераций.

Пусть $k = 1$. Чтобы получить значения $x_i^{(1)}$ из уравнений (6) для $(s,i) \in E$, в них подставляется значение $x_s^{(0)} = d$, в результате получаем $x_i^{(1)} = -d^2 \neq 0$. Если i-е уравнение в (6) не содержит $x_s^{(k)}$ в правой части, то есть если $a_{is} = 0$ и $(s,i) \notin E$, то $x_i^{(1)} = 0$. Таким образом, для первой итерации лемма 1 верна.

Предположим, что лемма 1 верна для k-й итерации, т. е. для $t < k$ и $i \in \mathcal{F}^{(k)}$ выполняется $x_i^{(t)} = 0$ и

$$x_i^{(k)} = \sum_{c \in C(s,i)} (-1)^{\ell(c)} d^{\ell(c)+1} \neq 0.$$

Покажем, что лемма 1 верна и для $(k+1)$-й итерации. Для $i \neq s$ имеем

$$x_i^{(k+1)} = \Big(\sum_{j=1}^{n} a_{ij} x_j^{(k)}\Big)(-d) = \left(\sum_{(i,j)\in E} \Big(\sum_{c \in C(s,j)} (-1)^{\ell(c)} d^{\ell(c)+1}\Big)\right)(-d) =$$

$$= \left(\sum_{(i,j)\in E} \Big(\sum_{c \in C(s,j)} (-1)^{\ell(c)+1} d^{\ell(c)+2}\Big)\right) = \sum_{c \in C(s,i)} (-1)^{\ell(c)+1} d^{\ell(c)+2} \neq 0.$$

Таким образом, условие (5) выполнено для $i \in \mathcal{F}^{(k+1)}$. Покажем, что $x_i^{(k+1)} = 0$, если $i \notin \mathcal{F}^{(k+1)}$ и вершина i не была посещена на предыдущих k итерациях комбинаторного BFS. Это значит, что среди вершин j, смежных i, нет таких, для которых в ходе итераций (6) были пройдены простые цепи длины k, соединяющие s и j в ходе предыдущих k итераций. Поскольку лемма 1 верна для k, для всех таких вершин j имеем $x_j^{(k)} = 0$. Поэтому

$$x_i^{(k+1)} = \Big(\sum_{i=1}^{n} a_{ij} x_j^{(k)}\Big)(-d) = \Big(\sum_{(i,j)\in E}^{n} x_j^{(k)}\Big)(-d) = 0 \cdot (-d) = 0.$$

Лемма 1 доказана. ■

Таким образом, по (5) и лемме 1, итерации (6) дают те же самые фронтиры $\mathcal{F}^{(k)}$, $k = 1, 2, \ldots$, что и итерации комбинаторного BFS. Это значит, что мы получаем тот же самый обход графа. Отсюда следует

Теорема 1. Для графа G и заданной стартовой вершины s комбинаторный BFS и итерации метода Якоби с правой частью $b = e_s$ и $x^{(0)} = d \cdot e_s$ дают один и тот же обход.

6. Итерации метода Гаусса — Зейделя как реализация комбинаторного CCS

6.1. Вычисление компонент вектора состояния с помощью обходов цепей, исходящих из вершин текущего фронтира

Рассматривая $(k+1)$-ю итерацию комбинаторного CCS и итерацию вида (7) с тем же номером, будем использовать следующие обозначения. В отличие от п. 5, через $C(\tilde{s}, i)$ обозначим множество простых цепей, соединяющих вершины $\tilde{s} \in \mathcal{F}^{(k)}$ с вершиной $i \in \mathcal{F}^{(k+1)}$. На $(k+1)$-й итерации комбинаторного CCS производятся переходы по рёбрам этих цепей. Пусть $c + (j, i)$ обозначает цепь, получаемую из c соединением её последней вершины j с вершиной i ребром $(j, i) \in E$.

Все вычисления на итерациях (6) и (7) могут быть представлены как действия, производимые в соответствии с обходами отдельных цепей (не обязательно простых), соединяющих стартовую вершину и вершину i, для которой вычисляется значение $x_i^{(k+1)}$. Будем рассматривать только простые цепи, поскольку этого достаточно для доказательств леммы и теоремы, сформулированных далее. В действительности при реализации вычислений вида (6) или (7) множество всех арифметических операций, выполняемых на одной итерации, состоит из вычислений, соответствующих всем маршрутам, соединяющим стартовую вершину с другими вершинами в графе. Мы иллюстрируем это далее примером на рис. 1, *а* и *б*.

Цепи, которые обходятся в ходе одной итерации комбинаторного CCS, то есть цепи $c = (i_0, i_1, \ldots, i_{\ell(c)}) \in C(\tilde{s}, i)$, где $i_0 = \tilde{s}$, могут быть двух типов. Цепь $c \in C(\tilde{s}, i)$ — цепь типа (I), если $i_0 < i_1$, и типа (II), если $i_0 > i_1$. Для цепей обоих типов, возможно, за исключением первого ребра (i_0, i_1), имеем $i_{j-1} < i_j$ для $j = 1, \ldots, \ell(c)$. Переход по ребру (i_0, i_1) производится на шаге 3 комбинаторного CCS, переходы по последующим рёбрам цепи — на шаге 4.

Для $\tilde{s}, i \in V$ пусть $\ell(\tilde{s}, i) = \max\{\ell(c) : c \in C(\tilde{s}, i)\}$ — максимальная длина простой цепи, исходящей из $\tilde{s} \in \mathcal{F}^{(k)}$, в результате переходов по рёбрам которой достигаеся вершина $i \in \mathcal{F}^{(k+1)}$ на $(k+1)$-й итерации.

Пусть $i_0 = \tilde{s}$; $i_{\ell(c)} = i$ — первая и последняя вершины в цепи $c = (i_0, \ldots, i_{\ell(c)})$; $v_{\tilde{s}}$ — значение, которое в ходе одной итерации передаётся по цепи c, соединяющей $\tilde{s}$ и вершину i, при подстановке $v_{\tilde{s}}$ в уравнения (7) в соответствии с номерами вершин из c. Вычисление этого значения производится с помощью алгоритма 5 (обхода цепи).

Алгоритм 5. Алгоритм обхода цепи

Вход: цепь c, $v_{\tilde{s}}$.
Выход: $x_{i,c}^{(k+1)}$.

1: $c' := \varnothing$; $\quad x_{i_0,c'}^{(k+1)} := v_{\tilde{s}}$.
2: **Для** $t = 1, \ldots, \ell(c)$:
3: $\quad c'' := c' + (i_{t-1}, i_t)$; $\quad x_{i_t,c''}^{(k+1)} := x_{i_{t-1},c'}^{(k+1)}(-d)$; $\quad c' := c''$.
4: **Вернуть** $x_{i_t,c'}^{(k+1)}$.

В результате передачи $v_{\tilde{s}}$ при обходе цепи c мы получаем вклад $x_{i,c}^{(k+1)}$ в значение $x_i^{(k+1)}$, который передаётся по этой цепи. Для того чтобы выполнялось $x_i^{(k+1)} \neq 0$, должны иметься цепи, по которым в вершину i передаётся ненулевое значение $v_{\tilde{s}}$.

Таким образом, для $\tilde{s} \in \mathcal{F}^{(k)}$ и $c \in C(\tilde{s}, i)$ мы определяем $x_{i,c}^{(k+1)}$ как вклад значения $v_{\tilde{s}}$, которое передаётся по цепи $c \in C(\tilde{s}, i)$ из $\tilde{s}$ в i:

$$x_{i,c}^{(k+1)} = v_{\tilde{s}}(-d)^{\ell(c)}, \tag{8}$$

где

$$v_{\tilde{s}} = \begin{cases} x_{\tilde{s}}^{(k+1)}, & \text{если } c \text{ — цепь типа (I)}, \\ x_{\tilde{s}}^{(k)}, & \text{если } c \text{ — цепь типа (II)}. \end{cases}$$

Все вычисления на итерации (7) могут быть представлены как вычисления вкладов отдельных цепей, которые производятся по алгоритму 5.

Для $\tilde{s} \in \mathcal{F}^{(k)}$ определим $x_{i(\tilde{s})}^{(k+1)}$ как

$$x_{i(\tilde{s})}^{(k+1)} = \sum_{c \in C(\tilde{s},i)} x_{i,c}^{(k+1)}. \tag{9}$$

Если $C(\tilde{s}, i) = \varnothing$, то $x_{i(\tilde{s})}^{(k+1)} = 0$, поскольку в этом случае нет цепей, через которые вклад $v_{\tilde{s}}$ может быть в ходе $(k+1)$-й итерации (7) передан в вершину i из вершины $\tilde{s}$. Значение $x_{i(\tilde{s})}^{(k+1)}$ равно сумме всех вкладов в значение $x_i^{(k+1)}$, которые передаются через все цепи из $C(\tilde{s}, i)$ в ходе $(k+1)$-й итерации (7).

Отметим, что при проведении итераций (7) вершины из $\mathcal{F}^{(k+1)}$ могут достигаться не только в результате обхода простых цепей. Обход цепи типа (II) может привести к тому, что правильная цепь, исходящая из вершины j, уже достигнутой на итерации (7), на той же итерации будет повторно проходить через вершину $\tilde{s}$, из которой j была достигнута. Значение, полученное при обходе таких циклов, будет добавлено к исходному значению $x_{\tilde{s}}^{(k)}$ и передано далее через все цепи из $C(\tilde{s}, i)$. То есть оно будет включено в значение $v_{\tilde{s}}$ из (8) при повторном посещении $\tilde{s}$. Это может происходить, если $\tilde{s} > j$.

На рис. 1 показана такая ситуация. В данном случае (7) имеет следующий вид:

$$\begin{cases} x_1^{(k+1)} = x_2^{(k)}(-d), \\ x_2^{(k+1)} = \left(-1 + x_1^{(k+1)} + x_3^{(k)}\right)(-d), \\ x_3^{(k+1)} = x_2^{(k+1)}(-d). \end{cases}$$

Стартовая вершина — вершина 2. На первой итерации $x_1^{(0)} = 0$, $x_2^{(0)} = d$, $x_3^{(0)} = 0$, поэтому

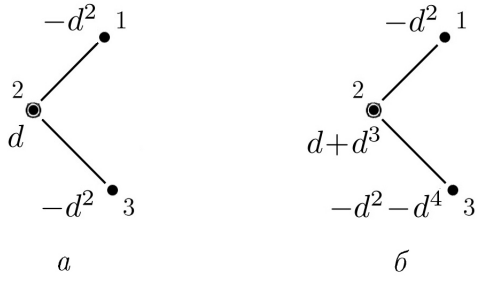
$$x_3^{(1)} = x_2^{(1)}(-d) = \left(-1 + x_1^{(1)}\right) d^2 = \left(-1 + x_2^{(0)}(-d)\right) d^2 = (-1 - d^2)d^2 = -d^2 - d^4.$$

Переданное по циклу $(2, 1, 2)$ значение $v_s = d$ вносит вклад $-d^4$ в значение $x_3^{(1)}$ при обходе всего маршрута $c_1 = (2, 1, 2, 3)$ на первой итерации. Суммируя его с вкладом, который передаётся по простой цепи $c_2 = (2, 3)$, состоящей из одного ребра, получаем итоговое значение $x_3^{(1)}$ на первой итерации как сумму вкладов, передаваемых из вершины 2 в вершину 3 по маршруту c_1 и по простой цепи c_2:

$$x_3^{(1)} = x_{3,c_1}^{(1)} + x_{3,c_2}^{(1)} = -d^4 - d^2.$$

Для графа на рис. 2 такие петли (циклы) отсутствуют в случае стартовой вершины 1. На рис. 2 также показано, как складываются вклады, передаваемые в вершину 5 по двум правильным цепям, которые обходятся в ходе одной итерации CSS. Это цепи $c_1 = (1, 2, 5)$ и $c_2 = (1, 3, 4, 5)$;

$$x_5^{(1)} = x_{5,c_1}^{(1)} + x_{5,c_2}^{(1)} = d^3 - d^4.$$



а *б*

Рис. 1. Получение значения $x_3^{(1)}$ для цепи на трёх вершинах в ходе одной итерации CCS

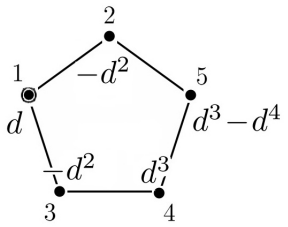


Рис. 2. Получение значения $x_5^{(1)}$ для цикла на пяти вершинах в ходе одной итерации CCS

6.2. Итерации метода Гаусса — Зейделя как реализация комбинаторного CCS

Для $i \neq s$ из (7) имеем

$$x_i^{(k+1)} = \Big(\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\Big)(-d) = \sum_{\substack{j<i \\ (i,j)\in E}} x_j^{(k+1)}(-d) + \sum_{\substack{j>i \\ (i,j)\in E}} x_j^{(k)}(-d). \quad (10)$$

Доказываемая далее лемма 2 утверждает, что получение на $(k+1)$-й итерации (7) вектора $x^{(k+1)}$ эквивалентно его получению по простым цепям из $C(\tilde{s},i)$ (9) для всех $\tilde{s} \in \mathcal{F}^{(k)}$.

Отметим, что мы не рассматриваем ситуацию наличия циклов, начинающихся и заканчивающихся в вершине $\tilde{s}$, пример которой приведён выше. Поскольку распространяющийся далее после прохождения такой петли (цикла) вклад будет одинаково распространяться по всем исходящим из вершины правильным простым цепям после повторного прохождения вершины, это не поменяет равенства или неравенства нулю суммы вкладов, переданных по цепям в ходе итерации (7).

Лемма 2. Пусть $i \in \mathcal{F}^{(k+1)}$ для комбинаторного CCS и $x_i^{(k+1)}$ — значение, вычисляемое на $(k+1)$-й итерации (7). Тогда

$$x_i^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} x_{i(\tilde{s})}^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} \sum_{c\in C(\tilde{s},i)} x_{i,c}^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} \sum_{c\in C(\tilde{s},i)} v_{\tilde{s}}(-d)^{\ell(c)},$$

где $v_{\tilde{s}} = x_{\tilde{s}}^{(k)}$ — передаваемое на $(k+1)$-й итерации из $\tilde{s}$ значение.

Доказательство. Индукция по количеству итераций комбинаторного CCS и итераций (7). Доказательство основания индукции и индуктивного предположения проведём индукцией по длине цепей из $C(\tilde{s},i)$ для $\tilde{s} \in \mathcal{F}^{(k)}$ и $i \in \mathcal{F}^{(k+1)}$.

Покажем, что лемма 2 верна для первой итерации. На этой итерации фронтир содержит только стартовую вершину s: $\mathcal{F}^{(0)} = \{s\}$. Пусть $i \in \mathcal{F}^{(1)}$ — такая вершина, что $\ell(s,i) = 1$. В этом случае $C(s,i)$ состоит из единственной цепи $c = (s,i)$ длины 1, содержащей только одно ребро $(s,i) \in E$. Таким образом, по (8) получаем $x_{i,c}^{(1)} = -d^2$, поскольку $v_s = d$ на первой итерации (7).

На первой итерации (7) имеем $x_j^{(0)} = 0$ для всех $j > i$, $(j,i) \in E$, $j \neq s$. Кроме того, $x_j^{(1)} = 0$ и для $j < i$, $(j,i) \in E$, $j \neq s$. Допустим, это не так и $x_j^{(1)} \neq 0$ для такого j. Значит, существует цепь типа (I) или типа (II), соединяющая вершины s и j, такая, что при подстановках на первой итерации значения $x_s^{(0)} = d$ в уравнения (7) с номерами, равными номерам вершин из этой цепи, мы получили бы $x_j^{(1)} \neq 0$. Но в этом случае $\ell(s,i) > 1$, поскольку $(j,i) \in E$, что противоречит предположению $\ell(s,i) = 1$.

Следовательно, все слагаемые в (10) нулевые, кроме $x_s^{(0)}$, если $i < s$, или кроме $x_s^{(1)}$, если $i > s$. Поэтому имеем

$$x_i^{(1)} = x_{i(s)}^{(1)} = x_{i,c}^{(1)} = -d^2,$$

где $c = (s,i)$. Таким образом, лемма 2 верна для вершины i, достигаемой на первой итерации (7) по цепям длины 1.

Предположим, что лемма 2 верна на первой итерации (7) для всех $i \in \mathcal{F}^{(1)}$, таких, что $\ell(s,i) = l$. Покажем, что она верна и для всех $i \in \mathcal{F}^{(1)}$, таких, что $\ell(s,i) = l+1$.

Поскольку на первой итерации $x_j^{(0)} = 0$ для $j > i$, $j \neq s$, то из (10) имеем

$$x_i^{(1)} = \sum_{\substack{j<i,\, j\neq s,\\ (i,j)\in E}} x_j^{(1)}(-d) + \alpha v_s(-d), \tag{11}$$

где $\alpha = 1$, если $(s,i) \in E$, $s > i$, и в этом случае $C(s,j) = \{(s,i)\}$. В случае $\alpha = 0$, $j \neq s$ выполняется $\ell(c) \leqslant \ell(s,j) \leqslant l$ для всех $c \in C(s,j)$, поскольку $\ell(s,i) = l+1$. Следовательно, по индукционному предположению для l имеем

$$x_j^{(1)} = \sum_{c' \in C(s,j)} x_{j,c'}^{(1)}. \tag{12}$$

Поскольку $x_{i,c}^{(1)} = x_{j,c'}^{(1)}(-d)$ для $c' \in C(s,j)$, такой, что $c = c' + (j,i)$, $c \in C(s,i)$, то, подставляя (12) в (11) и имея $v_s(-d) = x_{i,c}^{(1)}$ для $c = (s,i)$, получаем

$$x_i^{(1)} = \sum_{c \in C(s,i)} x_{i,c}^{(1)} = x_{i(s)}^{(1)}$$

для обоих возможных значений α.

Таким образом, для первой итерации лемма 2 верна для вершин $i \in \mathcal{F}^{(1)}$, достигаемых по цепям из $C(\tilde{s},i)$ любой длины.

Предположим, лемма 2 верна для всех итераций с номерами от 1 до k. Покажем, что она верна и для $(k+1)$-й итерации.

Пусть вершина $i \in \mathcal{F}^{(k+1)}$ такова, что $\ell(\tilde{s},i) = 1$ для вершин $\tilde{s} \in \mathcal{F}^{(k)}$. В этом случае все цепи $c \in C(\tilde{s},i)$ — это рёбра $(\tilde{s},i) \in E$, каждое из которых в (10) соответствует ненулевому слагаемому вида $x_{i,c}^{(k+1)} = x_{\tilde{s}}^{(k+1)}(-d)$, если $\tilde{s} < i$, или $x_{i,c}^{(k+1)} = x_{\tilde{s}}^{(k)}(-d)$, если $\tilde{s} > i$. Поэтому имеем

$$x_i^{(k+1)} = \sum_{\tilde{s} \in \mathcal{F}^{(k)}} \sum_{c \in C(\tilde{s},i)} x_{i,c}^{(k+1)} = \sum_{\tilde{s} \in \mathcal{F}^{(k)}} x_{i(\tilde{s})}^{(k+1)}.$$

Таким образом, лемма 2 верна для $i \in \mathcal{F}^{(k+1)}$, таких, что $\ell(\tilde{s}, i) = 1$ для $\tilde{s} \in \mathcal{F}^{(k)}$.

Предположим, что лемма 2 верна для $i \in \mathcal{F}^{(k+1)}$, таких, что $\ell(\tilde{s}, i) \leqslant l$, $\tilde{s} \in \mathcal{F}^{(k)}$. Покажем, что она верна для всех $i \in \mathcal{F}^{(k+1)}$, таких, что $\ell(\tilde{s}, i) = l + 1$ (индукция по l).

Пусть $\mathcal{F}_1^{(k)} = \{\tilde{s} \in \mathcal{F}^{(k)} : \tilde{s} < i\}$, $\mathcal{F}_2^{(k)} = \{\tilde{s} \in \mathcal{F}^{(k)} : \tilde{s} > i\}$, $\mathcal{F}_1^{(k)} \cup \mathcal{F}_2^{(k)} = \mathcal{F}^{(k)}$. Рассмотрим первое слагаемое S_1 в правой части (10):

$$S_1 = \sum_{\substack{j<i,\\(i,j)\in E}} x_j^{(k+1)}(-d).$$

Если $j < i$ и $x_j^{(k+1)} \neq 0$, то $C(\tilde{s}, j) \neq \varnothing$, поскольку по предположению индукции по k ненулевыми могут быть только те компоненты вектора состояния, которые соответствуют вершинам, достигнутым на итерациях до k-й включительно. Для получения ненулевого значения $x_j^{(k+1)}$ из правой части (10) требуются простые цепи, соединяющие такие вершины и j. Длина любой цепи из $C(\tilde{s}, j)$ меньше или равна l, поскольку иначе $\ell(\tilde{s}, i) > l + 1$. Поэтому по индукционному предположению по l имеем

$$S_1 = \sum_{\tilde{s}\in\mathcal{F}_1^{(k)}} \sum_{c\in C(\tilde{s},j)} x_{j,c}^{(k+1)}(-d).$$

Поскольку по (8) верно $x_{j,c}^{(k+1)}(-d) = x_{i,c'}^{(k+1)}$, где $c' = c + (j, i)$, с учётом (9) получаем

$$S_1 = \sum_{\tilde{s}\in\mathcal{F}_1^{(k)}} \sum_{c\in C(\tilde{s},i)} x_{i,c}^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}_1^{(k)}} x_{i(\tilde{s})}^{(k+1)}. \tag{13}$$

Рассмотрим второе слагаемое S_2 из правой части (10):

$$S_2 = \sum_{\substack{j>i,\\(i,j)\in E}} x_j^{(k)}(-d).$$

По индукционному предположению по k для $j > i$ имеем $x_j^{(k)} \neq 0$, если $j \in \mathcal{F}_2^{(k)}$. Поэтому

$$S_2 = \sum_{\tilde{s}\in\mathcal{F}_2^{(k)}} x_{i(\tilde{s})}^{(k+1)}, \tag{14}$$

поскольку $x_{\tilde{s}}^{(k)}(-d) = x_{i,c}^{(k+1)} = x_{i(\tilde{s})}^{(k+1)}$ для цепей $c = (\tilde{s}, i)$, $\tilde{s} \in \mathcal{F}_2^{(k)}$, $\ell(c) = 1$.

Складывая S_1 и S_2, из (13) и (14) получаем

$$x_i^{(k+1)} = S_1 + S_2 = \sum_{\tilde{s}\in\mathcal{F}_1^{(k)}} x_{i(\tilde{s})}^{(k+1)} + \sum_{\tilde{s}\in\mathcal{F}_2^{(k)}} x_{i(\tilde{s})}^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} x_{i(\tilde{s})}^{(k+1)}.$$

Значит, лемма 2 верна для всех $i \in \mathcal{F}^{(k+1)}$, таких, что $\ell(\tilde{s}, i) = l + 1$. Таким образом, по индукции по l утверждение леммы 2 верно для всех вершин из $\mathcal{F}^{(k+1)}$, т. е. лемма верна для $(k + 1)$-й итерации, и по индукции по k она верна для всех k ($k \leqslant n$). ■

Теорема 2. Для графа G и стартовой вершины s комбинаторный CCS и итерации метода Гаусса — Зейделя с правой частью $b = e_s$ и $x^{(0)} = d \cdot e_s$ дают один и тот же обход.

Доказательство. Необходимо показать, что в ходе выполнения итераций (7) условие (5) для вершины $i \in V$ выполнено тогда и только тогда, когда $i \in \mathcal{F}^{(k+1)}$ для комбинаторного CCS.

По лемме 2

$$x_i^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} \sum_{c\in C(\tilde{s},i)} v_{\tilde{s}}(-d)^{\ell(c)},$$

где $v_{\tilde{s}} = O(d^{\ell(s,\tilde{s})}) \neq 0$ и $\ell(s,\tilde{s})$ — максимальная длина простой цепи, соединяющей стартовую вершину s и вершину $\tilde{s} \in \mathcal{F}^{(k)}$. По (8), при переходе по каждому ребру такой цепи значение $v_{\tilde{s}}$, которое является суммой степеней d с некоторыми коэффициентами перед ними, умножается на $-d$. В этом случае одни и те же степени d в итоговом значении $x_{i,c}^{(k+1)}$ имеют один и тот же знак, поскольку они получаются в результате одного и того же количества умножений на $-d$, производимых в соответствии с алгоритмом 5 для цепей одной длины. Поэтому если $C(\tilde{s},i) \neq \varnothing$, то $x_i^{(k+1)} \neq 0$ и условие (5) выполнено для $i \in \mathcal{F}^{(k+1)}$ комбинаторного CCS после $(k+1)$-й итерации.

Пусть $i \in V \setminus \mathcal{C}^{(k+1)}$ для комбинаторного CCS, т. е. вершина i не достигается после $(k+1)$-й итерации. Это значит, что $C(\tilde{s},i) = \varnothing$ для всех $\tilde{s} \in \mathcal{F}^{(k)}$. Поскольку $x_{i(\tilde{s})}^{(k+1)} = 0$, если $C(\tilde{s},i) = \varnothing$, то по лемме 2 из этого следует, что

$$x_i^{(k+1)} = \sum_{\tilde{s}\in\mathcal{F}^{(k)}} x_{i(\tilde{s})}^{(k+1)} = 0,$$

т. е. условие (5) не выполнено для $i \in V \setminus \mathcal{C}^{(k+1)}$.

Таким образом, каждая итерация комбинаторного CCS даёт те же фронтиры $\mathcal{F}^{(k)}$, $k = 1, 2, \ldots$, что и итерации (7). Значит, мы получим те же самые обходы графа при их выполнении. ■

7. Примеры обходов графов, реализуемых итерациями методов Якоби и Гаусса — Зейделя

В рассматриваемых примерах стартовая вершина — это вершина 1; $d = 2$.

Сравним обходы графов, реализуемые BFS и CCS, для графа G_1 (рис. 3). Для этого графа преобразование $\mathbf{F}$ векторов состояния $x^{(k)}$ вида (6) задаётся следующими уравнениями:

$$\begin{cases} x_1^{(k+1)} = \left(-1 + x_2^{(k)}\right)(-d), \\ x_2^{(k+1)} = \left(x_1^{(k)} + x_3^{(k)} + x_6^{(k)}\right)(-d), \\ x_3^{(k+1)} = \left(x_2^{(k)} + x_4^{(k)} + x_7^{(k)}\right)(-d), \\ x_4^{(k+1)} = \left(x_3^{(k)}\right)(-d), \\ x_5^{(k+1)} = \left(x_6^{(k)}\right)(-d), \\ x_6^{(k+1)} = \left(x_2^{(k)} + x_5^{(k)} + x_7^{(k)}\right)(-d), \\ x_7^{(k+1)} = \left(x_3^{(k)} + x_6^{(k)} + x_8^{(k)}\right)(-d), \\ x_8^{(k+1)} = \left(x_7^{(k)}\right)(-d). \end{cases} \tag{15}$$

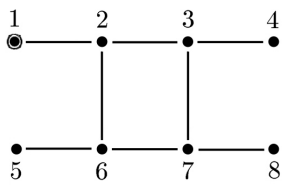


Рис. 3. Граф G_1

Обход, который эти итерации дадут для G_1, представлен на рис. 4. Требуется четыре итерации для того, чтобы завершить обход графа, используя алгебраический BFS, реализуемый итерациями метода Якоби (6).

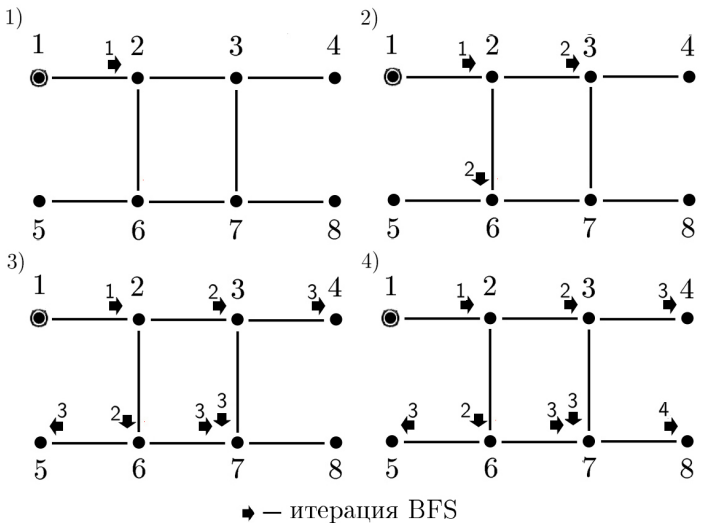


Рис. 4. Обход графа G_1, реализуемый BFS

Для итераций алгебраического CCS преобразование $\mathbf{F}$ векторов состояния $x^{(k)}$ задаётся следующими уравнениями:

$$\begin{cases} x_1^{(k+1)} = \left(-1 + x_2^{(k)}\right)(-d), \\ x_2^{(k+1)} = \left(x_1^{(k+1)} + x_3^{(k)} + x_6^{(k)}\right)(-d), \\ x_3^{(k+1)} = \left(x_2^{(k+1)} + x_4^{(k)} + x_7^{(k)}\right)(-d), \\ x_4^{(k+1)} = \left(x_3^{(k+1)}\right)(-d), \\ x_5^{(k+1)} = \left(x_6^{(k)}\right)(-d), \\ x_6^{(k+1)} = \left(x_2^{(k+1)} + x_5^{(k+1)} + x_7^{(k)}\right)(-d), \\ x_7^{(k+1)} = \left(x_3^{(k+1)} + x_6^{(k+1)} + x_8^{(k)}\right)(-d), \\ x_8^{(k+1)} = \left(x_7^{(k+1)}\right)(-d). \end{cases} \tag{16}$$

Производя итерации (16), получаем обход графа за две итерации. Как можно видеть (рис. 5), в отличие от BFS, на первой итерации CCS производятся переходы по рёбрам правильных цепей, исходящих из вершины 2, которая достигается из вершины 1 на той же итерации.

Векторы состояния $x^{(k)}$ и множества $\mathcal{C}^{(k)}$ вершин, достигаемых после k-й итерации BFS и CCS, представлены далее.

BFS (реализумый итерациями (15))

Инициализация: $x^{(0)} = (2, 0, 0, 0, 0, 0, 0, 0)$, $\mathcal{F}^{(0)} = \{1\}$, $\mathcal{C}^{(0)} = \{1\}$;
$x^{(1)} = (2, -4, 0, 0, 0, 0, 0, 0)$, $\mathcal{F}^{(1)} = \{2\}$, $\mathcal{C}^{(1)} = \{1, 2\}$;
$x^{(2)} = (10, -4, 8, 0, 0, 8, 0, 0)$, $\mathcal{F}^{(2)} = \{3, 6\}$, $\mathcal{C}^{(2)} = \{1, 2, 3, 6\}$;
$x^{(3)} = (10, -52, 8, -16, -16, 8, -32, 0)$, $\mathcal{F}^{(3)} = \{5, 7\}$, $\mathcal{C}^{(3)} = \{1, 2, 3, 4, 5, 6, 7\}$;
$x^{(4)} = (106, -52, 200, -16, -16, 200, -32, 64)$, $\mathcal{F}^{(4)} = \{8\}$, $\mathcal{C}^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

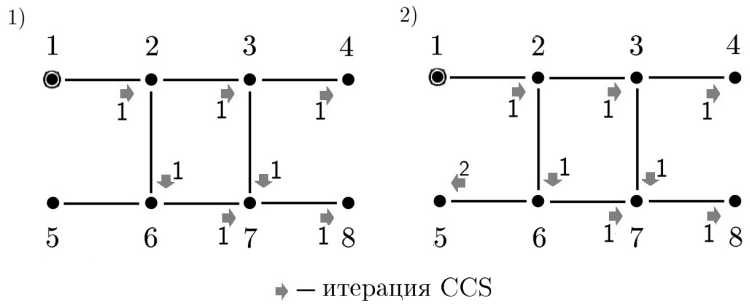


Рис. 5. Обход графа G_1, реализуемый CCS

CCS (реализуемый итерациями (16))

Инициализация: $x^{(0)} = (2,0,0,0,0,0,0,0)$, $\mathcal{F}^{(0)} = \{1\}$, $\mathcal{C}^{(0)} = \{1\}$;
$x^{(1)} = (2,-4,8,-16,0,8,-32,64)$, $\mathcal{F}^{(1)} = \{2,3,4,6,7,8\}$, $\mathcal{C}^{(1)} = \{1,2,3,4,6,7,8\}$;
$x^{(2)}$=$(10,-52,200,-400,-16,200,-928,1856)$, $\mathcal{F}^{(2)}$=$\{5\}$, $\mathcal{C}^{(2)}$=$\{1,2,3,4,5,6,7,8\}$.

Для графа G_2 (рис. 6) преобразование $\mathbf{F}$ вида (6) вектора состояния $x^{(k)}$ задаётся уравнениями

$$\begin{cases} x_1^{(k+1)} = \left(-1 + x_2^{(k)}\right)(-d), \\ x_2^{(k+1)} = \left(x_1^{(k)} + x_3^{(k)}\right)(-d), \\ x_3^{(k+1)} = \left(x_2^{(k)} + x_4^{(k)}\right)(-d), \\ x_4^{(k+1)} = \left(x_3^{(k)} + x_5^{(k)}\right)(-d), \\ x_5^{(k+1)} = \left(x_4^{(k)}\right)(-d). \end{cases} \tag{17}$$

Преобразование $\mathbf{F}$, которое производится CCS для этого графа, имеет вид

$$\begin{cases} x_1^{(k+1)} = \left(-1 + x_2^{(k)}\right)(-d), \\ x_2^{(k+1)} = \left(x_1^{(k+1)} + x_3^{(k)}\right)(-d), \\ x_3^{(k+1)} = \left(x_2^{(k+1)} + x_4^{(k)}\right)(-d), \\ x_4^{(k+1)} = \left(x_3^{(k+1)} + x_5^{(k)}\right)(-d), \\ x_5^{(k+1)} = \left(x_4^{(k+1)}\right)(-d). \end{cases} \tag{18}$$

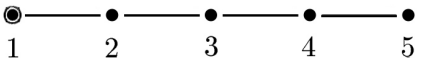


Рис. 6. Граф G_2 — правильная цепь

В графе G_2 имеется правильная цепь, исходящая из вершины 2, достигаемой на первой итерации, которой принадлежат все вершины графа, за исключением стартовой. Поэтому все вершины графа будут посещены в ходе одной итерации CCS (рис. 7). BFS для этого потребуется четыре итерации, что составляет максимально возможное количество итераций для всех возможных вариантов стартовой вершины.

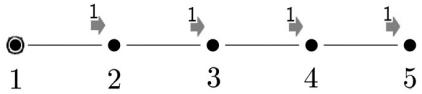


Рис. 7. Обход графа G_2, реализуемый CCS

BFS (реализуемый итерациями (17))

Инициализация: $x^{(0)} = (2, 0, 0, 0, 0)$, $\mathcal{F}^{(0)} = \{1\}$, $\mathcal{C}^{(0)} = \{1\}$;
$x^{(1)} = (2, -4, 0, 0, 0)$, $\mathcal{F}^{(1)} = \{2\}$, $\mathcal{C}^{(1)} = \{1, 2\}$;
$x^{(2)} = (10, -4, 8, 0, 0)$, $\mathcal{F}^{(2)} = \{3\}$, $\mathcal{C}^{(2)} = \{1, 2, 3\}$;
$x^{(3)} = (10, -36, 8, -16, 0)$, $\mathcal{F}^{(3)} = \{4\}$, $\mathcal{C}^{(3)} = \{1, 2, 3, 4\}$;
$x^{(4)} = (74, -36, 104, -16, 32)$, $\mathcal{F}^{(3)} = \{5\}$, $\mathcal{C}^{(4)} = \{1, 2, 3, 4, 5\}$.

CCS (реализуемый итерациями (18))

Инициализация: $x^{(0)} = (2, 0, 0, 0, 0)$, $\mathcal{F}^{(0)} = \{1\}$, $\mathcal{C}^{(0)} = \{1\}$;
$x^{(1)} = (2, -4, 8, -16, 32)$, $\mathcal{F}^{(1)} = \{2, 3, 4, 5\}$, $\mathcal{C}^{(1)} = \{1, 2, 3, 4, 5\}$.

При другой нумерации вершин (рис. 8) на каждой итерации CCS нет правильных цепей, исходящих из достигнутых на этих итерациях вершин. Поэтому для того, чтобы посетить все вершины графа, и BFS, и CSS требуется произвести четыре итерации; обход, который даёт CCS, полностью повторяет обход, который даёт BFS (рис. 9).

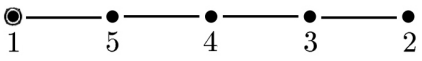


Рис. 8. Неправильная цепь

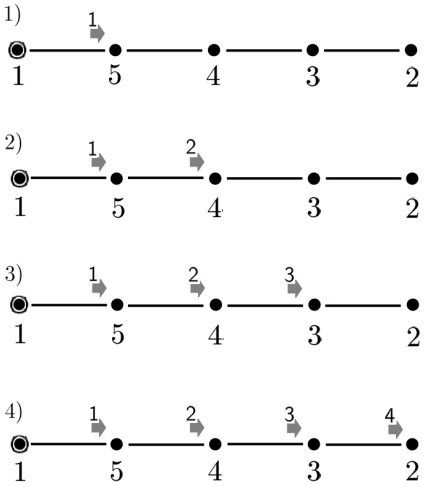


Рис. 9. Обход графа, реализуемый CCS

8. Беззнаковый CCS

В качестве преобразования $\mathbf{F}$ вектора состояния в алгоритме 3 обхода компоненты связности может быть использовано следующее преобразование:

$$x_i^{(k+1)} = \left(b_i + \sum_{l=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) d. \tag{19}$$

Назовём алгоритм 3, который использует (19) в качестве преобразования $\mathbf{F}$, *беззнаковым CCS*.

В ходе выполнения беззнакового CCS, который даёт тот же самый обход графа, что и CCS, вычисления производятся с использованием только неотрицательных целых чисел, если $d \in \mathbb{Z}$. Отсутствие операций деления и, возможно, умножения в случае $d = 1$ при численной реализации позволяет значительно уменьшить время, требуемое для обхода графа.

Преобразование (19) представляет собой модификацию преобразования $\mathbf{F}$, используемого CSS. Вместо преобразования $x^{(k+1)} = (b - Lx^{(k+1)} - Ux^{(k)})\, d$ используется преобразование $x^{(k+1)} = (b + Lx^{(k+1)} + Ux^{(k)})\, d$.

Если не учитывать умножения на d, итерация беззнакового CCS может быть рассмотрена как итерация BFS, в которой каждая компонента вектора состояния вычисляется с использованием компонент с меньшими индексами, уже вычисленными на текущей итерации. В то же время умножение (деление) на d и операция обращения по сложению может быть принципиальной для применения алгоритмов обхода графа при решении задач дискретного анализа и дискретной оптимизации. Например, реализация CCS для возмущённых матриц графов с нецелочисленными элементами может быть использована в подходе к решению задачи проверки изомофизма графов [11, 12].

9. Регуляризация вектора состояния и маскировка вершин

Использование умножения вместо деления в преобразовании (7) позволяет уменьшить время, требуемое для проведения обхода графа. Для того чтобы предотвратить как переполнение значений компонент, представляемых числами с ограниченной длиной мантиссы, так и их зануление в случае $d \in (0, 1)$, после заданного количества итераций M вектор состояния может быть регуляризован:

$$x_i^{(k)} \to \frac{1}{d^M} x_i^{(k)}.$$

Кроме того, без изменения порядка вычислительной сложности CCS существенное ускорение может быть получено при использовании маскировки вершин. *Маскировка вершины* означает исключение её из дальнейших вычислений. Маскируемые вершины — это вершины, из которых не могут быть достигнуты вершины, не достигнутые на предыдущих итерациях. Маскироваться могут как вершины, принадлежащие уже полученным компонентам связности, так и уже посещённые вершины компоненты связности, обход которой производится на текущей итерации алгоритма 4. Во втором случае при вычислениях на $(k + 1)$-й итерации (7) маскируются вершины, принадлежащие фронтирам, полученным на предыдущих $(k - 1)$ итерациях алгоритма 3.

Для экономии памяти и ускорения вычисления должны проводиться с *портретом* матрицы смежности. Портрет матрицы смежности содержит только ненулевые элементы матрицы, то есть он представляет собой список рёбер графа. При использовании портрета матрицы смежности одна итерация вида (6) или (7) имеет вычислительную сложность $O(m)$, поскольку для одной такой итерации требуется один проход по списку рёбер.

10. Обход графа и нумерация его вершин

Наличие правильных цепей в графе определяется нумерацией его вершин. Нумерация вершин и выбор стартовой вершины определяют вычислительную сложность проведения обхода графа с помощью CCS. Нумерация вершин графа — это обязательный параметр индивидуальной задачи обхода графа.

Предположим, что информация о графе, для которого необходимо провести обход, регулярно обновляется добавлением или удалением его вершин и рёбер. Оптимизация нумерации вершин может быть произведена, исходя из расстояния от них до вершины, которая выбирается как стартовая. При оптимальной нумерации вершин, когда все цепи, исходящие из стартовой вершины, правильные, CCS будет иметь минимально возможную для алгоритма обхода графа сложность $O(m)$.

Отметим, что назначение номеров вершин в соответствии с удалённостью представляемых ими объектов от некоторого заданного объекта часто является естественным подходом. Это имеет место, например, для транспортных сетей. Случайность в нумерации вершин неизбежна в таких приложениях, но она может быть минимизирована. Граф, в котором все вершины пронумерованы случайно, может быть интерпретирован как граф транспортной сети, хранящийся в базе данных со случайно пронумерованными записями, что может встретиться в таких приложениях только гипотетически при условии, что данные вводятся в течение долгого времени по мере развития сети. И даже при совершенно случайной нумерации вершин графа в большинстве случаев в таком графе присутствуют правильные цепи.

11. Количество итераций BFS и CCS, требуемых для обхода связного графа

Сравним количество итераций BFS и CCS, требуемых для обхода графа. Если не принимать во внимание такие детали реализации алгоритмов, как распараллеливание вычислений и маскировка вершин, то вычислительная сложность обхода графа определяется количеством итераций, требуемых для посещения всех его вершин.

Пусть N_{BFS} — количество итераций, требуемых для того, чтобы в ходе BFS посетить все вершины графа, и N_{CCS} — количество итераций, требуемых CCS для обхода того же графа из той же стартовой вершины. Значения N_{BFS} и N_{CCS} определяются выбором стартовой вершины, но, кроме того, N_{CCS} определяется также нумерацией вершин графа.

Поскольку и BFS, и CCS производят переходы через рёбра, инцидентные вершинам фронтиров, можно утверждать, что для любого графа $N_{\mathrm{CCS}} \leqslant N_{\mathrm{BFS}}$. Вместе с тем если в ходе выполнения CCS достигается хотя бы одна вершина, из которой исходит хотя бы одна правильная цепь, то достижимость вершин графа из стартовой вершины определяется CCS за меньшее количество итераций, чем требуется для этого BFS, то есть в этом случае $N_{\mathrm{CCS}} < N_{\mathrm{BFS}}$. Поэтому верно следующее утверждение.

Утверждение 1. Для одной и той же стартовой вершины CCS требует не больше итераций, чем BFS.

12. Вычислительный эксперимент

Цель эксперимента — оценить влияние диаметра графа на разницу в количестве проводимых в среднем итераций CCS и BFS. Для этого мы рассчитывали общее количество итераций, потребовавшееся для обходов случайных графов, и находили отношение этих значений для CCS и BFS для одних и тех же стартовых вершин. Случайные графы — это графы, в которых при их генерации:

1) случайно задаётся нумерация вершин;
2) случайно выбранные несмежные вершины соединяются ребром.

Как *расширенную звезду* определим граф, в котором одна вершина является общей для некоторого количества исходящих простых цепей. Назовём эти цепи *лучами* расширенной звезды. Чтобы оценить влияние диаметра на разницу в количестве про-

водимых в среднем итераций CCS и BFS, все графы генерировались на основе графов двух типов:

(I) расширенных звёзд с лучами разной (случайной) длины;
(II) расширенных звёзд с лучами одинаковой длины.

К сгенерированному графу (расширенной звезде со случайной нумерацией вершин) добавлялись рёбра, соединяющие несмежные до этого вершины. Пусть $E(G)$ — множество рёбер графа G. Алгоритм генерирации случайного графа следующий:

1) Генерируем на множестве вершин $V = \{1, \ldots, n\}$ расширенную звезду $G^{(0)}$ типа (I) или типа (II).
2) Добавляем случайные рёбра к $G^{(0)}$:

$$E(G^{(i+1)}) := E(G^{(i)}) \cup \{e_{i+1}\},$$

где $e_{i+1} \notin E(G^{(i)})$, $i = 1, \ldots, \tilde{m} - 1$; $\tilde{m}$ — количество добавляемых к $G^{(0)}$ случайных рёбер. Количество рёбер в сгенерированном графе $G = G^{(\tilde{m})}$ составляет

$$m = |E(G^{(\tilde{m})})| = |E(G^{(0)})| + \tilde{m} = n - 1 + \tilde{m}.$$

Параметры графов, используемых в эксперименте, следующие: n — количество вершин расширенной звёзды, на основе которой генерируется случайный граф; $\tilde{m}$ — количество добавляемых случайных рёбер, характеризующее плотность графа. Для графов, генерируемых на основе расширенных звёзд с лучами одинаковой длины, параметры следующие: ℓ — длина луча расширенной звёзды; r — количество лучей.

Заметим, что чем плотнее граф, то есть чем больше значение $\tilde{m}$, тем, как правило, меньше диаметр сгенерированного графа. При добавлении случайных рёбер уменьшение диаметра графа может происходить очень быстро, что можно оценить по резкому уменьшению количества итераций и CCS, и BFS при добавлении в расширенную звезду случайных рёбер, как показано далее в таблицах для значений $\tilde{m} = 0$ и $2n$. Кроме того, диаметр генерируемого графа типа (II), как правило, тем меньше, чем меньше длина ℓ луча расширенной звезды, из которой получен граф.

Пусть $N^{(i)}_{\mathrm{CCS}}$ и $N^{(i)}_{\mathrm{BFS}}$ — количества итераций, которые необходимо провести с помощью CCS и BFS соответственно для обхода i-го графа из набора, состоящего из M случайных графов с заданными параметрами. Пусть

$$\overline{N}_{\mathrm{CCS}} = \sum_{i=1}^{M} N^{(i)}_{\mathrm{CCS}}, \quad \overline{N}_{\mathrm{BFS}} = \sum_{i=1}^{M} N^{(i)}_{\mathrm{BFS}}.$$

В табл. 1–4 приведены результаты экспериментов, в ходе которых для вычисления $\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$ с помощью CCS и BFS производились обходы $M = 10\,000$ случайно сгенерированных графов типов (I) и (II) с $n = 101$ и указанными параметрами.

Таблица 1
Разреженные графы типа (I), $n = 101$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	$315\,990/619\,604 \approx 0{,}51$
$2n$	$44\,646/60\,148 \approx 0{,}56$
$5n$	$30\,288/52\,269 \approx 0{,}58$
$10n$	$25\,285/40\,431 \approx 0{,}63$

Таблица 2
Разреженные графы типа (II), $n = 101$, $\ell = 50$, $r = 2$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	$381\,702/752\,844 \approx 0{,}51$
$2n$	$24\,563/41\,462 \approx 0{,}59$
$5n$	$20\,007/30\,059 \approx 0{,}67$
$10n$	$19\,925/23\,346 \approx 0{,}85$

Таблица 3
Разреженные графы типа (II),
$n = 101$, $\ell = 20$, $r = 5$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	95 746/154 515 ≈ 0,62
$2n$	24 705/41 494 ≈ 0,59
$5n$	20 005/30 062 ≈ 0,67
$10n$	19 933/23 360 ≈ 0,85

Таблица 4
Разреженные графы типа (II),
$n = 101$, $\ell = 10$, $r = 10$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	170 693/304 627 ≈ 0,56
$2n$	24 637/41 540 ≈ 0,59
$5n$	20 005/30 052 ≈ 0,67
$10n$	19 939/23 429 ≈ 0,85

Для плотных графов с количеством рёбер $n+n^2/4$, $n = 101$, что составляет 60 % от количества рёбер в полном графе, имеем $\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}} = 16\,851/20\,000 \approx 0{,}84$. Усреднение получено по $M = 10\,000$ случайных графов.

Для вычисления каждого отношения $\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$, представленного в табл. 5–8, с помощью CCS и BFS производились обходы $M = 1\,000$ случайно сгенерированных графов указанных типов с указанными параметрами.

Таблица 5
Разреженные графы типа (I),
$n = 1001$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	310 023/618 989 ≈ 0,5
$2n$	3 140/6 006 ≈ 0,52
$5n$	2 102/4 046 ≈ 0,52
$10n$	2 000/3 189 ≈ 0,63

Таблица 6
Разреженные графы типа (II),
$n = 1001$, $\ell = 500$, $r = 2$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	376 913/752 406 ≈ 0,5
$2n$	3 119/6 001 ≈ 0,52
$5n$	2 076/4 045 ≈ 0,51
$10n$	2 000/3 185 ≈ 0,63

Таблица 7
Разреженные графы типа (II),
$n = 1001$, $\ell = 200$, $r = 5$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	154 000/298 341 ≈ 0,52
$2n$	3 114/6 011 ≈ 0,52
$5n$	2 079/4 041 ≈ 0,51
$10n$	2 000/3 220 ≈ 0,62

Таблица 8
Разреженные графы типа (II),
$n = 1001$, $\ell = 100$, $r = 10$

$\tilde{m}$	$\overline{N}_{\mathrm{CCS}}/\overline{N}_{\mathrm{BFS}}$
0	79 238/148 825 ≈ 0,53
$2n$	3 122/5 993 ≈ 0,52
$5n$	2 089/4 063 ≈ 0,51
$10n$	2 000/3 202 ≈ 0,62

Результаты вычислительного эксперимента показывают, что чем больше диаметр графа, тем больше в среднем разница в количестве итераций CCS и BFS, требуемых для обхода графа. Для разреженных графов эта разница в среднем составляет от 15 до 49 % процентов от количества итераций, требуемых для обхода BFS, при $n = 101$ (табл. 1–4) и от 37 до 50 % процентов при $n = 1001$ (табл. 5–8). Для плотных графов при $n = 101$ разница составляет в среднем 16 %.

Выводы

Рассмотрены методы простой итерации решения систем линейных алгебраических уравнений с модифицированными матрицами смежности графов и заданной правой частью как реализации обходов графа. Такой подход даёт два варианта алгоритмов обхода графа. Один из них реализуется итерациями метода Якоби, второй — итерациями метода Гаусса — Зейделя. Обход, реализуемый в ходе проведения итераций метода

Якоби, эквивалентен поиску в ширину, тогда как обход, реализуемый с помощью метода Гаусса — Зейделя, не эквивалентен ни поиску в ширину, ни поиску в глубину. Для любой индивидуальной задачи нахождения компонент связности графа количество итераций, требуемых для такого алгоритма, не превышает количества итераций, требуемых для поиска в ширину для одной и той же стартовой вершины в графе. Для многих индивидуальных задач нахождения компонент связности графа алгоритм обхода, ассоциированный с итерациями метода Гаусса — Зейделя, требует меньшего количества итераций.

ЛИТЕРАТУРА

1. *Cormen T. H., Leiserson C. E., Rivest R. L., and Stein C.* Introduction to Algorithms (3rd ed). MIT Press, 2009.
2. *Zuse K.* Der Plankalkül. Konrad Zuse Internet Archive. 1972. S. 96–105.
3. *Moore E. F.* The shortest path through a maze // Proc. Intern. Symp. Theory of Switching. P. II. Cambridge, MA: Harvard University Press, 1959. P. 285–292.
4. *Lee C. Y.* An algorithm for path connections and its applications // IRE Trans. Electronic Comput. 1961. V. EC-10. No. 3. P. 346–365.
5. *Beamer S., Asanović K., and Patterson D.* Direction-optimized breadth-first search // Proc. SC'12. Salt Lake City, Utah, 2012. Article 12. P. 1–10.
6. *Bücker H. M. and Sohr C.* Reformulating a breadth-first search algorithm on an undirected graph in the language of linear algebra // Proc. MCSI'14. IEEE Computer Society, 2014. P. 33–35.
7. *Azad A. and Buluç A.* A work-efficient parallel sparse matrix-sparse vector multiplication algorithm // Proc. IPDPS'17. Orlando, Florida, USA, 2017. P. 688–697.
8. *Yang C., Buluç A., and Owens J. D.* Graphblast: A high-performance linear algebra-based graph framework on the GPU // ACM Trans. Math. Software. 2022. V. 48 (1). P. 1–51.
9. *Yang C., Wang Y., and Owens J. D.* Fast sparse matrix and sparse vector multiplication algorithm on the GPU // Proc. IPDPSW'15. IEEE Computer Society, 2015. P. 841–847.
10. *Burkhardt P.* Optimal algebraic Breadth-First Search for sparse graphs // ACM Trans. Knowl. Discov. Data. 2021. V. 15 (5). Article 77.
11. *Пролубников А. В.* Сведение задачи проверки изоморфизма графов к задаче проверки равенства полиномов от n переменных // Тр. ИММ УрО РАН. 2016. Т. 22. № 1. С. 235–240.
12. *Пролубников А. В.* Точность и сложность вычислений, необходимые для проверки изоморфизма графов сравнением полиномов // Вычислительные технологии. 2016. Т. 21. № 6. С. 71–88.

REFERENCES

1. *Cormen T. H., Leiserson C. E., Rivest R. L., and Stein C.* Introduction to Algorithms (3rd ed). MIT Press, 2009.
2. *Zuse K.* Der Plankalkül. Konrad Zuse Internet Archive, 1972, S. 96–105. (in German)
3. *Moore E. F.* The shortest path through a maze. Proc. Intern. Symp. Theory of Switching. P. II, Cambridge, MA, Harvard University Press, 1959, pp. 285–292.
4. *Lee C. Y.* An algorithm for path connections and its applications. IRE Trans. Electronic Comput., 1961, vol. EC-10, no. 3, pp. 346–365.
5. *Beamer S., Asanović K., and Patterson D.* Direction-optimized breadth-first search. Proc. SC'12, Salt Lake City, Utah, 2012, article 12, pp. 1–10.

6. *Bücker H. M. and Sohr C.* Reformulating a breadth-first search algorithm on an undirected graph in the language of linear algebra. Proc. MCSI'14, IEEE Computer Society, 2014, pp. 33–35.
7. *Azad A. and Buluç A.* A work-efficient parallel sparse matrix-sparse vector multiplication algorithm. Proc. IPDPS'17, Orlando, Florida, USA, 2017, pp. 688–697.
8. *Yang C., Buluç A., and Owens J. D.* Graphblast: A high-performance linear algebra-based graph framework on the GPU. ACM Trans. Math. Software, 2022, vol. 48 (1), pp. 1–51.
9. *Yang C., Wang Y., and Owens J. D.* Fast sparse matrix and sparse vector multiplication algorithm on the GPU. Proc. IPDPSW'15, IEEE Computer Society, 2015, pp. 841–847.
10. *Burkhardt P.* Optimal algebraic Breadth-First Search for sparse graphs // ACM Trans. Knowl. Discov. Data. 2021. V. 15 (5). Article 77.
11. *Prolubnikov A. V.* Svedenie zadachi proverki izomorfizma grafov k zadache proverki ravenstva polinomov ot n peremennykh [Reduction of the graph isomorphism problem to checking the equality of polynomials of n variables]. Trudy Instituta Matematiki i Mekhaniki UrO RAN, 2016, vol. 22, no. 1, pp. 235–240. (in Russian)
12. *Prolubnikov A. V.* Tochnost' i slozhnost' vychisleniy, neobkhodimye dlya proverki izomorfizma grafov sravneniem polinomov [Accuracy and complexity of computations needed to check graph isomorphism checking polynomials]. Vychislitel'nye Tekhnologii, 2016, vol. 21, no. 6, pp. 71–88. (in Russian)

УДК 519.7 DOI 10.17223/20710410/68/6

ROLE COLORING OF GRAPHS FROM ROOTED PRODUCTS

M. Komathi, P. Ragukumar

School of Advanced Sciences, Vellore Institute of Technology, Vellore, India

E-mail: komathirk2108@gmail.com, ragukumar2003@gmail.com

A k-role coloring is an assignment of k colors to the vertices of a graph such that if any two vertices receive the same color, then the set of colors assigned to their neighborhood will also be the same. Any graph with n vertices can have n-role coloring. Although it is easy to determine whether a graph with n vertices accepts a 1-role coloring, the challenge of k-role coloring is known to be difficult for $k \geqslant 2$. In fact, k-role coloring is known to be NP-complete for $k \geqslant 2$ on general graphs. In this paper, we determine k-role coloring of the rooted product of various graphs.

Keywords: *role coloring, role graph, rooted product, binary product.*

РОЛЕВАЯ РАСКРАСКА ГРАФОВ ИЗ КОРНЕВЫХ ПРОИЗВЕДЕНИЙ

М. Комати, П. Рагукумар

Школа передовых наук, Технологический институт, г. Веллор, Индия

k-Ролевая раскраска — это назначение k цветов вершинам графа таким образом, что если любые две вершины окрашены в один и тот же цвет, то набор цветов, назначенных их соседям, также будет одинаковым. Любой граф с n вершинами может быть раскрашен n ролями. Легко определить, допускает ли граф с n вершинами 1-ролевую раскраску, но задача k-ролевой раскраски для $k \geqslant 2$ на произвольных графах является NP-полной. В работе описана k-ролевая раскраска корневого произведения различных графов.

Ключевые слова: *ролевая раскраска, ролевой граф, корневое произведение, бинарное произведение.*

1. Introduction

All graphs considered in this paper are simple, finite, and undirected (except the role graph R; it may have loops). The graph $G = (V, E)$ has the vertex set $V(G)$ and the edge set $E(G)$. The (open) neighborhood $N_G(v) = N(v)$ of vertex v in a graph G is the set of all vertices in G that are adjacent to v, $v \in V$. The degree of a vertex v is indicated by $\deg(v)$, and the minimum and maximum degrees of vertices in G are represented by $\delta(G)$ and $\Delta(G)$, respectively. Let $\alpha(v)$ denote the color of the vertex v, and $\alpha(N(v))$ denote the color set of the neighborhood of v. For the standard graph terminology notions, we follow J. A. Bondy and U. S. R. Murty [1].

Social networks are a part of everyone's life these days. A social network is envisioned as a graph where the edges indicate the relationships between the persons and the vertices represent the individuals in order to research their behavior. In 1991, M. G. Everett and S. Borgatti [2] defined role assignment under the term "role coloring" based on graph models for social networks. A k-role coloring for any graph G is the assignment of precisely k colors

to its vertices such that if any two vertices get the same color, then the set of colors assigned to their neighborhood is also the same. That is, k-role coloring is a surjective map $\alpha : V(G) \to \{1, \ldots, k\}$ such that, for all $u, v \in V(G)$, if $\alpha(u) = \alpha(v)$, then $\alpha(N(u)) = \alpha(N(v))$ [3]. Figure 1 provides an example of role coloring of a graph G.

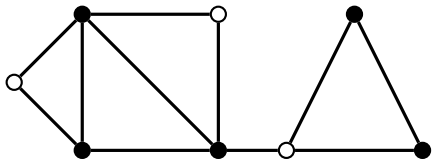


Fig. 1. 2-Role coloring of G

In general, every graph has two trivial role coloring for $k = 1, n$. The color image graph R of a graph G is called a role graph. The role graph R is defined as the graph with $V(R) = \{1, 2, \ldots k\}$ and $E(R) = \{(\alpha(u), \alpha(v)) : (u, v) \in E(G)\}$ and $|V(R)| \leqslant |V(G)|$. Also, for all $v \in V(G)$, $\deg_G(v) \geqslant \deg_R(\alpha(v))$ [3]. Figure 2 displays the possible role graphs for 2-role coloring of connected graphs.

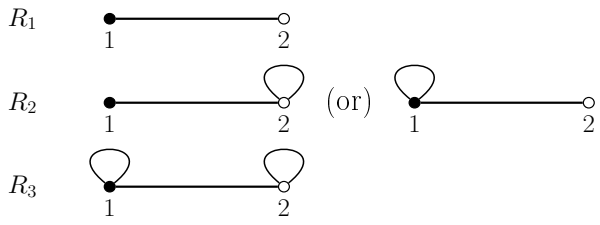


Fig. 2. Role graph

Since each color is assigned to some vertex of G, it is easy to see that if G is connected, the role graph R is also connected. This problem is equivalent to deciding if there exists a locally surjective homomorphism between the graphs G and R [4]. Finding out whether a graph G has a 2-role coloring is NP-complete, as demonstrated by F. S. Roberts and L. Sheng [5]. If the graph is chordal, then k-role assignment can be solved in linear time for $k = 2$ and NP-complete for $k \geqslant 3$ [6]. Role assignments can be computed in polynomial time for proper interval graphs [7]. C. Purcell and P. Rombach [8] proved that k-role coloring is NP-hard for planar graphs, while for trees and cographs it can be solved in polynomial time. They also examined the role coloring for hereditary classes of graphs [9]. Characterization has been done to acquire 3-role coloring in split graphs; it is one of the fascinating graph classes where 2-role coloring is always achievable [10]. S. Pandey and V. Sahlot [3] demonstrated that k-role coloring is NP-complete for bipartite graphs when $k \geqslant 3$. D. Castonguay et al. [11] demonstrated that role assignments restricted to Cartesian products are invariably 2-role colorable.

Based on the work [3], the complexity of 2-role coloring of non-bipartite graphs is evident. So, we are intended to characterize graphs that are 2-role colorable from the rooted product of G and H. Also, we restrict G and H by considering at least one of the graph as non-bipartite.

The rooted product of graphs is one of the well-known binary operations. It was introduced by C. D. Godsil and B. D. McKay [12] in 1978.

Definition 1. The rooted product of two graphs G and H is defined as the graph obtained from G and H by taking one copy of G and $|V(G)|$ copies of H and identifying the i-th vertex of G with the root vertex v in the i-th copy of H for every $i = 1, \ldots, |V(G)|$. It is denoted by $G \circ_v H$.

The paper is organised as follows. The results of role coloring the rooted product of cycles with cycles are presented in Section 2. In Section 3, we determine the role coloring of the rooted product of graphs generated by considering at least one graph from G and H as non-bipartite. The conclusion is given in Section 4.

2. Rooted product of C_m and C_n

Theorem 1. Let $G \cong C_m$ and $H \cong C_n$, where $m = 2k$, $k \geqslant 2$ and $n = 2t$, $t \geqslant 2$. Then $G \circ_v H$ is 2-role colorable with role graph R_1.

Proof. Let $\{u_1, \ldots, u_m\} = V(C_m)$ and $\{v_1, \ldots, v_n\} = V(C_n)$. Let v_r be any arbitrary vertex in C_n. Now we obtain $C_m \circ_v C_n$ by identifying each $u_i \in V(C_m)$ with v_r, this produces m copies of C_n with vertices $\{v_{1,1}, v_{1,2}, \ldots, v_{1,n}, v_{2,1}, v_{2,2}, \ldots, v_{2,n}, \ \ldots, v_{m,1}, v_{m,2}, \ldots, v_{m,n}\}$. Let us assume $v_r = v_1$. Now define $\alpha : V(C_m \circ_v C_n) \to \{1, 2\}$ as follows:

$$\alpha(v_{i,1}) = \begin{cases} 1, & \text{if } i \text{ is odd,} \\ 2, & \text{if } i \text{ is even,} \end{cases} \qquad 1 \leqslant i \leqslant m.$$

Now, for all $v_{1,j} \in V(C_n^{(1)})$ we have:

$$\alpha(v_{1,j}) = \begin{cases} 1, & \text{if } j \text{ is odd,} \\ 2, & \text{if } j \text{ is even,} \end{cases} \qquad 1 \leqslant j \leqslant n.$$

In general, for all $v_{i,j} \in V(C_m \circ_v C_n)$ we have:

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } i, j \text{ have the same parity,} \\ 2, & \text{otherwise.} \end{cases}$$

This gives a 2-role coloring of $C_m \circ_v C_n$ with role graph R_1 since every vertex assigned color 1 has color 2 in its neighborhood and every vertex assigned color 2 has color 1 in its neighborhood. ■

Theorem 2. Let $G \cong C_m$ and $H \cong C_n$, where $m \geqslant 3$ and $n = 2t + 1$, $t \geqslant 1$. Then $G \circ_v H$ is 2-role colorable with role graph R_3.

Proof. Let $\{v_{1,1}, v_{1,2}, \ldots, v_{1,n}, v_{2,1}, v_{2,2}, \ldots, v_{2,n}, \ \ldots, v_{m,1}, v_{m,2}, \ldots, v_{m,n}\}$ be the vertices of $C_m \circ_v C_n$. Let v_r be any arbitrary vertex in C_n. Let $v_r = v_1$ and $v_{i,1}$ be the root vertices identified with the vertices of C_m. Since C_n is odd and non bipartite, assigning colors with role graph R_1 is not possible. Let us define $\alpha : V(C_m \circ_v C_n) \to V(R_3)$.

C a s e (i). Let $H \cong C_{2t+1}$, where t is an odd positive integer. Let us consider $\alpha(v_{i,1}) = 1$ for all $v_{i,1} \in V(C_m \circ_v C_n)$. Here $2 \notin \alpha(N(v_{i,1}))$, thus we have $\alpha(v_{i,2}) = \alpha(v_{i,3}) = 2$. Again $1 \notin \alpha(N(v_{i,3}))$, thus $\alpha(v_{i,4}) = \alpha(v_{i,5}) = 1$. Proceeding in this way we get

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 1 \pmod 4, \\ 2, & \text{if } j \equiv 2 \text{ or } 3 \pmod 4. \end{cases}$$

C a s e (ii). Let $H \cong C_{2t+1}$, where t is an even positive integer. If suppose $\alpha(v_{i,1}) = 1$, then there exist two vertices $v_{i,g}, v_{i,h} \in V(C_m \circ_v C_{2t+1})$, where $\alpha(v_{i,g}) = \alpha(v_{i,h})$ but $\alpha(N(v_{i,g})) \neq$ $\neq \alpha(N(v_{i,h}))$. Thus, we have

$$\alpha(v_{i,1}) = \begin{cases} 1, & \text{if } i \text{ is odd,} \\ 2, & \text{if } i \text{ is even.} \end{cases}$$

In general, for all $v_{i,j} \in V(C_m \circ_v C_{2t+1})$, $j > 1$, we have

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } \big(i \text{ is odd, } j \equiv 1 \text{ or } 2 \pmod 4\big) \text{ or } \big(i \text{ is even, } j \equiv 0 \text{ or } 3 \pmod 4\big), \\ 2, & \text{if } \big(i \text{ is odd, } j \equiv 0 \text{ or } 3 \pmod 4\big) \text{ or } \big(i \text{ is even, } j \equiv 1 \text{ or } 2 \pmod 4\big). \end{cases}$$

Here, each vertex assigned color 1 has both the colors 1 and 2 in its neighborhood; similarly, every vertex assigned color 2 has both the colors 1 and 2 in its neighborhood. This gives a 2-role coloring of $C_m \circ_v C_n$ with role graph R_3. ■

An example illustrating Theorem 2 is shown in Fig. 3.

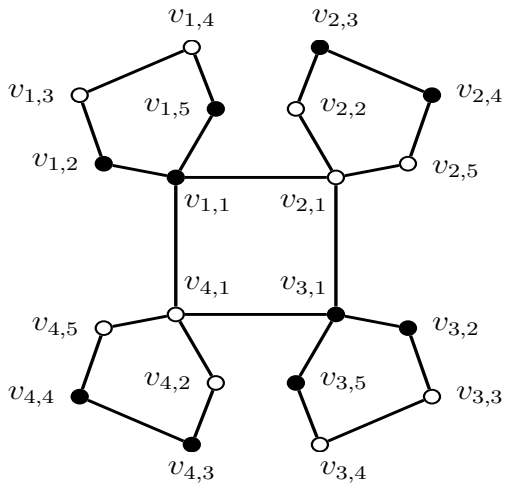


Fig. 3. 2-Role coloring of $C_4 \circ_v C_5$

Theorem 3. Let $G \cong C_m$ and $H \cong C_n$, where $m = 2k + 1$, $k \geqslant 1$ and $n = 2t$, $t \geqslant 2$. If n satisfies any of the following conditions:

(i) $n \equiv 0$ or $6 \pmod{12}$,
(ii) $n \equiv 2$ or $8 \pmod{12}$,
(iii) $n \equiv 4 \pmod{12}$,

then $G \circ_v H$ is 2-role colorable.

Proof. Let $\{u_1, \ldots, u_m\} = V(C_m)$ and $\{v_1, \ldots, v_n\} = V(C_n)$. Let $\{v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be the vertices of $C_m \circ_v C_n$. Let v_r be any arbitrary vertex in C_n. Let $v_r = v_1$ and $v_{i,1}$ be the root vertices identified with the vertices of C_m. Now we define $\alpha : V(C_m \circ_v C_n) \to \{1, 2\}$ as follows.

C a s e (i). Let $n \equiv 0$ or $6 \pmod{12}$, then for all $v_{i,1} \in V(C_{2k+1} \circ_v C_{2t})$ we have $\alpha(v_{i,1}) = 1$. In general, for all $v_{i,j} \in V(C_{2k+1} \circ_v C_{2t})$ we have

$$\alpha(v_{i,j}) = \begin{cases} 2, & \text{if } j \equiv 0 \pmod 3, \\ 1, & \text{otherwise.} \end{cases}$$

C a s e (ii). If suppose $n \equiv 2$ or $8 \pmod{12}$, then

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 1 \pmod 3, \\ 2, & \text{if } j \equiv 2 \pmod 3. \end{cases}$$

Here, every vertex assigned color 1 has both the colors 1 and 2 in its neighborhood. Every vertex assigned color 2 has color 1 in its neighborhood. Thus, it is a 2-role coloring with role graph R_2.

C a s e (iii). Let us consider the case $n \equiv 4 \pmod{12}$. Then for all $v_{i,j} \in V(C_{2k+1} \circ_v C_{2t})$ we have

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 1 \pmod 4, \\ 2, & \text{if } j \equiv 2 \text{ or } 3 \pmod 4. \end{cases}$$

Here, every vertex assigned color 1 has both the colors 1 and 2 in its neighborhood; similarly, every vertex assigned color 2 has both the colors 1 and 2 in its neighborhood. This gives a 2-role coloring of $C_m \circ_v C_n$ with role graph R_3. ■

An example illustrating Theorem 3 is shown in Fig. 4.

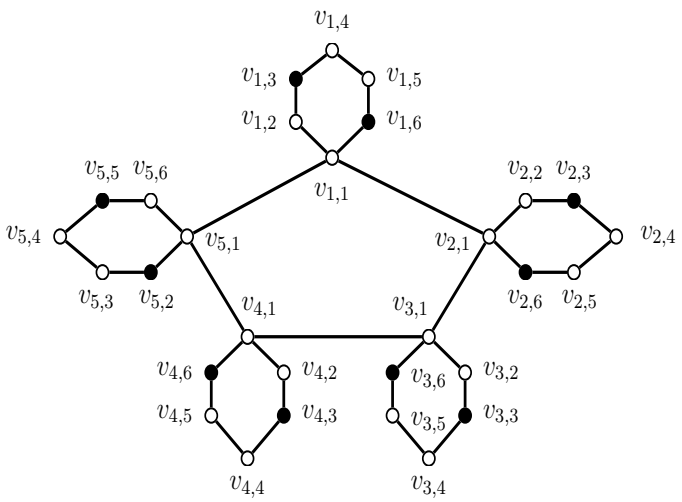


Fig. 4. 2-Role coloring of $C_5 \circ_v C_6$

Theorem 4. Let $G \cong C_m$ and $H \cong C_n$, where $m = 2k + 1$, $k \geqslant 1$ and $n = 2t$, $t \geqslant 2$. If $n \equiv 10 \pmod{12}$, then $G \circ_v H$ is not 2-role colorable.

Proof. Let $\{v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be the vertices of $G \circ_v H$. Let v_r be any arbitrary vertex in C_n. Let $v_r = v_1$ and $v_{i,1}$ be the root vertices identified with the vertices of C_m. Let C_n be an even cycle, thus assigning colors with role graph R_1 results in a contradiction, since the graph C_m is not bipartite. Hence, it can be role colored with the role graph R_2 or R_3. By Theorem 3, the only case left is $n \equiv 10 \pmod{12}$. Let us assume $\alpha : V(C_n) \to V(R_2)$ with a loop on 1 such that, given the vertices $v_1, v_2, v_n \in V(H)$, we have $\alpha(v_1) = \alpha(v_2) = \alpha(v_n) = 1$, where $2 \notin \alpha(N(v_1))$. Now consider $v_{1,j} \in V(C_n^{(1)})$ from $V(C_m \circ_v C_n)$, thus we have

$$\alpha(v_{1,j}) = \begin{cases} 2, & \text{if } j \equiv 0 \pmod 3, \\ 1, & \text{otherwise.} \end{cases}$$

Here $\alpha(v_{1,1}) = 1$, where $2 \notin \alpha(N(v_{1,1}))$. Thus, we assign $\alpha(v_{2,1}) = 2$, which satisfies the neighborhood condition. Hence, for all $v_{2,j} \in V(C_m \circ_v C_n)$, $1 \leqslant j \leqslant n$, we have

$$\alpha(v_{2,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 2 \pmod 3, \\ 2, & \text{otherwise.} \end{cases}$$

Here $\alpha(v_{2,1}) = \alpha(v_{2,m}) = 2$ and $v_{2,1}$ is adjacent to $v_{2,m}$, it follows that $\alpha(N(v_{2,1})) = \alpha(N(v_{2,m})) \in \{1, 2\}$ but this is not true for other vertices colored 2. Hence, it is not 2-role colorable with role graph R_2. Now we define $\alpha : V(C_n) \to V(R_3)$. Consider the vertices $v_1, v_2, v_{n-1}, v_n \in V(C_n)$ such that $\alpha(v_1) = \alpha(v_2) = \alpha(v_{n-1}) = \alpha(v_n) = 1$, where $2 \notin \alpha(N(v_1))$ and $2 \notin \alpha(N(v_n))$. Let us consider $v_{1,j} \in V(C_n^{(1)})$. Thus, we have

$$\alpha(v_{1,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 1 \pmod 4, \\ 2, & \text{otherwise.} \end{cases}$$

Here $1 \notin \alpha(N(v_{1,1}))$, therefore $\alpha(v_{2,1}) = 1$. But $\alpha(v_{1,n}) = 2$, where $2 \notin \alpha(N(v_{1,n}))$ since $\alpha(v_{1,1}) = \alpha(v_{1,(n-1)}) = 1$. Hence, $C_m \circ_v C_n$ is not 2-role colorable with role graph R_3 when $n \equiv 10 \pmod{12}$. ■

The following table summarizes the results from Theorem 1–4.

Role coloring of rooted product of cycles with cycles

Cycles (C_m)	Cycles (C_n)	k-Role coloring of cycles C_m and C_n
When m is even	When n is even	$k = 2$
When m is even	When n is odd	$k = 2$
When m is odd	When n is odd	$k = 2$
When m is odd	When n is even	$k = 2$ when $n \not\equiv 10 \pmod{12}$

3. Rooted product on other graph classes

In this section, we find the role coloring of graphs that are obtained from rooted product of other graph classes.

Theorem 5. Let G be any graph and $H \cong K_n$ or W_n. Then $G \circ_v H$ is 2-role colorable.

Proof. Let $\{v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be the vertices of $G \circ_v H$.

C a s e (i). If suppose $H \cong W_n$, then the root can be either a universal vertex or any vertex in a cycle. Let v_r be any arbitrary vertex in W_n or K_n. Let $v_{i,1}$ be the universal vertex in W_n and $v_r = v_k$, then $v_{i,k}$ be the root vertices identified with the vertices of G. Now define $\alpha : V(G \circ_v H) \to \{1, 2\}$ as follows:

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j = 1, \\ 2, & \text{otherwise.} \end{cases}$$

Let us assume $v_r = v_1$. Then again $\alpha(v_{i,1}) = 1$ and $\alpha(v_{i,j}) = 2$ for $j \neq 1$. If suppose v_r is a universal vertex, then every vertex assigned color 1 has both the colors 1 and 2 in its neighborhood; similarly, every vertex assigned color 2 has both the colors 1 and 2 in its neighborhood. Thus, we obtain a 2-role coloring with role graph R_3. Otherwise, it can have 2-role coloring with role graph R_2.

C a s e (ii). Let us consider $H \cong K_n$. Let $v_r = v_k$ be any arbitrary vertex and $v_{i,k}$ be the root vertices identified with the vertices of G. Then we have

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j = k, \\ 2, & \text{otherwise.} \end{cases}$$

Hence, $G \circ_v K_n$ is 2-role colorable with role graph R_3 since every vertex colored 1 has both the colors 1 and 2 in its neighborhood; similarly, every vertex colored 2 has both the colors 1 and 2 in its neighborhood. ■

An example illustrating Theorem 5 is shown in Fig. 5.

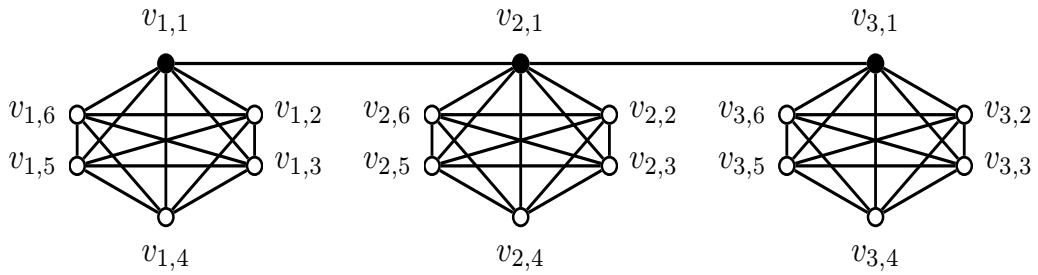


Fig. 5. 2-Role coloring of $P_3 \circ_v K_6$

Theorem 6. Let $G \cong W_m$ or K_m and $H \cong C_n$, where $n = 2t + 1$, $t \geqslant 1$. Then $G \circ_v H$ is 2-role colorable with role graph R_3.

Proof. Let $\{u_1, \ldots, u_m\} = V(G)$ and $\{v_1, \ldots, v_n\} = V(H)$. Let $\{v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be the vertices of $G \circ_v H$. Let v_r be any arbitrary vertex of C_n. Let $v_r = v_1$ be the root. Let $\{v_{1,1}, \ldots, v_{m,1}\} \in V(G)$ in the graph $G \circ_v H$. Here we have two cases based on t.

C a s e (i). Let us consider the case where t is an odd positive integer. Now we define $\alpha : V(G \circ_v H) \to \{1, 2\}$ as follows:

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 1 \pmod 4, \\ 2, & \text{if } j \equiv 2 \text{ or } 3 \pmod 4. \end{cases}$$

C a s e (ii). Let us consider the case where t is an even positive integer. Let $v_{i,1}$ be a universal vertex in W_m and any arbitrary vertex in K_m. Then we have

$$\alpha(v_{1,j}) = \begin{cases} 1, & \text{if } j \equiv 1 \text{ or } 2 \pmod 4, \\ 2, & \text{if } j \equiv 0 \text{ or } 3 \pmod 4, \end{cases}$$

$$\alpha(v_{i,j}) = \begin{cases} 1, & \text{if } j \equiv 0 \text{ or } 3 \pmod 4, \\ 2, & \text{if } j \equiv 1 \text{ or } 2 \pmod 4, \end{cases} \quad i > 1.$$

Here, every vertex assigned color 1 has both the colors 1 and 2 in its neighborhood; similarly, every vertex assigned color 2 has both the colors 1 and 2 in its neighborhood. Hence, this is a 2-role coloring of $G \circ_v H$ with role graph R_3. ■

Lemma 1. Let P_n be a path, where $n \geqslant 2$. If P_n is 2-role colorable with role graph R_2, then $|E(P_n)| = 3k$, where k is a positive integer.

Proof. Let $\{v_1, \ldots, v_n\} = V(P_n)$. Let us assume that $|E(P_n)| \neq 3k$, $k = 1, 2, \ldots$ Let us define $\alpha : V(P_n) \to V(R_2)$. Thus, $|V(P_n)| \geqslant 4$, and hence the length of P_n should be at least 3. Now assume $|E(P_n)| > 3k$ and $|V(P_n)| > 3k + 1$. Let us define $\alpha : V(P_n) \to \{1, 2\}$ as follows:

$$\alpha(v_i) = \begin{cases} 2, & \text{if } i \equiv 0 \text{ or } 2 \pmod 3, \\ 1, & \text{otherwise.} \end{cases}$$

Thus, each vertex assigned color 1 must have color 2 in its neighborhood, and each vertex assigned color 2 must have both the colors 1 and 2 in its neighborhood. Here, $\alpha(v_n) = 2$ and either $2 \notin \alpha(N(v_n))$ or $1 \notin \alpha(N(v_n))$, since $n \equiv 0$ or $2 \pmod 3$. This gives a contradiction that P_n is 2-role colorable. Hence, $|E(P_n)| = 3k$. ■

Theorem 7. Let G be a non-bipartite graph and $H \cong P_n$ be a path where $n \geqslant 2$. Let v_r be a root vertex in P_n. If P_n satisfies any of the following conditions:

(i) $|V(P_n)| = 3k + 1$, where k is a positive integer and $v_r = v_s$, $s \equiv 0$ or $2 \pmod 3$;
(ii) $|V(P_n)| = 3k + 2$ and $v_r = v_n$,

then $G \circ_v H$ is 2-role colorable with role graph R_2.

Proof. Let $\{u_1, \ldots, u_m\} = V(G)$ and $\{v_1, \ldots, v_n\} = V(P_n)$. Let $\{v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be the vertices of $G \circ_v P_n$. Let us define a mapping $\alpha : V(G \circ_v P_n) \to \to \{1, 2\}$. Here, assigning colors to P_n with role graph R_1 is not possible because G is non-bipartite. Hence, we consider the role graph R_2.

C a s e (i). Let $|V(P_n)| = 3k + 1$ and $v_r = v_s$, $s \equiv 0$ or $2 \pmod 3$. By Lemma 1, it is obvious that P_n of length $3k$ is 2-role colorable by role graph R_2. Thus, for all $v_i \in V(P_n)$, if $\alpha(v_i) = 1$, then $\alpha(N(v_i)) = 2$, and if $\alpha(v_i) = 2$, then $\alpha(N(v_i)) \in \{1, 2\}$. If suppose $v_r = v_1$, then $\alpha(v_1) = 1$. Now for all $v_{i,1} \in V(G \circ_v P_n)$ we have $\alpha(v_{i,1}) = 1$, where $\alpha(N(v_{i,1})) \in \{1, 2\}$, but for all $v_{i,j} \in V(G \circ_v P_n)$, $j \neq 1, s$, we have $\alpha(v_{i,j}) = 1$, where $1 \notin \alpha(N(v_{i,j}))$. Thus, we consider $v_r = v_s$, here $\alpha(v_s) = 2$ for all $s \equiv 0$ or $2 \pmod 3$ such that for all $v_{i,s} \in V(G \circ_v P_n)$ we have $\alpha(v_{i,s}) = 2$ and $\alpha(N(v_{i,s})) \in \{1, 2\}$. This gives a 2-role coloring of $G \circ_v P_n$ with role graph R_2.

C a s e (ii). Let $|V(P_n)| = 3k + 2$ and $v_r = v_n$. Here $2 \notin \alpha(N(v_n))$, but $v_{i,n} \in V(G \circ_v P_n)$ be the root vertices identified with vertices of G such that $\alpha(v_{i,n}) = 2$ and $\alpha(N(v_{i,n})) \in \in \{1, 2\}$, which satisfies the adjacency condition with role graph R_2. ■

Theorem 8. Let G be a non-bipartite graph and $H \cong S_n$ be a star where $n \geqslant 2$. Let v_r be a root vertex in S_n. Then $G \circ_v H$ is 2-role colorable if and only if v_r is the central vertex.

Proof.

$\Rightarrow$: Let $G \circ_v S_n$ be 2-role colorable. On the contrary, we assume $v_r = v_1$ as the leaf vertex in S_n. Let us define a mapping $\alpha : V(G \circ_v S_n) \to \{1, 2\}$. Let $v_{i,1}$ be the root vertices identified with the vertices of G and $v_{i,2}$ be the central vertex of $S_n^{(i)}$. Let $(v_{i,j})$, $j \neq 1, 2$, be the leaf vertices of $S_n^{(i)}$. Now, we assume that $\alpha(v_{i,j}) = 1$ for $j \neq 1, 2$ and $\alpha(v_{i,2}) = 2$. Here, $v_{i,1}$ cannot be colored with role graph R_1, since the graph G is non-bipartite. Thus, if we assign color 1 to $v_{i,1}$, then $\alpha(N(v_{i,1})) \in \{1, 2\}$ but this is not true for all $(v_{i,j})$, since $1 \notin \alpha(N(v_{i,j}))$ for $j \neq 1, 2$. And if we assign color 2 to $v_{i,1}$ then $1 \notin \alpha(N(v_{i,1}))$. Thus, the color of $v_{i,1}$ cannot be the same as the color of $(v_{i,j})$ for $j \neq 1$, which contradicts the assumption that $G \circ_v S_n$ is 2-role colorable. Hence, v_r must be the central vertex.

$\Leftarrow$: Let $v_r = v_1$ be the central vertex. Let us define $\alpha : V(G \circ_v S_n) \to \{1, 2\}$ as follows:

$$\alpha(v_{i,j}) = \begin{cases} 1, & j = 1, \\ 2, & \text{otherwise.} \end{cases}$$

Here, every vertex assigned color 1 has both the colors 1 and 2 in its neighborhood. Every vertex assigned color 2 has color 1 in its neighborhood. Hence, it is a 2-role coloring with role graph R_2. ■

4. Conclusion

In this paper, we explored the role coloring of non-bipartite graphs generated by rooted products between various generic graph classes. Since k-role coloring is NP-complete on non-bipartite graphs when $k = 2$, we characterized graphs obtained from rooted product that are 2-role colorable.

Acknowledgement

The first author expresses her gratitude to the Vellore Institute of Technology, Vellore, for providing financial support that enabled the author to carry out the research work.

REFERENCES

1. *Bondy J. A. and Murty U. S. R.* Graph Theory. London, Springer, 2008.
2. *Everett M. G. and Borgatti S.* Role colouring a graph. Math. Social Sci., 1991, vol. 21, no. 2, pp. 183–188.
3. *Pandey S. and Sahlot V.* Role coloring bipartite graphs. Discrete Appl. Math., 2022, vol. 322, pp. 276–285.
4. *Fiala J. and Paulusma D.* A complete complexity classification of the role assignment problem. Theor. Comput. Sci., 2005, vol. 349, no. 1, pp. 67–81.
5. *Roberts F. S. and Sheng L.* How hard is it to determine if a graph has a 2 role assignment? Networks, 2001, vol. 37, no. 2, pp. 67–73.
6. *Van't Hof P., Paulusma D., and van Rooij J. M.* Computing role assignments of chordal graphs. Theor. Comput. Sci., 2010, vol. 411, no. 40–42, pp. 3601–3613.
7. *Heggernes P., van't Hof P., and Paulusma D.* Computing role assignments of proper interval graphs in polynomial time. LNCS, 2011, vol. 6460, pp. 167–180.
8. *Purcell C. and Rombach P.* On the complexity of role colouring planar graphs, trees and cographs. J. Discrete Algorithms, 2015, vol. 35, pp. 1–8.
9. *Purcell C. and Rombach P.* Role colouring graphs in hereditary classes. Theor. Comput. Sci., 2021, vol. 876, pp. 12–24.
10. *Dourado M. C.* Computing role assignments of split graphs. Theor. Comput. Sci., 2016, vol. 635, pp. 74–84.
11. *Castonguay D., Dias E. S., Mesquita F. N., and Nascimento J. R.* Computing role assignments of cartesian product of graphs. RAIRO-Operations Research, 2023, vol. 57, no. 3, pp. 1075–1086.
12. *Godsil C. D. and McKay B. D.* A new graph product and its spectrum. Bull. Australian Math. Society, 1978, vol. 18, no. 1, pp. 21–28.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.16 DOI 10.17223/20710410/68/7

О СРЕДНЕМ ЧИСЛЕ ДОПУСТИМЫХ РЕШЕНИЙ В ЗАДАЧЕ О РЮКЗАКЕ

М. С. А. Волков*, Э. Н. Гордеев*, В. К. Леонтьев**

*Московский государственный технический университет им. Н. Э. Баумана, г. Москва, Россия

**Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН, г. Москва, Россия

E-mail: sabina-volkoff@yandex.ru, werhorn@yandex.ru, vkleontiev@yandex.ru

Анализируются характеристики решений задачи об ограниченном рюкзаке. Выведены выражения для вычисления среднего значения целевой функции среди всех допустимых решений, а также формулы, связывающие количество решений с подзадачами меньшей размерности. Для случаев, когда переменные принимают значения из множества $\{0, 1\}$ или $\{0, 1, 2\}$, определены формулы для оценки среднего числа допустимых решений во всех задачах заданной размерности при ограниченных значениях коэффициентов весов. Рассмотрена производящая функция, описывающая количество решений рюкзачных задач фиксированной размерности, где компоненты вектора весов принадлежат заданному диапазону. Полученные результаты могут быть полезны при анализе вычислительной сложности алгоритмов решения задачи о рюкзаке.

Ключевые слова: *задача о рюкзаке, производящие функции, динамическое программирование, NP-полные задачи, метод коэффициентов*

ON THE AVERAGE NUMBER OF SOLUTIONS IN THE KNAPSACK PROBLEM

M. S. A. Volkov*, E. N. Gordeev*, V. K. Leontiev**

*Bauman Moscow State Technical University, Moscow, Russia

**Dorodnitsyn Computing Center of the Russian Academy of Sciences, Moscow, Russia

Exact analytical expressions are derived for the average number of solutions to the bounded knapsack problem over a set of fixed-dimension instances. The average number of solutions for a set of knapsack problems with the constraint $\sum\limits_{i=1}^{n} a_i x_i \leqslant b$, where the coefficients a_i do not exceed a given value p, is denoted as $|\bar{V}_p|$. Formulas are obtained that relate the number of solutions to problem parameters such as the dimension n, weight limit p, and allowable variable values. For Boolean variables $x_i \in \{0, 1\}$, the following formula is derived:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \sum_{k=0}^{n} \binom{n}{k} \binom{b}{n-k} (b+2)^k.$$

For the case $x_i \in \{0, 1, 2\}$, a generalized expression is obtained:

$$|\bar{V}_b| = \sum_{k=0}^{n} \binom{n}{k} (b+1)^{k-n} \sum_{t=0}^{n-k} \binom{n-k}{t} 2^t \left(\binom{(n-k+t+b-1)/2}{n-k} \left[n-k+t+b=1 \pmod 2\right] + \right.$$
$$\left. + \binom{(n-k+t+b)/2}{n-k} \left[n-k+t+b=0 \pmod 2\right] \right).$$

Additionally, a formula is derived that defines the generating function for the volume of the set of solutions to problems of dimension n with components of the weight vector $(a_1, \ldots, a_n)$ taking values in the range from 0 to p. The obtained results can be applied to assess the computational complexity of knapsack problem algorithms, select optimal solution methods, develop decomposition algorithms, and analyze combinatorial structures arising in discrete optimization problems.

Keywords: *knapsack problem, generating functions, dynamic programming, NP-complete problems, coefficient method.*

Введение

Задача об ограниченном рюкзаке представляет собой обобщение классической задачи о 0-1-рюкзаке, в которой каждый предмет может быть выбран не более заданного количества раз. В данной постановке каждому предмету сопоставляются три параметра: вес, ценность и максимально допустимое число копий. Как и в базовом варианте, цель состоит в том, чтобы подобрать такой набор предметов, который максимизирует суммарную ценность, не превышая заданное ограничение на вес. Математическая модель этой задачи представлена следующими условиями [1]:

$$\sum_{j=1}^{n} c_j x_j \to \max; \tag{1}$$

$$\sum_{i=1}^{n} a_i x_i \leqslant b, \tag{2}$$

где $x = (x_1, \ldots, x_n)$ — n-мерный вектор с целочисленными компонентами $x_i \in \{0, 1, \ldots, m\}$; $c_1, \ldots, c_n$, $a_1, \ldots, a_n$, b — неотрицательные целые числа.

Задача о рюкзаке является одной из фундаментальных проблем комбинаторной оптимизации, находя применение в различных областях науки и техники, где необходимо выбрать оптимальный набор элементов из ограниченного множества. Варианты данной задачи часто возникают при ослаблении условий задач целочисленного программирования, что обусловило её активное изучение в последние десятилетия. Это привело к появлению значительного объёма исследований, затрагивающих как алгоритмические аспекты решения задачи, так и её теоретические свойства. Детальный обзор существующих подходов представлен, в частности, в фундаментальных работах [1, 2].

Задача о рюкзаке относится к классу NP-трудных задач, что означает, что нахождение точного решения требует значительных вычислительных ресурсов даже при относительно небольших значениях n. В связи с этим широко применяются методы декомпозиции, позволяющие разбирать исходную задачу на более мелкие подзадачи, что упрощает процесс поиска оптимального решения.

Одним из ключевых декомпозиционных подходов является метод ветвей и границ. Он основан на поэтапном разделении задачи на подзадачи с последующим исключением тех из них, которые не могут содержать оптимальное решение.

Особый интерес представляют исследования, посвященные анализу точной и приближённой оценок сложности метода ветвей и границ применительно к задаче о рюкзаке. Значительный вклад в эту область внесли М. А. Посыпкин и Р. М. Колпаков, которые рассматривали частные случаи применения данного метода. В работе [3] предложены две верхние оценки сложности решения задачи о рюкзаке методом ветвей и границ, выраженные через параметры исходных данных, а также выделен случай, при котором сложность метода ограничена полиномиально относительно размерности задачи. В [3] рассмотрены также оценки сложности метода для задачи о сумме подмножеств, являющейся частным случаем задачи о рюкзаке. В [4] представлена верхняя оценка сложности решения задачи о сумме подмножеств с дополнительным критерием отсечения подзадач, основанным на сравнении предельного и минимального числа предметов, которые могут быть добавлены в рюкзак.

В [5–7] рассматриваются вопросы, связанные со сложностью параллельных вычислений при решении задач оптимизации, включая задачу о рюкзаке, в распределённых вычислительных средах. Различные вариации задачи о рюкзаке и подходы к их решению исследуются в работах [8–10].

Один из результатов настоящей работы можно рассматривать как обобщение метода, предложенного в [11], где автор анализирует оптимальную стратегию применения метода ветвей и границ к частному случаю задачи о рюкзаке с равными весами предметов и двумя возможными значениями их стоимости. Это расширяет понимание эффективности метода ветвей и границ в различных сценариях задачи о рюкзаке.

Проведённые исследования, посвящённые точным и приближённым оценкам сложности метода, учитывают особенности различных вариантов задачи, что способствует более точному прогнозированию его производительности и выбору наиболее эффективной стратегии решения. В таких подходах активно используются оценки значений функционала на множестве допустимых решений, что делает задачу их вычисления актуальной.

В данной работе выведены комбинаторные формулы для оценки мощности множества решений задачи о рюкзаке в зависимости от заданных параметров. В п. 1 представлены производящие функции в виде рациональных выражений, описывающие множество допустимых решений и соответствующие значения функционала. Пункт 2 содержит оценки среднего числа допустимых решений для всех задач размерности n, где компоненты вектора весов находятся в пределах от 0 до b. Полученные результаты могут быть полезны при анализе вычислительной сложности алгоритмов решения задачи о рюкзаке. Понимание структуры пространства решений и распределения значений целевой функции на множестве допустимых решений позволяет оценить потенциальную сложность или эффективность применяемых алгоритмов.

Для получения ключевых результатов использован метод коэффициентов [12]. Этот метод является разновидностью метода производящих функций, задавая линейный функционал для множества формальных степенных рядов с конечным числом членов с отрицательными степенями. Метод соотносит каждому степенному ряду коэффициент при его члене с показателем минус первой степени. Для рядов, сходящихся в окрестности нуля, значение коэффициента совпадает с вычетом функции в точке 0. Применение метода производящих функций к задаче о рюкзаке с булевыми переменными рассмотрено в работах [13, 14]. В данной работе этот подход используется для получения формул для задачи об ограниченном рюкзаке с возможностью повторного использования предметов.

1. Вспомогательные утверждения

Определим производящие функции в виде формальных степенных рядов, которые описывают множество допустимых решений и множество возможных значений функционала задачи. Обозначим множество допустимых решений исходной задачи V_b. Оно состоит из n-мерных векторов x с $x_i \in \{0, 1, \ldots, m\}$, $i = 1, \ldots, n$, удовлетворяющих неравенству (2). Объёмом V_b назовём число $|V_b|$ допустимых решений неравенства (2). Для выражения распределения точек на множестве допустимых решений используется степенной ряд

$$P_b(z_1, z_2, \ldots, z_n) = \sum_{x \in V_b} z_1^{a_1 x_1} z_2^{a_2 x_2} \ldots z_n^{x_n a_n}.$$

В работе [15] при помощи производящих функций получены оценки функционала задачи о рюкзаке с булевыми переменными. Приведённые далее лемма и следствие строго доказаны в [16], они используются для получения и обоснования текущих результатов, связанных с оценкой количества решений задачи об ограниченном рюкзаке.

Лемма 1 [16]. Для задачи об ограниченном рюкзаке (1), (2) справедлива формула

$$\sum_{b=0}^{\infty} P_b(z_1, \ldots, z_n) u^b = \frac{(1+(z_1u)^{a_1}+\ldots+(z_1u)^{ma_1}) \ldots (1+(z_nu)^{a_n}+\ldots+(z_nu)^{ma_n})}{1-u}.$$

Следствие 1 [16]. Для объёма множества допустимых решений задачи (1), (2) с $m \in \mathbb{N}$ имеет место

$$|V_b| = \operatorname*{coef}_u \left\{ \frac{(1 + u^{a_1} + \ldots + u^{ma_1}) \ldots (1 + u^{a_n} + \ldots + u^{ma_n})}{(1-u)u^{b+1}} \right\}. \tag{3}$$

Здесь и далее $\operatorname*{coef}_u\{A(u)\} = \dfrac{1}{2\pi i} \oint\limits_{|u|=\rho} A(u)du = a_{-1}$, где a_{-1} — коэффициент при минус первой степени многочлена $A(u)$. Подробное описание данного функционала и его свойств приведено в [12].

2. Среднее число решений множества задач одинаковой размерности

В выражении (3) приведена формула для вычисления числа решений конкретной задачи о рюкзаке. Определим среднее число решений для некоторого множества задач о рюкзаке с фиксированными параметрами.

Обозначим через $\bar{V}_p$ среднее число решений набора задач об ограниченном рюкзаке (1), (2), где коэффициенты весов a_i, $i = 1, \ldots, n$, не превышают некоторого заданного значения p. Это число выражается следующей формулой:

$$|\bar{V}_p| = \frac{1}{(p+1)^n} \sum_{\substack{0 \leqslant a_i \leqslant p, \\ i=1,\ldots,n}} |V_b(a_1, \ldots, a_n)|. \tag{4}$$

Рассмотрим вопрос о среднем числе допустимых решений задач о рюкзаке при различных значениях количества копий предметов m.

Пусть значение b и размерность задачи n фиксированы, а компоненты вектора весов $(a_1, \ldots, a_n)$ принимают значения в диапазоне от 0 до b. Формула для вычисления среднего числа решений по всем таким задачам в частном случае, когда $m = 1$, получена и доказана в работе [17]. Далее приводится формулировка соответствующей теоремы, которая будет обобщена на случай, когда переменные принимают значения из множества $x \in \{0, 1, 2\}^n$.

Теорема 1 [17]. При $x \in \{0,1\}^n$ справедлива формула

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \sum_{k=0}^{n} \mathrm{C}_n^k \mathrm{C}_b^{n-k} (b+2)^k. \tag{5}$$

Продемонстрируем справедливость формулы на элементарном примере.

Пример 1. При $n = 2$, $b = 2$, $x \in \{0,1\}^n$ существует девять ограничений: $2x_1 + 2x_2 \leqslant 2$; $2x_1 + x_2 \leqslant 2$; $x_1 + 2x_2 \leqslant 2$; $x_1 + x_2 \leqslant 2$; $x_1 + 0x_2 \leqslant 2$; $2x_1 + 0x_2 \leqslant 2$; $0x_1 + x_2 \leqslant 2$; $0x_1 + 2x_2 \leqslant 2$; $x_1 + 0x_2 \leqslant 2$.

Очевидно, что последние шесть неравенств выполняются для всех x, т. е. имеют по четыре решения. Первые три неравенства имеют по три решения.

Таким образом, $|\bar{V}_b| = 33/9 = 11/3$. Такое же значение получается из формулы (5):

$$|\bar{V}_b| = \frac{1}{(2+1)^2} \sum_{k=0}^{2} \mathrm{C}_2^k \, \mathrm{C}_2^{2-k} \, (2+2)^k = \frac{1}{9}(\mathrm{C}_2^0 \, \mathrm{C}_2^2 \, 4^0 + \mathrm{C}_2^1 \, \mathrm{C}_2^1 \, 4^1 + \mathrm{C}_2^2 \, \mathrm{C}_2^0 \, 4^2) = \frac{33}{9} = \frac{11}{3}.$$

Формула (5) может быть полезна для оценки сложности алгоритмов решения задачи о 0-1-рюкзаке, поскольку позволяет определить число допустимых решений и их распределение. Это помогает прогнозировать вычислительные затраты и выбирать наиболее подходящие методы решения. Кроме того, она может использоваться для анализа структуры множества решений, что важно при разработке приближённых алгоритмов и изучении эффективности разных подходов к задаче о рюкзаке.

Рассмотрим теперь вопрос о среднем значении мощности множества допустимых решений в более общем случае. Следующая формула выражает число решений каждой задачи размерности n с компонентами вектора весов $(a_1, \ldots, a_n)$, принимающими значения в диапазоне от 0 до p:

$$R_p(z_1, \ldots, z_n) = \sum_{\substack{0 \leqslant a_i \leqslant p, \\ i=1,\ldots,n}} z_1^{a_1}, \ldots, z_n^{a_n} |V_b(a_1, \ldots, a_n)|. \tag{6}$$

Теорема 2. Справедлива формула

$$R_p(z_1, \ldots, z_n) = \\ = \operatorname*{coef}_u \left\{ \prod_{k=1}^{n} \left(\frac{1 - z_k^{p+1}}{1 - z_k} + \frac{1 - (z_k u)^{p+1}}{1 - z_k u} + \ldots + \frac{1 - (z_k u^m)^{p+1}}{1 - z_k u^m} \right) / \big(u^{b+1}(1-u)\big) \right\}. \tag{7}$$

Доказательство. Подставим в (6) выражение для числа решений из (3):

$$R_p(z_1, \ldots, z_n) = \sum_{\substack{0 \leqslant a_i \leqslant p, \\ i=1,\ldots,n}} z_1^{a_1}, \ldots, z_n^{a_n} \operatorname*{coef}_u \left\{ \frac{(1 + u^{a_1} + \ldots + u^{ma_1}) \ldots (1 + u^{a_n} + \ldots + u^{ma_n})}{(1-u)u^{b+1}} \right\}.$$

Объединим выражения, в которых суммирование производится по одинаковому компоненту a_i:

$$R_p(z_1, \ldots, z_n) = \operatorname*{coef}_u \left\{ \frac{1}{(1-u)u^{b+1}} \Big(\sum_{a_1=1}^{p} z_1^{a_1}(1 + u^{a_1} + \ldots + u^{ma_1}) \ldots \sum_{a_n=1}^{p} z_n^{a_n}(1 + u^{a_n} + \ldots + u^{ma_n}) \Big) \right\}.$$

Теперь заметим, что каждое выражение под знаком суммы можно разложить в $p+1$ сумму геометрической прогрессии:

$$\sum_{a_k=1}^{p} z_k^{a_k}(1+u^{a_k}+\ldots+u^{ma_k}) = \sum_{a_k=1}^{p} z_k^{a_k} + \sum_{a_k=1}^{p} z_k^{a_k}u_k^{a_k} + \ldots + \sum_{a_k=1}^{p} z_k^{a_k}u_k^{ma_k} =$$
$$= \frac{1-z_k^{p+1}}{1-z_k} + \frac{1-(z_ku)^{p+1}}{1-z_ku} + \ldots + \frac{1-(z_ku^m)^{p+1}}{1-z_ku^m}.$$

Подставляя полученное выражение в исходную формулу, получим (7). ■

Заменив все аргументы левой части формулы (7) значением z, получим производящую функцию, выражающую общее число решений задач с одинаковой суммой коэффициентов:

$$R_p(z) = \operatorname*{coef}_u\left\{\prod_{k=1}^{n}\left(\frac{1-z^{p+1}}{1-z} + \frac{1-(zu)^{p+1}}{1-zu} + \ldots + \frac{1-(zu^m)^{p+1}}{1-zu^m}\right)\Big/\big(u^{b+1}(1-u)\big)\right\}.$$

Данная производящая функция может быть адаптирована для решения задач, связанных с использованием специфичных комбинаторных моделей, отражающих особенности конкретной области.

Теорема 3. При $x \in \{0,1,2\}^n$ справедлива формула

$$|\bar{V}_b| = \sum_{k=0}^{n} \mathrm{C}_n^k (b+1)^{k-n} \sum_{t=0}^{n-k} \mathrm{C}_{n-k}^t 2^t \Big(\mathrm{C}_{(n-k+t+b-1)/2}^{n-k}\big[n-k+t+b=1 \pmod 2\big] + \\ + \mathrm{C}_{(n-k+t+b)/2}^{n-k}\big[n-k+t+b=0 \pmod 2\big]\Big). \tag{8}$$

Здесь $[P]$ — скобка Айверсона, равная 1, если условие P выполняется, и 0 в противном случае.

Доказательство. Подставляя значение из формулы (3) при $m=2$ в выражение (4), получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{\sum\limits_{a_1=0}^{b}(1+u^{a_1}+u^{2a_1})\ldots\sum\limits_{a_n=0}^{b}(1+u^{a_n}+u^{2a_n})}{(1-u)u^{b+1}}\right\}.$$

Заметим, что каждую сумму в числителе под знаком коэффициента можно разложить, а степени u собрать в две суммы геометрических прогрессий:

$$\sum_{a_i=0}^{b}(1+u^{a_i}+u^{2a_i}) = b+3+\sum_{a_i=1}^{b}u^{a_i}+\sum_{a_i=1}^{b}u^{2a_i} = b+3+\frac{(1-u^b)u}{1-u}+\frac{(1-u^{2b})u^2}{1-u^2}.$$

Подставим полученное выражение вместо сумм по a_i для каждого $i=1,\ldots,n$:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{\Big(b+3+\frac{(1-u^b)u}{1-u}\Big(1+\frac{(1+u^b)u}{1+u}\Big)\Big)^n}{(1-u)u^{b+1}}\right\}.$$

Разложим числитель по формуле бинома Ньютона:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n}\mathrm{C}_n^k(b+3)^k\Big(\frac{u(1-u^b)}{1-u}\Big)^{n-k}\Big(1+\frac{(1+u^b)u}{1+u}\Big)^{n-k}\right\}.$$

Получившуюся сумму разложим на слагаемые с $k < n$ и $k = n$:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u(1-u^b)}{1-u}\Big)^{n-k}\Big(1+\frac{(1+u^b)u}{1+u}\Big)^{n-k}\right\}+ \\ +\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}. \tag{9}$$

Теперь разложим множитель $(1-u^b)^{n-k}$ из числителя первого слагаемого по формуле бинома Ньютона:

$$(1-u^b)^{n-k} = 1 - \mathrm{C}_{n-k}^1u^b + \mathrm{C}_{n-k}^2u^{2b} - \ldots + (-1)^{n-k}\mathrm{C}_{n-k}^{n-k}u^{(n-k)b} = \sum_{i=0}^{n-k}(-1)^i\mathrm{C}_{n-k}^iu^{ib}. \tag{10}$$

Подставляя в формулу (9) разложение (10), получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\Big(\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\Big(1+\frac{(1+u^b)u}{1+u}\Big)^{n-k}\times \\ \times\sum_{i=0}^{n-k}(-1)^i\mathrm{C}_{n-k}^iu^{ib}\Big)\right\}+\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}.$$

Из свойств коэффициента следует, что данное выражение принимает нулевое значение, когда степень u в числителе больше или равна степени u с положительным знаком в знаменателе. Поскольку в первом слагаемом $k < n$, данное выражение обращается в нуль для всех $i > 0$. Таким образом, получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\Big(1+\frac{(1+u^b)u}{1+u}\Big)^{n-k}\right\}+ \\ +\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}.$$

Разложим множитель $\left(1+\dfrac{(1+u^b)u}{1+u}\right)^{n-k}$ в первом слагаемом по формуле бинома Ньютона:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\sum_{t=0}^{n-k}\mathrm{C}_{n-k}^t\Big(\frac{(1+u^b)u}{1+u}\Big)^t\right\}+ \\ +\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}. \tag{11}$$

Разложим теперь первый коэффициент по множителю $(1+u^b)^t$ аналогично (10):

$$(1+u^b)^t = 1 + \mathrm{C}_t^1u^b + \mathrm{C}_t^2u^{2b} + \ldots + \mathrm{C}_t^tu^{tb} = \sum_{i=0}^{t}(-1)^i\mathrm{C}_t^iu^{ib}.$$

Подставляя это разложение в выражение (11) и проводя аналогичные рассуждения, получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\sum_{t=0}^{n-k}\mathrm{C}_{n-k}^t\Big(\frac{u}{1+u}\Big)^t\right\}+$$
$$+\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}.$$

Преобразуем обратно в бином последнюю сумму в первом слагаемом:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n-1}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\Big(\frac{u}{1+u}+1\Big)^{n-k}\right\}+$$
$$+\frac{(b+3)^n}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\right\}.$$

Заметим, что $\mathrm{C}_n^n\Big(\frac{u}{1-u}\Big)^{n-n}\Big(\frac{u}{1+u}+1\Big)^{n-n}=1$, и занесём второе слагаемое под знак суммы первого:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n}\mathrm{C}_n^k(b+3)^k\Big(\frac{u}{1-u}\Big)^{n-k}\Big(\frac{u}{1+u}+1\Big)^{n-k}\right\}.$$

И снова соберём получившуюся сумму в бином:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\Big(b+3+\frac{2u^2+u}{1-u^2}\Big)^n\right\}.$$

Теперь разложим по формуле бинома Ньютона, оставляя 2 во втором слагаемом:

$$|\bar{V}_b| = \frac{1}{(b+1)^n}\operatorname*{coef}_u\left\{\frac{1}{u^{b+1}(1-u)}\sum_{k=0}^{n}\mathrm{C}_n^k(b+1)^k\Big(\frac{u+2}{1-u^2}\Big)^{n-k}\right\}.$$

Вынесем множители, не зависящие от u, за знак коэффициента, разложим выражение по последнему множителю и сгруппируем получившиеся множители:

$$|\bar{V}_b| = \sum_{k=0}^{n}\mathrm{C}_n^k(b+1)^{k-n}\operatorname*{coef}_u\left\{\sum_{t=0}^{n-k}\mathrm{C}_{n-k}^t 2^t u^{n-k-t-b-1}(1-u^2)^{k-n-1}(1+u)\right\}.$$

Разложив последнее выражение по множителю $(1+u)$ на два слагаемых и преобразовав получившиеся выражения в биномиальные коэффициенты в соответствии с правилом $\operatorname*{coef}_u\Big\{(1-v)^n v^{-k-1}\Big\}=(-1)^k\mathrm{C}_n^k$ метода коэффициентов для u^2, получим

$$|\bar{V}_b| = \sum_{k=0}^{n}\mathrm{C}_n^k(b+1)^{k-n}\sum_{t=0}^{n-k}\mathrm{C}_{n-k}^t 2^t\Big(\mathrm{C}_{k-n-1}^{(k-n+t+b-1)/2}(-1)^{(k-n+t+b-1)/2}\big[n-k+t+b=1\,(\mathrm{mod}\,2)\big]+$$
$$+\mathrm{C}_{k-n-1}^{(k-n+t+b)/2}(-1)^{(k-n+t+b)/2}\big[n-k+t+b=0\,(\mathrm{mod}\,2)\big]\Big).$$

Наконец, преобразуя биномиальные коэффициенты по правилу $(-1)^{n-m}\mathrm{C}_{-(m+1)}^{n-m}=\mathrm{C}_n^m$ [18, с. 89], получим искомое выражение (8). ■

Пример 2. При $n = 2$, $b = 2$, $x \in \{0,1,2\}^n$ существует девять ограничений: $2x_1+2x_2\leqslant 2$; $2x_1 + x_2 \leqslant 2$; $x_1 + 2x_2 \leqslant 2$; $x_1 + x_2 \leqslant 2$; $2x_1 + 0x_2 \leqslant 2$; $0x_1 + 2x_2 \leqslant 2$; $x_1 + 0x_2 \leqslant 2$; $0x_1 + x_2 \leqslant 2$; $0x_1 + 0x_2 \leqslant 2$. Очевидно, что последние три неравенства выполняются для всех x, т. е. имеют девять решений. Остальные имеют соответственно 3, 4, 4, 6, 6 и 6 решений. Таким образом, $|\bar{V}_b| = 56/9$. Такое же значение получается из формулы (8):

$$|\bar{V}_b| = \sum_{k=0}^{2} \mathrm{C}_2^k (b+1)^{k-2} \sum_{t=0}^{2-k} \mathrm{C}_{2-k}^t 2^t \Big(\mathrm{C}_{(2-k+t+b-1)/2}^{2-k} \big[n-k+t+b = 1 \pmod 2\big] + \mathrm{C}_{(2-k+t+b)/2}^{2-k} \big[n-k+t+b = 0 \pmod 2\big] \Big) = \mathrm{C}_2^0 3^{-2} (\mathrm{C}_2^0 2^0 \mathrm{C}_2^2 + \mathrm{C}_2^1 2^1 \mathrm{C}_2^2 + \mathrm{C}_2^2 2^2 \mathrm{C}_3^2) + \mathrm{C}_2^1 3^{-1} (\mathrm{C}_1^0 2^0 \mathrm{C}_1^1 + \mathrm{C}_1^1 2^1 \mathrm{C}_2^1) + \mathrm{C}_2^2 3^0 (\mathrm{C}_0^0 2^0 \mathrm{C}_2^2) = 56/9.$$

Таким образом, используя полученные формулы, можно оценить количество возможных решений в зависимости от размера множества предметов и ограничений на их вес. Данные формулы могут быть полезны для выбора оптимального подхода к решению задач, определения вероятности их успешного решения, а также для исследования свойств задач и оптимизации процесса поиска их решений.

Заключение

Рассмотрены методы вычисления и оценки количества допустимых решений задачи об ограниченном рюкзаке. Исследование основано на анализе комбинаторных свойств задачи, что позволило получить новые формулы для вычисления среднего числа допустимых решений для случаев $x \in \{0,1\}^n$ и $x \in \{0,1,2\}^n$, где компоненты вектора весов $(a_1, \ldots, a_n)$ принадлежат диапазону от 0 до b. В частности, в одном из общих случаев найдена производящая функция, определяющая объём множества допустимых решений для задач размерности n, в которых компоненты вектора весов принимают значения в пределах от 0 до фиксированного числа p.

Полученные результаты могут стать основой для дальнейшего изучения свойств множества допустимых решений задачи о рюкзаке. Найденные формулы также могут применяться в вычислительных процедурах для оценки оптимальности алгоритмов решения различных типов подобных задач, а также в декомпозиционных и эвристических алгоритмах их решения.

ЛИТЕРАТУРА

1. *Kellerer H., Pferschy U., and Pisinger D.* Knapsack Problems. Berlin: Springer, 2004. 548 p.
2. *Martello S. and Toth P.* Knapsack Problems: Algorithms and Computer Implementations. N.Y.: John Wiley & Sons, 1990. 308 p.
3. *Колпаков Р. М., Посыпкин М. А.* Верхняя и нижняя оценки трудоемкости метода ветвей и границ для задачи о ранце // Дискретная математика. 2010. Т. 22. Вып. 1. С. 58–73.
4. *Колпаков Р. М., Посыпкин М. А., Си Ту Тант Син.* Верхняя оценка сложности одного из вариантов метода ветвей и границ для задачи о сумме подмножеств // Intern. J. Open Inform. Technol. 2016. V. 4. No. 2. P. 1–6.
5. *Колпаков Р. М., Посыпкин М. А., Сигал И. Х.* О нижней оценке вычислительной сложности одной параллельной реализации метода ветвей и границ // Автоматика и телемеханика. 2010. № 10. С. 156–166.
6. *Колпаков Р. М., Посыпкин М. А.* О масштабируемости и эффективности одного метода решения задачи о ранце в распределенной вычислительной среде // Труды ИСА РАН. 2009. Т. 46. С. 164–174.

7. *Колпаков Р. М., Посыпкин М. А.* Об эффективной стратегии распараллеливания при решении задач о сумме подмножеств методом ветвей и границ // Дискретная математика. 2019. Т. 31. № 4. С. 20–37.
8. *Колпаков Р. М., Посыпкин М. А., Си Ту Тант Син.* Сложность решения задачи о сумме подмножеств методом ветвей и границ с доминированием и мощностным отсевом // Автоматика и телемеханика. 2017. № 3. С. 96–110.
9. *Колпаков Р. М., Посыпкин М. А.* Асимптотическая оценка сложности метода ветвей и границ с ветвлением по дробной переменной для задачи о ранце // Дискретн. анализ исслед. опер. 2008. Т. 15. № 1. С. 58–81.
10. *Колпаков Р. М., Посыпкин М. А.* О наилучшем выборе переменной ветвления в задаче о сумме подмножеств // Дискретная математика. 2017. Т. 29. № 1. С. 51–58.
11. *Колпаков Р. М.* Оптимальная стратегия решения частного случая задачи о ранце методом ветвей и границ // Вестник Московского университета. Сер. 1. Математика. Механика. 2021. № 3. С. 13–22.
12. *Егорычев Г. П.* Интегральное представление и вычисление комбинаторных сумм. Новосибирск: Наука, 1977. 285 с.
13. *Леонтьев В. К., Гордеев Э. Н.* Производящие функции в задаче о ранце // Доклады АН. 2018. Т. 481. № 5. С. 478–480.
14. *Гордеев Э. Н., Леонтьев В. К.* О некоторых комбинаторных свойствах задачи о рюкзаке // Ж. вычисл. матем. и матем. физ. 2019. Т. 59. № 8. С. 1439–1447.
15. *Леонтьев В. К., Гордеев Э. Н.* О числе решений системы булевых уравнений // Автоматика и телемеханика. 2021. № 9. С. 150–168.
16. *Волков М. С. А.* Комбинаторные свойства задачи об ограниченном рюкзаке // Прикладная дискретная математика. 2024. № 63. С. 117–130.
17. *Леонтьев В. К., Гордеев Э. Н.* Зависимость среднего числа решений в задаче о ранце от параметров области ограничений // Безопасные информационные технологии: Сб. трудов 11-й Междунар. науч.-технич. конф. М.: МГТУ им. Н. Э. Баумана, 2021. С. 85–90.
18. *Кнут Д. Э.* Искусство программирования. Т. 1: Основные алгоритмы. 3-е изд. М.: Издательский дом «Вильямс», 2002. 720 с.

REFERENCES

1. *Kellerer H., Pferschy U., and Pisinger D.* Knapsack Problems. Berlin, Springer, 2004. 548 p.
2. *Martello S. and Toth P.* Knapsack Problems: Algorithms and Computer Implementations. N.Y., John Wiley & Sons, 1990. 308 p.
3. *Kolpakov R. M. and Posypkin M. A.* Upper and lower bounds for the complexity of the branch and bound method for the knapsack problem. Discrete Math. Appl., 2010, vol. 20, no. 1, pp. 95–112.
4. *Kolpakov R. M., Posypkin M. A., and Si Tu Tant Sin.* Verkhnyaya otsenka slozhnosti odnogo iz variantov metoda vetvey i granits dlya zadachi o summe podmnozhestv [Upper bound for the complexity of one of the variants of the branch and bound method for the subset sum problem]. Intern. J. Open Inform. Technol., 2016, vol. 4, no. 2, pp. 1–6. (in Russian)
5. *Kolpakov R. M., Posypkin M. A., and Sigal I. K.* On a lower bound on the computational complexity of a parallel implementation of the branch-and-bound method. Autom. Remote Control, 2010, vol. 71, no. 10, pp. 2152–2161.
6. *Kolpakov R. M. and Posypkin M. A.* O masshtabiruemosti i effektivnosti odnogo metoda resheniya zadachi o rantse v raspredelennoy vychislitel'noy srede [On the scalability and efficiency of a method for solving the knapsack problem in a distributed computing environment]. Proc. ISA RAN, 2009, vol. 46, pp. 164–174. (in Russian)

7. *Kolpakov R. M. and Posypkin M. A.* Effective parallelization strategy for the solution of subset sum problems by the branch-and-bound method. Discrete Math. Appl., 2020, vol. 30, no. 5, pp. 313–325.
8. *Kolpakov R. M., Posypkin M. A., and Sin S. T. T.* Complexity of solving the Subset Sum problem with the branch-and-bound method with domination and cardinality filtering. Autom. Remote Control, 2017, vol. 78, no. 3, pp. 463–474.
9. *Kolpakov R. M. and Posypkin M. A.* Asimptoticheskaya otsenka slozhnosti metoda vetvey i granits s vetvleniem po drobnoy peremennoy dlya zadachi o rantse [Asymptotic estimate on the complexity of the branch-and-bound method with branching by a fractional variable for the knapsack problem]. Diskretnyi Analiz i Issledovanie Operatsii, 2008, vol. 15, no. 1, pp. 58–81. (in Russian)
10. *Kolpakov R. M. and Posypkin M. A.* On the best choice of a branching variable in the subset sum problem. Discrete Math. Appl., 2018, vol. 28, no. 1, pp. 29–34.
11. *Kolpakov R. M.* Optimal strategy for solving a special case of the knapsack problem by the branch and bound method. Moscow University Mathematics Bulletin, 2021, vol. 76, no. 3, pp. 97–106.
12. *Egorychev G. P.* Integral'noe predstavlenie i vychislenie kombinatornykh summ [Integral Representation and the Computation of Combinatorial Sums]. Novosibirsk, Nauka, 1977. 285 p. (in Russian)
13. *Leont'ev V. K. and Gordeev E. N.* Proizvodyashchie funktsii v zadache o rantse [The generating functions in the knapsack problem]. Doklady Akademii Nauk, 2018, vol. 481, no. 5, pp. 478–480. (in Russian)
14. *Gordeev E. N. and Leont'ev V. K.* On combinatorial properties of the knapsack problem. Comput. Math. Math. Phys., 2019, vol. 59, no. 8, pp. 1380–1388.
15. *Leontiev V. K. and Gordeev E. N.* On the number of solutions to a system of Boolean equations. Autom. Remote Control, 2021, vol. 82, no. 9, pp. 1581–1596.
16. *Volkov M. S. A.* Kombinatornye svoystva zadachi ob ogranichennom ryukzake [Combinatorial properties of the bounded knapsack problem]. Prikladnaya Diskretnaya Matematika, 2024, no. 63, pp. 117–130. (in Russian)
17. *Leont'ev V. K. and Gordeev E. N.* Zavisimost' srednego chisla resheniy v zadache o rantse ot parametrov oblasti ogranicheniy [Dependence of the average number of solutions in the knapsack problem on the parameters of the constraint domain]. Proc. 11th Intern. Conf. "Secure Information Technologies", Moscow, Bauman Moscow Technical University, 2021, pp. 85–90. (in Russian)
18. *Knuth D. E.* The Art of Computer Programming. Vol. 1: Fundamental Algorithms. Third Edition. Massachusetts, Addison-Wesley, 1997. 672 p.

УДК 519.8 DOI 10.17223/20710410/68/8

ПРИБЛИЖЁННОЕ РЕШЕНИЕ МАКСИМИННОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ ОБЪЕКТОВ НА СЕТИ С ОГРАНИЧЕНИЯМИ НА МИНИМАЛЬНЫЕ РАССТОЯНИЯ[1]

Г. Г. Забудский

Институт математики им. С. Л. Соболева СО РАН, г. Новосибирск, Россия

E-mail: zabudsky@ofim.oscsbras.ru

Рассматривается задача оптимального размещения объектов на неориентированной взвешенной сети, расположенной на плоскости. Вершинам приписаны положительные веса, а рёбра представлены отрезками. Вес вершины отражает требование размещать объекты как можно дальше от неё. Заданы ограничения на минимально допустимые расстояния от вершин до объектов. Необходимо найти такие точки на рёбрах сети для размещения объектов, чтобы минимальное взвешенное расстояние от вершин до объектов было максимальным. Предложен алгоритм решения задачи с заданной точностью для двух объектов.

Ключевые слова: *выпуклая оболочка, задача размещения, максиминный критерий, опасный объект, сеть.*

APPROXIMATE SOLUTION OF THE MAXIMIN PROBLEM OF LOCATING FACILITIES ON A NETWORK WITH CONSTRAINTS ON MINIMUM DISTANCES

G. G. Zabudsky

Sobolev Institute of Mathematics, Novosibirsk, Russia

We consider the problem of the optimal location of facilities on an undirected weighted network located on a plane. The vertices are assigned positive weights and the edges are segments. The weight of a vertex reflects the requirement to locate the facilities as far away from it as possible. Constraints are given on the minimum admissible distances from vertices to the facilities. It is necessary to find such points on the edges of the network to locate the facilities that the minimum weighted distance from the vertices to the facilities is maximum. An algorithm for solving the problem with a given accuracy for two facilities is proposed.

Keywords: *convex hull, location problem, maximin criterion, obnoxious facility, network.*

Введение

Задачи оптимального размещения объектов различного назначения имеют много практических приложений. В общем случае задача заключается в размещении одного или нескольких объектов в заданной области с фиксированными в ней объектами (клиентами) таким образом, чтобы оптимальной была некоторая функция (функции)

[1]Работа выполнена в рамках госзадания ИМ СО РАН, проект № FWNF-2022-0020.

расстояний между клиентами и размещаемыми объектами. Обзор исследований задач оптимального размещения на сетях и плоскости можно найти в [1–5].

В теории оптимального размещения наиболее исследованы задачи, в которых объекты должны быть расположены как можно ближе к клиентам. Такие объекты называют желательными, например поликлиники, пожарные части, магазины. Достаточно хорошо изучены задачи с критериями минимизации максимального расстояния (задачи о центрах) и суммарного расстояния (задачи о медианах) от клиентов до объектов [1].

В последние годы в связи с возросшими экологическими требованиями проводятся исследования по проблемам размещения нежелательных (опасных) объектов. Они обслуживают население, но оказывают негативное влияние на него. Предполагается, что влияние уменьшается по мере увеличения расстояния до объектов. Поэтому необходимо размещать такие объекты как можно дальше от населения. При этом можно минимизировать негативное влияние на наиболее пострадавшее население или среднее влияние на всё население района. В первом случае максимизируется минимальное расстояние (максиминная задача), а во втором — суммарное расстояние (максисуммная задача) от клиентов до объектов [5–8].

На практике можно выделить следующие ситуации, в которых необходимо учитывать негативное влияние при размещении объектов:

1) угроза общественной безопасности или нарушение комфорта людей (исправительный центр);
2) ущерб окружающей среде и здоровью населения (химический завод);
3) требование чистой и здоровой среды (санаторий).

Решение задач размещения опасных объектов на сетях с расстояниями, измеряемыми по транспортной сети, может быть использовано при учёте влияния первого типа. Обоснованием этого является то, что чем больше расстояние от населённых пунктов до объектов по транспортной сети, тем они безопаснее и тем меньше неудобств доставляют обществу [6–8]. Для объектов второго и третьего типов более реальным будет применение, например, евклидовой метрики [9, 10]. Так, в случае химического завода загрязнение распространяется не по транспортной сети. Общим для всех типов объектов является то, что они не должны располагаться вблизи густонаселённых районов.

Основная часть исследований задач размещения опасных объектов посвящена вариантам размещения одного объекта. Полиномиальные алгоритмы для задач на специальных сетях и общего вида предложены в [6–8].

Небольшое количество работ посвящены задачам размещения нескольких опасных объектов на сетях. В [11] представлены результаты исследования сложности их решения. Для общего случая доказано, что задачи NP-трудные, даже если сеть состоит из одного ребра. Нахождение 2/3-приближённого решения для максиминной задачи также является NP-трудным. Эвристические алгоритмы для решения задач предложены, например, в [12, 13].

В данной работе рассматривается максиминная задача размещения объектов на сети, расположенной на плоскости. Вершины сети соответствуют клиентам, а рёбра — дорогам. Вершины имеют положительные веса. Рёбра представлены отрезками с длинами в евклидовой метрике. Заданы ограничения на минимально допустимые расстояния от клиентов до объектов. Необходимо найти такое размещение объектов на сети, чтобы минимальное расстояние от них до ближайшего клиента было максимальным.

Предложен алгоритм решения задачи с заданной точностью для двух объектов. Исходная непрерывная задача решается с помощью серии дискретных задач.

1. Постановка задачи

Задана область на плоскости с населёнными пунктами, соединёнными сетью дорог, и объекты, например мусороперерабатывающие заводы, которые необходимо разместить на дорожной сети. Объекты оказывают негативное влияние на население. Влияние уменьшается с увеличением расстояния от населённых пунктов до объектов. Заданы минимально допустимые расстояния от населённых пунктов до объектов (санитарные зоны), в которых нельзя размещать объекты. Кроме того, между объектами также определено минимальное расстояние, чтобы избежать суммарного негативного влияния от них. Необходимо найти такое размещение объектов, чтобы были выполнены ограничения по минимальным расстояниям и негативное влияние было минимальным. В качестве критерия рассматривается минимизация влияния на наиболее пострадавшее население. Поэтому максимизируется минимальное расстояние от населённых пунктов до ближайшего объекта.

Введём обозначения и сформулируем математическую модель. Пусть $G = (V, E)$ — неориентированная сеть, соответствующая населенным пунктам и дорогам; $V = \{v_1, \ldots, v_n\}$ — множество вершин, $I = \{1, \ldots, n\}$ — множество их номеров. Для каждой вершины v_i заданы координаты (a_i, b_i) и вес $\alpha_i > 0$, $i \in I$. Если $\alpha_i < \alpha_j$, то объекты должны размещаться дальше от вершины v_i, чем от вершины v_j. Вес вершины может быть величиной, обратной количеству населения в пункте, соответствующем вершине. Рёбра сети $E = \{e_1, \ldots, e_m\}$ с множеством номеров $J = \{1, \ldots, m\}$ представлены отрезками, длины которых определяются в евклидовой метрике $\rho(v_i, v_j)$, $i \neq j$, $i, j \in I$. Множества точек размещения объектов и их номеров обозначим через $z = \{z_1, z_2, \ldots, z_p\}$ и $P = \{1, \ldots, p\}$ соответственно. Обозначим через d_i, $i \in I$, и d минимально допустимое расстояние между вершиной v_i и объектами и объектов между собой соответственно.

Множество возможных точек размещения объектов на сети (точки на рёбрах и вершины) будем обозначать как $Z(G)$. Положение объекта на ребре определяется расстоянием от его вершин. Например, точка x размещена на ребре (v_i, v_j) на расстоянии $\rho(v_i, x) = \lambda\rho(v_i, v_j)$ от вершины v_i и на расстоянии $\rho(v_j, x) = (1 - \lambda)\rho(v_i, v_j)$ от вершины v_j, где $0 \leqslant \lambda \leqslant 1$.

Математическая модель максиминной задачи размещения на сети имеет вид

$$\min_{i \in I} \min_{j \in P} \alpha_i \rho(v_i, z_j) \to \max; \tag{1}$$

$$\rho(v_i, z_j) \geqslant d_i,\ i \in I,\ j \in P; \tag{2}$$

$$\rho(z_i, z_j) \geqslant d,\ i, j \in P,\ i \neq j; \tag{3}$$

$$z \subseteq Z(G). \tag{4}$$

Максиминная задача размещения одного объекта на сети, в которой расстояния измеряются по кратчайшим путям, рассматривается, например, в [8]. Для сети общего вида предложен полиномиальный алгоритм решения. Алгоритм основан на поиске узких рёберных точек, аналогично задаче размещения объекта на сети с максимизацией суммарного расстояния от вершин до объекта [6, 7]. В [10] предложен алгоритм поиска приближённого решения максиминной задачи для одного объекта на сети, расположенной на плоскости. Двухкритериальная задача размещения объекта на сети дорог с максисуммным и максиминным критериями рассмотрена в [9].

Далее рассмотрим вариант максиминной задачи для размещения двух объектов — z_1 и z_2.

2. Область допустимых решений задачи (1)–(4)

Проверка существования допустимого решения задачи (1)–(4) включает два этапа. На первом этапе находится область S, в которой выполняются ограничения на минимально допустимые расстояния между вершинами и объектами. На втором этапе проверяется возможность размещения объектов в области S с ограничением (3).

2.1. Этап 1

Область S последовательно определяется на рёбрах сети. Опишем алгоритм для произвольного ребра (v_h, v_q).

Шаг 1. От исходной системы координат переходим к системе, в которой ребро будет расположено на оси абсцисс. Вершина v_h — в начале координат, а вершина v_q — на расстоянии $\rho(v_h, v_q)$ от начала координат.

Шаг 2. Для каждой вершины находим отрезки на ребре, в которых выполняются ограничения (2).

Шаг 3. Находим множество непересекающихся отрезков.

Шаг 4. Определяем область S в исходной системе координат.

На шаге 1 обозначим координаты вершины v_i в новой системе как (a'_i, b'_i), $i \in I$, которые определяются следующим образом:

$$a'_i = (a_i - a_h)\cos\varphi + (b_i - b_h)\sin\varphi,$$
$$b'_i = -(a_i - a_h)\sin\varphi + (b_i - b_h)\cos\varphi,$$

где φ — угол наклона прямой, проведённой через вершины v_h и v_q (точки с координатами (a_h, b_h) и (a_q, b_q)). Вершины v_h и v_q в новой системе имеют координаты $(0, 0)$ и $(\rho(v_h, v_q), 0)$ соответственно.

На шаге 2 для текущей вершины v_k определяем расстояние ρ_k от неё до ребра (v_h, v_q). Значение ρ_k вычисляется следующим образом:

$$\rho_k = \begin{cases} b'_k, & 0 \leqslant a'_k \leqslant \rho(v_h, v_q), \\ \sqrt{a'^2_k + b'^2_k}, & a'_k < 0, \\ \sqrt{(a'_k - \rho(v_h, v_q))^2 + b'^2_k}, & a'_k > \rho(v_h, v_q). \end{cases}$$

Если $\rho_k > d_k$, то переходим к другой вершине. В противном случае для $y = 0$ решаем уравнение

$$\sqrt{(x - a'_k)^2 + (y - b'_k)^2} = d_k.$$

Если r_1^k, r_2^k — действительные корни уравнения, то область на ребре, в которой не выполняется ограничение (2) относительно вершины v_k, представляет интервал (r_1^k, r_2^k). Область, в которой выполняется ограничение, образована объединением двух отрезков $[0, r_1^k]$ и $[r_2^k, \rho(v_h, v_q)]$. После просмотра всех вершин сети получаем набор не более $2n$ отрезков на ребре (v_h, v_q), в которых выполняются ограничения (2) для всех вершин. Обозначим их как $[s_1^i, s_2^i]$, $i \in I^{2n} = \{1, \ldots, 2n\}$. Отрезки могут пересекаться. Если объединение интервалов, в которых не выполняются ограничения (2), покрывает всё ребро, то оно не принадлежит области S.

Опишем алгоритм построения непересекающихся отрезков области S на ребре (v_h, v_q) (шаг 3). Пусть отрезки перенумерованы так, что имеют место неравенства

$$s_1^1 < s_1^2 \ldots < s_1^{2n}.$$

Формируем множество номеров отрезков, левая граница которых приналежит $[s_1^1, s_2^1]$:

$$I_1^1 = \{i \in I^{2n} : s_1^1 \leqslant s_1^i \leqslant s_2^1\}.$$

Если $I_1^1 = \varnothing$, полагаем $o_1^1 = s_1^1$ и $o_2^1 = s_2^1$. Иначе находим номер i_1, такой, что

$$\max_{i \in I^{2n}} s_2^i = s_2^{i_1}.$$

Формируем множество номеров отрезков I_2^1, $I_2^1 \cap I_1^1 = \varnothing$. Полагаем $I^{2n} = I^{2n} \setminus I_1^1$,

$$I_2^1 = \{i \in I^{2n} : s_2^1 \leqslant s_1^i \leqslant s_2^{i_1}\}.$$

Если $I_2^1 = \varnothing$, то полагаем $o_1^1 = s_1^1$ и $o_2^1 = s_2^{i_1}$. Иначе находим номер i_2, такой, что

$$\max_{i \in I_1^2} s_2^i = s_2^{i_2}.$$

Формируем множество I_3^1 и так далее. В итоге получим первый отрезок $[o_1^1, o_2^1]$ области S на ребре, который не пересекается с другими отрезками.

Для построения следующего отрезка находим минимальный номер k, для которого $s_1^k > o_2^1$. Формируем множество номеров отрезков

$$I_1^2 = \{i \in I^{2n} : s_1^k \leqslant s_1^i \leqslant s_2^k\}$$

и повторяем процесс построения отрезка.

В результате находим область S на ребре (v_h, v_q) в виде множества непересекающихся отрезков $[o_1^1, o_2^1], [o_1^2, o_2^2], \ldots, [o_1^k, o_2^k]$, $k \leqslant 2n$. Трудоёмкость построения непересекающихся отрезков не превосходит $O(n)$.

На шаге 4 определяем координаты границ отрезков области S на ребре (v_h, v_q) в исходной системе координат. Для отрезка с номером i имеют место следующие формулы:

$$x_1^i = a_h + o_1^i \cos\varphi, \quad y_1^i = b_h + o_1^i \sin\varphi,$$
$$x_2^i = a_h + o_2^i \cos\varphi, \quad y_2^i = b_h + o_2^i \sin\varphi.$$

После выполнения шагов 1–4 для всех рёбер сети G получим набор $O(n^3)$ отрезков области S: $[(x_1^i, y_1^i), (x_2^i, y_2^i)]$, $i \in I^{n^3} = \{1, \ldots, n^3\}$.

2.2. Э т а п 2

На этом этапе проверяется возможность размещения объектов z_1 и z_2 в области S на расстоянии не менее d друг от друга.

Утверждение 1. Максимальное расстояние между двумя отрезками на плоскости достигается в их граничных точках.

Доказательство. Максимальное расстояние между точкой t и отрезком достигается в одной из граничных точек отрезка. Это следует из того, что можно считать, что точка t находится на оси ординат, а отрезок — на оси абсцисс. Расстояние от точки t до любой точки s отрезка — это диагональ в треугольнике, один катет которого — ордината точки t, он общий для всех точек отрезка, а другой — координата точки s. Диагональ треугольника максимальна, когда второй катет имеет максимальную длину. Это достигается в одной из граничных точках отрезка. Фиксируя точку в одном из концов одного отрезка, аналогично можно показать, что максимальное расстояние от неё до другого отрезка достигается в его граничной точке. ■

В рассуждениях можно было использовать свойство, что максимальное расстояние между точкой на плоскости и выпуклым многоугольником достигается в одной из вершин многоугольника [3].

Обозначим через $S1$ множество граничных точек отрезков области S, ближайщих к вершинам соответствующих рёбер.

Следствие 1. Максимальное расстояние между точками множества S достигается в точках множества $S1$.

Доказательство. Отрезки области S на ребре вложены в отрезок с граничными точками, ближайшими к вершинам ребра. Максимальное расстояние между двумя такими отрезками для различных рёбер сети достигается в их граничных точках, т. е. в точках множества $S1$. ■

Если число отрезков области S на ребре (v_p, v_q) равно k, то для выполнения этапа 2 достаточно рассматривать две точки с координатами (x_1^1, y_1^1) и (x_2^k, y_2^k). В множестве $S1$, состоящем из $O(n^2)$ точек, необходимо найти наиболее удалённые друг от друга точки — диаметр множества $S1$. Если диаметр $S1$ больше либо равен d, то исходная задача имеет допустимое решение. Нахождение диаметра множества точек перебором пар имеет трудоёмкость $O(n^4)$.

В работах [14, 15] описан алгоритм нахождения диаметра множества из k точек на плоскости. Алгоритм основан на том, что диаметр множества точек равен диаметру их выпуклой оболочки. Алгоритм построения выпуклой оболочки k точек имеет трудоёмкость $O(k \log k)$ [16]. Диаметр выпуклой оболочки находится за линейное от количества точек время. Поэтому сложность определения диаметра множества $S1$ оценивается как $O(n^2 \log n)$ операций.

Приведём необходимые понятия и кратко опишем эффективный алгоритм нахождения диаметра множества точек на плоскости при условии, что выпуклая оболочка построена [14, 15]. Алгоритм построения выпуклой оболочки широко известен и его можно найти, например, в [16].

Определение 1. Опорной прямой выпуклого многоугольника называют прямую, проходящую через его вершину и обладающую тем свойством, что многоугольник лежит по одну сторону от неё.

Определение 2. Пара точек многоугольника, через которые можно провести параллельные опорные прямые, называется противолежащей парой.

Теорема 1. Диаметр выпуклой фигуры равен наибольшему из расстояний между двумя параллельными опорными прямыми этой фигуры.

Из теоремы 1 следует, что для нахождения диаметра множества точек необходимо рассматривать только противолежащие пары. Сделать это можно за линейное от количества точек $O(n^2)$ в множестве $S1$ время с помощью метода, который называется «вращающиеся калиперы» (англ. rotating calipers).

На рис. 1 опорные параллельные прямые L и M проведены через вершины A и D, эти вершины образуют противолежащую пару. Если вращать прямые L и M против часовой стрелки вокруг вершин, они будут опорными до тех пор, пока одна из них не совпадёт со стороной многоугольника. Угол поворота до ребра (E, D) меньше, чем до ребра (A, B), поэтому вершины A и E будут следующей противолежащей парой. Далее M будет вращаться вокруг вершины E. Продолжая процесс вращения, получим все пары противолежащих вершин. При этом для получения новой противолежащей

пары необходимо сравнить углы между опорными прямыми и рёбрами многогранника. Трудоёмкость нахождения всех противолежащих пар для $S1$ составляет $O(n^2)$.

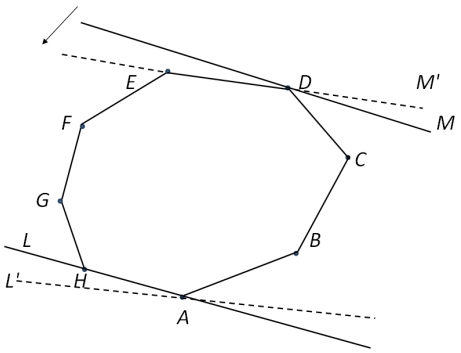


Рис. 1. Формирование противолежащих пар

3. Алгоритм решения задачи (1)–(4)

Модель (1)–(4) для двух объектов может быть представлена следующим образом:

$$T \to \max; \tag{5}$$

$$\rho(v_i, z_j) \geqslant \max_{i \in I}(d_i, T/\alpha_i),\ i \in I,\ j = 1, 2; \tag{6}$$

$$\rho(z_1, z_2) \geqslant d; \tag{7}$$

$$z \subset Z(G). \tag{8}$$

Идея алгоритма нахождения приближённого решения задачи (5)–(8) состоит в переборе значений параметра T и проверке существования допустимого решения для них. Через $S(T)$ обозначим область, в которой выполняются ограничения (6) для фиксированного T. Область $S(T)$ находится аналогично области S. В этом случае минимальное расстояние от вершины сети v_i до объектов равно $\max(d_i, T/\alpha_i)$ для $i \in I$. Значения T выбираются из определённого отрезка с применением метода дихотомии. Аналогично $S1$ определим область $S1(T)$. Для очередного значения T решается вспомогательная задача — задача распознавания (ЗР) в области $S1(T)$.

Задача 1 (ЗР). Можно или нет разместить два объекта z_1 и z_2 в области $S1(T)$ так, чтобы выполнялось ограничение на минимальное расстояние между ними?

Если в результате решения ЗР получаем ответ «да», то на следующей итерации происходит увеличение недопустимых для размещения областей пропорционально весам вершин сети. Если ответ «нет» — пропорциональное уменьшение. Это происходит до тех пор, пока не найдётся допустимое значение параметра T, удовлетворяющее заданной точности решения задачи.

Определим отрезок, содержащий оптимальное значение T^* параметра T. Вычислим значения $H = \max\limits_{i \in I} a_i - \min\limits_{i \in I} a_i$ и $U = \max\limits_{i \in I} b_i - \min\limits_{i \in I} b_i$.

Утверждение 2. Если $S1(T) \neq \varnothing$, то для оптимального значения T^* справедливы следующие неравенства:

$$\min_{i \in I} \alpha_i d_i \leqslant T^* \leqslant \max_{i \in I} \alpha_i \sqrt{H^2 + U^2}.$$

Доказательство. Справедливость левого неравенства следует из того, что рассматривается задача максимизации и для неё существует допустимое решение.

Поэтому $T^* \geqslant \min\limits_{i\in I} \alpha_i d_i$. Правое неравенство следует из того, что $\max(d_i, T/\alpha_i) \leqslant$ $\leqslant \alpha_i \max\limits_{j=1,2} \rho(v_i, z_j) \leqslant \max\limits_{i\in I} \alpha_i \sqrt{H^2+U^2}$. ■

Замечание 1. При решении ЗР находится размещение z_1 и z_2, так как диаметр множества определяется с помощью нахождения вершин выпуклой оболочки, на которых он достигается.

Приведём описание алгоритма решения ЗР для фиксированного T_k.

Обозначим через $O_k = [l_k, r_k]$ отрезок, рассматриваемый на шаге k, где $l_1 = \min\limits_{i\in I}(\alpha_i d_i)$, $r_1 = \max\limits_{i\in I} \alpha_i \sqrt{H^2+U^2}$. На шаге k выбирается T_k — середина отрезка O_k и решается ЗР. Если ответ в ЗР «да», то полагаем $O_{k+1} = [T_k, r_k]$, иначе — $O_{k+1} = [l_k, T_k]$.

Количество итераций (решения ЗР) равно $\log\left((r_1 - l_1)/\varepsilon\right)$, где ε — точность, с которой находится решение. Трудоёмкость решения ЗР оценивается как $O(n^2 \log n)$ операций. Общая трудоёмкость алгоритма решения исходной задачи не превышает $O\big(n^2 \log n \log\left((r_1 - l_1)/\varepsilon\right)\big)$.

Заключение

Рассмотрена максиминная задача размещения двух объектов на сети, расположенной на плоскости. Заданы положительные веса вершин и минимальные расстояния от вершин до объектов. Предложен алгоритм нахождения приближённого решения задачи с заданной точностью.

Сетью может быть транспортная сеть, соединяющая населенные пункты. Объекты должны быть размещены так, чтобы их негативное влияние на наиболее пострадавшее население было минимальным. Решение задачи может быть полезным при выборе мест расположения, например, мусороперерабатывающих предприятий.

ЛИТЕРАТУРА

1. *Кристофидес Н.* Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
2. *Drezner Z.* Facility Location. A Survey of Applications and Methods. N.Y.: Springer, 1995. 571 p.
3. *Nickel S.* Location Theory. A Unified Approach. Berlin: Springer Verlag, 2005. 437 p.
4. *Farani R. Z. and Hekmatfar M.* Facility Location: Concepts, Models, Algorithms and Case Studies. Berlin: Springer Verlag, 2009. 549 p.
5. *Eiselt H. A. and Marianov V.* Foundations of Location Analysis. N.Y.: Springer, 2011. 509 p.
6. *Church R. L. and Garfinkel R. S.* Locating an obnoxious facility on a network // Trans. Sci. 1978. V. 12. No. 2. P. 107–118.
7. *Забудский Г. Г.* Решение макси-суммной задачи размещения на сети с ограничениями на транспортные затраты // Прикладная дискретная математика. 2023. № 60. С. 120–127.
8. *Melachrinoudis E. and Zhang G.* An $O(mn)$ algorithm for the 1-maximin problem on a network // Comput. & Oper. Res. 1999. V. 26. No. 9. P. 849–869.
9. *Heydari R. and Melachrinoudis E.* Location of a semi-obnoxious facility with elliptic maxmin and network minisum objectives // Eur. J. Oper. Res. 2012. V. 223. No. 2. P. 452–460.
10. *Zabudsky G. and Lisina M.* Approximately algorithm for maximin location problem on network // Proc. XII Intern. Conf. "Dynamics of Systems, Mechanisms and Machines". 13–15 November 2018, Omsk, Russia. P. 1–6.
11. *Tamir A.* Obnoxious facility location on graphs // SIAM J. Discrete Math. 1991. V. 4. No. 4. P. 550–567.

12. *Welch S. B. and Wesolowsky S.* The obnoxious p facility network location problem with facility interaction // Eur. J. Oper. Res. 1997. V. 102. No. 2. P. 302–319.
13. *Tamir A.* Locating two obnoxious facilities using the weighted maximin criterion // Oper. Res. Let. 2006. V. 34. No. 1. P. 97–105.
14. *Яглом И. М., Болтянский И. М.* Выпуклые фигуры. М.: Технико-теоретическая литература, 1951. 344 с.
15. *Shamos M. I.* Computational Geometry. PhD Thesis. New Haven, Yele University, 1978. 236 p.
16. *Прерата Ф., Шеймос М.* Вычислительная геометрия: Введение. М.: Мир, 1989. 478 с.

REFERENCES

1. *Christofides N.* Graph Theory: An algorithmic approach. N.Y., Academic Press, 1975. 400 p.
2. *Drezner Z.* Facility Location. A Survey of Applications and Methods. N.Y., Springer, 1995. 571 p.
3. *Nickel S.* Location Theory. A Unified Approach. Berlin, Springer Verlag, 2005. 437 p.
4. *Farani R. Z. and Hekmatfar M.* Facility Location: Concepts, Models, Algorithms and Case Studies. Berlin, Springer Verlag, 2009. 549 p.
5. *Eiselt H. A. and Marianov V.* Foundations of Location Analysis. N.Y., Springer, 2011. 509 p.
6. *Church R. L. and Garfinkel R. S.* Locating an obnoxious facility on a network. Trans. Sci., 1978, vol. 12, no. 2, pp. 107–118.
7. *Zabudskiy G. G.* Reshenie maksi-summnoy zadachi razmeshcheniya na seti s ogranicheniyami na transportnye zatraty [Solving of the maxisum location problem on network with a restriction on transport costs]. Prikladnaya Diskretnaya Matematika, 2023, no. 60, pp. 120–127. (in Russian)
8. *Melachrinoudis E. and Zhang G.* An $O(mn)$ algorithm for the 1-maximin problem on a network. Comput. & Oper. Res., 1999, vol. 26, no. 9, pp. 849–869.
9. *Heydari R. and Melachrinoudis E.* Location of a semi-obnoxious facility with elliptic maxmin and network minisum objectives. Eur. J. Oper. Res., 2012, vol. 223, no. 2, pp. 452–460.
10. *Zabudsky G. and Lisina M.* Approximately algorithm for maximin location problem on network. Proc. XII Intern. Conf. "Dynamics of Systems, Mechanisms and Machines", 13–15 November 2018, Omsk, Russia, pp. 1–6.
11. *Tamir A.* Obnoxious facility location on graphs. SIAM J. Discrete Math., 1991, vol. 4, no. 4, pp. 550–567.
12. *Welch S. B. and Wesolowsky S.* The obnoxious p facility network location problem with facility interaction. Eur. J. Oper. Res., 1997, vol. 102, no. 2, pp. 302–319.
13. *Tamir A.* Locating two obnoxious facilities using the weighted maximin criterion. Oper. Res. Let., 2006, vol. 34, no. 1, pp. 97–105.
14. *Yaglom I. M. and Boltyanskiy I. M.* Vypuklye figury [Convex Figures]. Moscow, Tekhniko-teoreticheskaya literatura, 1951. 344 p. (in Russian)
15. *Shamos M. I.* Computational Geometry. PhD Thesis. New Haven, Yele University, 1978. 236 p.
16. *Prerarata F. and Sheimos M.* Computational Geometry: Introduction. N.Y., Springer Verlag, 1985. 478 p.

СВЕДЕНИЯ ОБ АВТОРАХ

БАРОТОВ Достонжон Нумонжонович — старший преподаватель кафедры математики и анализа данных Финансового университета при Правительстве РФ, г. Москва. E-mail: **DNBarotov@fa.ru**

БАРОТОВ Рузибой Нумонжонович — преподаватель кафедры математического анализа им. профессора А. Мухсинова Худжандского государственного университета им. акад. Б. Гафурова, г. Худжанд. E-mail: **ruzmet.tj@mail.ru**

ВОЛКОВ Мария Сабина Александровна — аспирантка Московского государственного технического университета им. Н. Э. Баумана, г. Москва.
E-mail: **sabina-volkoff@yandex.ru**

ГОРДЕЕВ Эдуард Николаевич — доктор физико-математических наук, профессор Московского государственного технического университета им. Н. Э. Баумана, г. Москва. E-mail: **werhorn@yandex.ru**

ЗАБУДСКИЙ Геннадий Григорьевич — доктор физико-математических наук, профессор, ведущий научный сотрудник Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **zabudsky@ofim.oscsbras.ru**

ЛЕОНТЬЕВ Владимир Константинович — доктор физико-математических наук, профессор, заведующий сектором комбинаторного анализа Вычислительного центра им. А. А. Дородницына ФИЦ ИУ РАН, г. Москва. E-mail: **vkleontiev@yandex.ru**

ПАНПУРИН Андрей Александрович — студент РТУ МИРЭА, г. Москва.
E-mail: **aa.panpurin@yandex.ru**

ПРОЛУБНИКОВ Александр Вячеславович — кандидат физико-математических наук, доцент Новосибирского государственного университета, г. Новосибирск.
E-mail: **a.v.prolubnikov@mail.ru**

ТРИФОНОВ Дмитрий Игоревич — эксперт технического комитета по стандартизации «Криптографическая защита информации», г. Москва.
E-mail: **d.arlekino@gmail.com**

ФОМИН Денис Бонославович — кандидат физико-математических наук, ведущий научный сотрудник Академии криптографии Российской Федерации, г. Москва.
E-mail: **dbfomin@kryptonian.ru**

ЧИЖОВ Иван Владимирович — кандидат физико-математических наук, доцент кафедры информационной безопасности факультета ВМК МГУ имени М. В. Ломоносова, с.н.с. Федерального исследовательского центра «Информатика и управление» РАН, зам. по науке руководителя лаборатории криптографии АО «НПК „Криптонит“», г. Москва. E-mail: **ichizhov@cs.msu.ru**

KOMATHI M. Mani — Master of Science, Research Scholar, Department of Mathematics, Vellore Institute of Technology, Vellore. E-mail: **komathirk2108@gmail.com**

RAGUKUMAR Pandurangan — Doctor of Philosophy, Assistant Professor Senior, Department of Mathematics, Vellore Institute of Technology, Vellore.
E-mail: **ragukumar2003@gmail.com**

Журнал «Прикладная дискретная математика» входит в перечень ВАК рецензируемых научных изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание учёной степени кандидата и доктора наук по специальностям 2.3.5. «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей» (технические науки), 2.3.6. «Методы и системы защиты информации, информационная безопасность» (физико-математические и технические науки), 1.1.5. «Математическая логика, алгебра, теория чисел и дискретная математика» (физико-математические науки), 1.2.3. «Теоретическая информатика, кибернетика» (физико-математические науки), а также в перечень журналов, рекомендованных ФУМО ВО ИБ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал индексируется в базах данных Web of Science (Emerging Sources Citation Index (ESCI) и Russian Science Citation Index (RSCI)), Scopus, MathSciNet и Zentralblatt MATH. По решению ВАК от 21.12.2023 он отнесён к первой категории (К1) научных журналов, входящих в Перечень ВАК.

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте journals.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также правила подготовки рукописей статей для публикации в журнале.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надёжности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*