МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2026 № 71

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 24.02.2026. Формат 60 × 84 1/8. Усл. п. л. 14,8. Тираж 300 экз.
Заказ № 6689. Цена свободная. Дата выхода в свет 23.03.2026.

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

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЁЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 519.714.5 DOI 10.17223/20710410/71/1

ДВОИЧНЫЕ ПОРОГОВЫЕ ПОДСТАНОВКИ

А. И. Зобов, И. В. Чередник

РТУ МИРЭА, г. Москва, Россия

E-mail: zobowai@gmail.com, icherednick@mail.ru

Продолжается исследование двоичных пороговых подстановок — биективных преобразований множества двоичных векторов, координатные функции которых являются пороговыми. Доказано, что семейство всех двоичных пороговых подстановок множества $\{0,1\}^n$ порождает импримитивную группу, которая действует на множестве 2^{n-1} блоков $\{\mathbf{a},\overline{\mathbf{a}}\}$ подстановочно подобно сплетению $S_2 \wr S_{2^{n-1}}$. Показано, что в классе $\{0,1\}$-матриц лишь подстановочные матрицы реализуют пороговые подстановки. Предложен рекурсивный способ построения класса полноцикловых пороговых подстановок, исследована возможность практического применения таких подстановок.

Ключевые слова: *двоичные пороговые функции, двоичные биективные преобразования, пороговые подстановки.*

BINARY TRESHOLD SUBSTITUTIONS

A. I. Zobov, I. V. Cherednik

RTU MIREA, Moscow, Russia

We continue the study of binary threshold substitutions — bijective transformations of the set of binary vectors whose coordinate functions are threshold functions. It is proven that the set of binary threshold substitutions generates an imprimitive group, which acts on the set of 2^{n-1} blocks $\{\mathbf{a},\overline{\mathbf{a}}\}$ permutationally similar to the wreath product $S_2 \wr S_{2^{n-1}}$. It is shown that, within the class of $\{0,1\}$-matrices, only permutation matrices realize threshold substitutions. A recursive method for constructing a class of full-cycle threshold substitutions is proposed. The possibility of practical applications of such substitutions is investigated.

Keywords: *binary threshold functions, binary bijective transformations, threshold substitutions.*

Введение

Пороговые булевы функции $f\colon \{0,1\}^n \to \{0,1\}$, определяемые линейными неравенствами с действительными коэффициентами

$$f(x_1,\ldots,x_n)=1 \iff a_1x_1+\ldots+a_nx_n \geqslant b,$$

давно являются классическими объектами исследований [1–3], а их практическая привлекательность обуславливается простотой реализации в модели нейросетевых вычислений. В последнее время различные представители научной школы В. Г. Никонова ведут активные исследования [4–10] в области синтеза в базисе пороговых функций таких дискретных отображений, которые допускают эффективную реализацию в модели нейросетевых вычислений и/или на альтернативной элементной базе, а также пригодны к использованию в узлах защиты информации. Однако результаты работ [4–8] по существу позволяют построить лишь штучные примеры новых пороговых подстановок, а в [9, 10] исследуются не стандартные пороговые функции, а задаваемые квадратичными неравенствами. В данной работе предлагается способ построения нового семейства двоичных полноцикловых биективных преобразований, координатные функции которых являются пороговыми.

Всюду далее, если не оговорено иное, мы будем исследовать пороговые двоичные функции в псевдобулевом представлении $f\colon \{\pm1\}^n \to \{\pm1\}$ с естественным сохранением стандартной терминологии (сбалансированность, двойственность и пр.). Такой «центрально-симметричный» подход к представлению пороговых двоичных отображений действительно в ряде случаев удобнее классического булевого представления [3].

Определение 1. Псевдобулева функция $f\colon \{\pm1\}^n \to \{\pm1\}$ называется

— *пороговой*, если существуют $a_1, \ldots, a_n, b \in \mathbb{R}$, для которых выполняется условие

$$f(x_1, \ldots, x_n) = 1 \iff a_1x_1 + \ldots + a_nx_n \geqslant b; \tag{1}$$

— *сбалансированной*, если $|f^{-1}(1)| = |f^{-1}(-1)|$;

— *самодвойственной*, если $f(-\mathbf{x}) = -f(\mathbf{x})$ для каждого $\mathbf{x} = (x_1, \ldots, x_n) \in \{\pm1\}^n$.

Утверждение 1. Пусть $a_1, \ldots, a_n, b \in \mathbb{R}$ и пороговая функция f, определяемая условием (1), является сбалансированной. Тогда пороговая функция f также может быть задана центрально-симметричным условием

$$f(x_1, \ldots, x_n) = 1 \iff a_1x_1 + \ldots + a_nx_n \geqslant 0$$

и при этом множество $\{\pm1\}^n$ не содержит решений уравнения $a_1x_1 + \ldots + a_nx_n = 0$.

Доказательство. Пусть неравенство $a_1x_1 + \ldots + a_nx_n \geqslant 0$ задаёт пороговую функцию g. Тогда, как нетрудно видеть,

$$g^{-1}(-1) = \{\mathbf{b}_1, \ldots, \mathbf{b}_t\}, \quad g^{-1}(1) = \{-\mathbf{b}_1, \ldots, -\mathbf{b}_t\} \cup \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^n-2t}\}, \quad t \leqslant 2^{n-1},$$

где наборы $\mathbf{c}_i = (c_1^{(i)}, \ldots, c_n^{(i)}) \in \{\pm1\}^n$ удовлетворяют условию $a_1c_1^{(i)} + \ldots + a_nc_n^{(i)} = 0$ при всех $i \in \{1, \ldots, 2^n - 2t\}$. Во введённых обозначениях:

— при $b > 0$ выполняется соотношение $\big(\{\mathbf{b}_1, \ldots, \mathbf{b}_t\} \cup \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^n-2t}\}\big) \subset f^{-1}(-1)$;

— при $b \leqslant 0$ имеет место $\big(\{-\mathbf{b}_1, \ldots, -\mathbf{b}_t\} \cup \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^n-2t}\}\big) \subset f^{-1}(1)$.

В обоих случаях из условия сбалансированности $|f^{-1}(-1)| = 2^{n-1} = |f^{-1}(1)|$ необходимо следует, что $t = 2^{n-1}$ и соответственно

$$g^{-1}(-1) = \{\mathbf{b}_1, \ldots, \mathbf{b}_{2^{n-1}}\} = f^{-1}(-1), \quad g^{-1}(1) = \{-\mathbf{b}_1, \ldots, -\mathbf{b}_{2^{n-1}}\} = f^{-1}(1).$$

Утверждение 1 доказано. ■

Итак, согласно утверждению 1, любую сбалансированную пороговую функцию $f\colon \{\pm1\}^n \to \{\pm1\}$ можно задать неравенством $a_1x_1 + \ldots + a_nx_n \geqslant 0$ и при этом множество $\{\pm1\}^n$ не содержит решений уравнения $a_1x_1 + \ldots + a_nx_n = 0$. Значит, произвольную

сбалансированную пороговую функцию f можно записать формулой

$$f(x_1, \ldots, x_n) = \operatorname{sgn}(a_1 x_1 + \ldots + a_n x_n)$$

или кратко в векторной форме записи $f(\mathbf{x}) = \operatorname{sgn}(\mathbf{a}\mathbf{x}^{\downarrow})$, где

$$\operatorname{sgn}(y) = \begin{cases} -1, & y < 0, \\ 0, & y = 0, \\ 1, & y > 0. \end{cases}$$

Следствие 1. Пороговая функция $f\colon \{\pm 1\}^n \to \{\pm 1\}$ является самодвойственной тогда и только тогда, когда она является сбалансированной.

Доказательство. Необходимость очевидна, а достаточность следует из центрально-симметричного определения сбалансированной пороговой функции $f(\mathbf{x})$ в виде $f(\mathbf{x}) = \operatorname{sgn}(\mathbf{a}\mathbf{x}^{\downarrow})$. ■

Замечание 1. Поскольку множество $\mathbb{Q}$ всюду плотно в $\mathbb{R}$, а линейные функции непрерывны, нетрудно понять, что в задании произвольной пороговой функции в виде (1) коэффициенты $a_1, \ldots, a_n, b$ всегда могут быть выбраны рациональными и, более того, целочисленными [3]. Поэтому в некоторых случаях для удобства (как, например, далее в теореме 3) без ограничения общности можно полагать, что неравенства, определяющие пороговые функции, имеют целочисленные коэффициенты.

Центральным объектом исследования данной работы являются биективные преобразования $F(\mathbf{x}) = \big(f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})\big)$ множества $\{\pm 1\}^n$, у которых все координатные функции являются пороговыми:

$$f_i(\mathbf{x}) = 1 \iff \mathbf{a}_i \mathbf{x}^{\downarrow} \geqslant b_i, \quad i \in \{1, \ldots, n\}.$$

Кратко такие преобразования будем называть *пороговыми подстановками*. Практическая значимость пороговых подстановок обуславливается максимальной простотой их реализации в базисе пороговых функций; теоретическая значимость состоит в том, что связанные с данными преобразованиями проблемы по существу являются фундаментальными задачами теории целочисленных линейных неравенств.

Каждая координатная функция пороговой подстановки $F(\mathbf{x}) = \big(f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})\big)$ является сбалансированной и, согласно утверждению 1, может быть задана условием

$$f_i(\mathbf{x}) = 1 \iff \mathbf{a}_i \mathbf{x}^{\downarrow} \geqslant 0,$$

а также кратко в виде $f_i(\mathbf{x}) = \operatorname{sgn}(\mathbf{a}_i \mathbf{x}^{\downarrow})$, $i \in \{1, \ldots, n\}$. Таким образом, для пороговой подстановки $F(\mathbf{x})$ можно использовать краткую функциональную форму записи $F(\mathbf{x}) = \operatorname{sgn}\big(A\mathbf{x}^{\downarrow}\big)$, где $A = \begin{pmatrix} \mathbf{a}_1 \\ \ldots \\ \mathbf{a}_n \end{pmatrix}$ — матрица, которую будем называть *матрицей пороговой подстановки* $F(\mathbf{x})$.

Очевидно, что матрица $A \in \mathbb{R}_{n,n}$ определяет пороговую подстановку $\operatorname{sgn}\big(A\mathbf{x}^{\downarrow}\big)$ в том и только в том случае, когда гиперплоскости $\mathbf{a}_1\mathbf{x}^{\downarrow} = 0$, …, $\mathbf{a}_n\mathbf{x}^{\downarrow} = 0$ разбивают пространство $\mathbb{R}^n$ на 2^n частей, каждая из которых содержит единственный набор из множества $\{\pm 1\}^n$. Отсюда, согласно классическому результату Шлефли, следует, что если матрица $A \in \mathbb{R}_{n,n}$ определяет пороговую подстановку $F(\mathbf{x}) = \operatorname{sgn}\big(A\mathbf{x}^{\downarrow}\big)$, то данная матрица A обязательно является обратимой.

Утверждение 2 (теорема Шлефли). Максимальное число n-мерных открытых многогранных конусов, возникающих при разбиении пространства $\mathbb{R}^n$ гиперплоскостями $\mathbf{a}_1\mathbf{x}^{\downarrow} = 0, \ldots, \mathbf{a}_t\mathbf{x}^{\downarrow} = 0$, равно $2\sum\limits_{i=0}^{n-1}\binom{t-1}{i}$, и этот максимум достигается в том и только в том случае, если гиперплоскости находятся в общем положении.

Доказательство. См., например, в [3]. ■

К сожалению, на данный момент более не известно ничего содержательного относительно свойств матрицы A, определяющей пороговую подстановку $\operatorname{sgn}\left(A\mathbf{x}^{\downarrow}\right)$ [3]. Так, например, неизвестно точное количество пороговых подстановок на множестве $\{\pm 1\}^n$ при произвольном n, и похоже, что даже задача определения «задаёт ли конкретная матрица A пороговую подстановку» является трудной. Поясним последнее утверждение: для того чтобы отображение $\operatorname{sgn}\left(A\mathbf{x}^{\downarrow}\right)$ было биективным, согласно критерию Хаффмана [11], необходимо и достаточно, чтобы для любого подмножества $\{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$ система неравенств

$$\begin{cases} \mathbf{a}_{i_1}\mathbf{x}^{\downarrow} > 0, \\ \ldots \\ \mathbf{a}_{i_k}\mathbf{x}^{\downarrow} > 0 \end{cases}$$

имела ровно 2^{n-k} решений из множества $\{\pm 1\}^n$. Однако уже при $k = 1$ мы имеем известную NP-полную задачу: неравенство

$$a_1x_1 + \ldots + a_nx_n > 0$$

имеет 2^{n-1} решений из $\{\pm 1\}^n$ тогда и только тогда, когда уравнение

$$a_1x_1 + \ldots + a_nx_n = 0$$

не имеет $\{\pm 1\}$-решений или, что то же самое, уравнение

$$|a_1|x_1 + \ldots + |a_n|x_n = 0$$

не имеет $\{\pm 1\}$-решений; последнее условие равносильно тому, что для любого разбиения $\{1, \ldots, n\} = \{s_1, \ldots, s_r\} \sqcup \{s_{r+1}, \ldots, s_n\}$ выполняется неравенство

$$|a_{s_1}| + \ldots + |a_{s_r}| \neq |a_{s_{r+1}}| + \ldots + |a_{s_n}|$$

— известная NP-полная задача о разбиении [12]. При этом стоит отметить, что NP-полной является лишь массовая задача о разбиении, в то время как для некоторых частных случаев данная задача может иметь даже тривиальные решения. Так, например, в случае, когда $a_1, \ldots, a_n \in \mathbb{Z}$ и $a_1 + \ldots + a_n$ — нечётное число, решение задачи вполне очевидно.

Отметим ещё одну насущную практическую проблему, связанную с пороговыми подстановками: будет ли обратная к пороговой подстановке также пороговой и как, в случае положительного ответа, построить матрицу, которая задаёт обратную пороговую подстановку? В настоящий момент относительно данной проблемы не известно ничего содержательного. Однако пример ортогональной матрицы

$$A = \begin{pmatrix} \dfrac{4}{\sqrt{41}} & \dfrac{3}{\sqrt{41}} & \dfrac{4}{\sqrt{41}} \\ \dfrac{11}{\sqrt{11562}} & \dfrac{80}{\sqrt{11562}} & \dfrac{-71}{\sqrt{11562}} \\ \dfrac{-13}{\sqrt{282}} & \dfrac{8}{\sqrt{282}} & \dfrac{7}{\sqrt{282}} \end{pmatrix},$$

которая задаёт пороговую подстановку, при том, что обратная к ней A^{T} вообще не задаёт пороговую подстановку, наводит на мысль, что едва ли обозначенная проблема допускает решение в терминах классической линейной алгебры [5, 6].

В заключение рассмотрим простой, но чрезвычайно полезный класс пороговых подстановок, который в дальнейшем будем неоднократно использовать.

Пример 1. Преобразование множества $\{\pm 1\}^n$, определённое по правилу

$$(x_1, \ldots, x_n) \mapsto (\varepsilon_1 x_{\pi(1)}, \ldots, \varepsilon_n x_{\pi(n)}),$$

где $\varepsilon_1, \ldots, \varepsilon_n \in \{\pm 1\}$; π — перестановка из симметрической группы S_n, может быть реализовано в качестве пороговой подстановки с матрицей

$$\begin{array}{c} \begin{array}{ccccccc} & \pi(1) & & \pi(n) & & \pi(2) & \end{array} \\ \begin{pmatrix} \ldots & \varepsilon_1 & 0 & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & 0 & \varepsilon_2 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & \varepsilon_n & 0 & \ldots & 0 \end{pmatrix} \end{array}$$

(матрицы данного вида будем называть $\{\pm 1\}$-подстановочными).

Множество всех таких преобразований образует группу относительно композиции отображений. По аналогии с булевым случаем, данную группу будем называть группой Джевонса и обозначать $\mathfrak{Q}_n$. Напомним, что группа Джевонса совпадает с множеством всех изометрий пространства $\{\pm 1\}^n$ относительно метрики Хэмминга [13]. На текущий момент группа Джевонса $\mathfrak{Q}_n$ — единственная алгебраическая структура, которая обнаружена в множестве всех пороговых подстановок.

1. Строение группы, порождённой пороговыми подстановками

Согласно следствию 1, координатные функции пороговой подстановки являются самодвойственными, а следовательно, произвольная пороговая подстановка реализует самодвойственное преобразование с блоками $[\mathbf{a}] = \{\pm \mathbf{a}\}$, $\mathbf{a} \in \{\pm 1\}^n$. Значит, группа $\mathfrak{G}_n$, порождаемая множеством всех пороговых подстановок, заведомо является импримитивной с системой блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$.

Любая группа подстановок с системой импримитивности $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$, содержится в группе подстановок $\mathfrak{S}_n$, которая подобна сплетению $S_2 \wr S_{2^{n-1}}$ и действует на множестве 2^{n-1} блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$, следующим образом: все 2^{n-1} блоков переставляются свободным образом, а внутри блоков осуществляются независимые биективные преобразования (тождественное или инвертирование). Итак, $\mathfrak{G}_n \subset \mathfrak{S}_n$. Однако на самом деле справедлив следующий результат:

Теорема 1. $\mathfrak{G}_n = \mathfrak{S}_n$.

Доказательство. Включение $\mathfrak{G}_n \subset \mathfrak{S}_n$ отмечено ранее. Для доказательства обратного включения $\mathfrak{S}_n \subset \mathfrak{G}_n$ сделаем предварительно несколько примечаний.

1. Пороговая подстановка с матрицей

$$\begin{pmatrix} n-2 & -1 & \ldots & -1 \\ -1 & n-2 & \ldots & -1 \\ \vdots & & \ddots & \vdots \\ -1 & \ldots & -1 & n-2 \end{pmatrix}$$

определяет транспозицию T_1, которая действует нетождественным образом только внутри класса $[(1,\dots,1)]$:

$$(1,\dots,1) \leftrightarrow (-1,\dots,-1).$$

2. Аналогично, транспозиция, которая переставляет векторы в произвольном блоке $[(\varepsilon_1,\dots,\varepsilon_n)]$, $(\varepsilon_1,\dots,\varepsilon_n) \in \{\pm 1\}^n$, может быть реализована как пороговая подстановка с матрицей

$$\begin{pmatrix} \varepsilon_1 & 0 & \dots & 0 \\ 0 & \varepsilon_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \varepsilon_n \end{pmatrix} \begin{pmatrix} n-2 & -1 & \dots & -1 \\ -1 & n-2 & \dots & -1 \\ \vdots & & \ddots & \vdots \\ -1 & \dots & -1 & n-2 \end{pmatrix} \begin{pmatrix} \varepsilon_1 & 0 & \dots & 0 \\ 0 & \varepsilon_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \varepsilon_n \end{pmatrix}$$

(фактически — это сопряжение транспозиции T_1 подстановкой трансляции

$$(x_1,\dots,x_n) \mapsto (\varepsilon_1 x_1,\dots,\varepsilon_n x_n)$$

из группы Джевонса).

3. Пороговая подстановка T_2 с матрицей

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & n-3 & -1 & \dots & -1 \\ 0 & -1 & n-3 & \dots & -1 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & -1 & \dots & -1 & n-3 \end{pmatrix}_{n\times n}$$

действует нетождественным образом исключительно на четырёх элементах:

$$\begin{gathered} (1,1,\dots,1) \leftrightarrow (1,-1,\dots,-1), \\ (-1,-1,\dots,-1) \leftrightarrow (-1,1,\dots,1), \end{gathered}$$

значит, она определяет транспозицию блоков

$$[(1,1,\dots,1)] \;\leftrightarrow\; [(-1,1,\dots,1)].$$

4. Сопрягая транспозицию блоков T_2 подстановкой из группы Джевонса

$$(x_1,x_2,x_3,\dots,x_n) \mapsto (-x_2,x_1,x_3,\dots,x_n),$$

получаем транспозицию блоков

$$[(-1,1,1,\dots,1)] \;\leftrightarrow\; [(-1,-1,1,\dots,1)].$$

Аналогичным образом можно построить пороговые подстановки, реализующие транспозиции блоков

$$\begin{gathered} [(-1,-1,1,\dots,1)] \;\leftrightarrow\; [(-1,-1,-1,\dots,1)], \\ \dots \\ [(-1,\dots,-1,1,1)] \;\leftrightarrow\; [(-1,\dots,-1,-1,1)], \end{gathered}$$

а также все возможные транспозиции блоков, которые имеют представителями соседние векторы.

Итак, согласно примечанию 4, существует набор пороговых подстановок-транспозиций блоков, который обеспечивает «связность» всех возможных блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$, а следовательно, ввиду известной теоремы Пойа, является системой образующих симметрической группы подстановок на множестве блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$. Кроме того, согласно примечанию 2, существуют пороговые подстановки, которые реализуют транспозиции внутри каждого из блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$. Значит, справедливо включение $\mathfrak{S}_n \subset \mathfrak{G}_n$. ■

Из доказательства теоремы 1 нетрудно вывести верхнюю оценку ширины группы $\mathfrak{S}_n$ относительно множества пороговых подстановок.

Следствие 2. Произвольная подстановка из группы $\mathfrak{S}_n$ представляется в виде произведения не более чем $n2^n$ пороговых подстановок.

Доказательство. Приведённая оценка вытекает из следующих очевидных соображений. Для построения произвольной подстановки из $\mathfrak{S}_n$ необходимо построить соответствующую перестановку блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$, а также выполнить не более 2^{n-1} инвертирований в блоках. Построение произвольной перестановки блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$, требует не более $2^{n-1}-1$ транспозиций блоков произвольного вида $([\mathbf{a}], [\mathbf{b}])$, каждую из которых можно вычислить с использованием не более $2n-3$ транспозиций из примечания 4 доказательства теоремы 1. Поясним подробнее: произвольные наборы $\mathbf{a}$, $\mathbf{b}$ всегда можно соединить цепочкой $\mathbf{a} = \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k = \mathbf{b}$ длины $k \leqslant n-1$, в которой подряд идущие наборы являются соседними, а значит, для транспозиции $([\mathbf{a}], [\mathbf{b}])$ справедливо следующее представление в виде произведения транспозиций:

$$([\mathbf{a}], [\mathbf{b}]) = ([\mathbf{a}_1], [\mathbf{a}_2]) \ldots ([\mathbf{a}_{k-2}], [\mathbf{a}_{k-1}])([\mathbf{a}_{k-1}], [\mathbf{a}_k])([\mathbf{a}_{k-1}], [\mathbf{a}_{k-2}]) \ldots ([\mathbf{a}_2], [\mathbf{a}_1]).$$

Итого требуется не более $2^{n-1} + (2^{n-1}-1)(2n-3) \leqslant n2^n$ пороговых подстановок. ■

Геометрическое представление пороговых функций подсказывает, что при $n \geqslant 3$ группа $\mathfrak{S}_n$ обязательно содержит подстановки, которые не являются пороговыми.

Следствие 3. При любом $n \geqslant 3$ класс всех пороговых подстановок множества $\{\pm 1\}^n$ не замкнут относительно операции композиции.

Доказательство. При $n \geqslant 3$ группа $\mathfrak{S}_n$ содержит подстановку g, действие которой удовлетворяет следующим условиям:

$$\begin{array}{ll}
g(1,1,\ldots,1) = (-1,-1,\ldots,-1), & g(-1,-1,\ldots,-1) = (1,1,\ldots,1), \\
g(-1,1,\ldots,1) = (1,-1,1,\ldots,1), & g(1,-1,\ldots,-1) = (-1,1,-1,\ldots,-1), \\
g(1,-1,\ldots,1) = (1,1,-1,\ldots,1), & g(-1,1,\ldots,-1) = (-1,-1,1,\ldots,-1), \\
\multicolumn{2}{c}{\ldots} \\
g(1,1,\ldots,-1,1) = (1,1,\ldots,1,-1), & g(-1,-1,\ldots,1,-1) = (-1,-1,\ldots,-1,1), \\
g(1,1,\ldots,1,-1) = (1,-1,\ldots,-1,-1), & g(-1,-1,\ldots,-1,1) = (-1,1,\ldots,1,1)
\end{array}$$

(на множестве всех остальных блоков допустимо любое взаимно однозначное соответствие). Если предположить, что первая координатная функция подстановки g допускает пороговую реализацию с определяющим неравенством $a_1x_1 + a_2x_2 + \ldots + a_nx_n > 0$, то из условий, определяющих действие g, необходимо вытекает следующее противоречие:

$$\begin{cases} a_1 + a_2 + a_3 + \ldots + a_n < 0, \\ -a_1 + a_2 + a_3 + \ldots + a_n > 0, \\ a_1 - a_2 + a_3 + \ldots + a_n > 0, \\ \ldots \\ a_1 + a_2 + \ldots - a_{n-1} + a_n > 0, \\ a_1 + a_2 + \ldots + a_{n-1} - a_n > 0 \end{cases} \quad \Longrightarrow \quad \begin{cases} a_1 + a_2 + a_3 + \ldots + a_n < 0, \\ (n-2)(a_1 + a_2 + a_3 + \ldots + a_n) > 0. \end{cases}$$

Следствие 3 доказано. ■

В заключение приведём пример, который опровергает наивное предположение, что множество всех пороговых подстановок действует симметрическим образом хотя бы на множестве блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^n$.

Пример 2. Рассмотрим подстановку на множестве блоков $[\mathbf{a}]$, $\mathbf{a} \in \{\pm 1\}^4$:

$$\begin{gathered} [(1,1,1,1)] \mapsto [(1,1,1,1)], \\ [(1,1,-1,-1)] \mapsto [(1,1,1,-1)], \\ [(1,-1,-1,1)] \mapsto [(1,1,-1,1)], \\ [(1,-1,1,-1)] \mapsto [(1,1,-1,-1)], \\ [(-1,1,1,1)] \mapsto [(1,-1,1,1)], \\ [(1,-1,1,1)] \mapsto [(1,-1,1,-1)], \\ [(1,1,-1,1)] \mapsto [(1,-1,-1,1)], \\ [(1,1,1,-1)] \mapsto [(1,-1,-1,-1)]. \end{gathered}$$

Если предположить, что такая подстановка может быть реализована некоторой пороговой подстановкой g, то для действия g возможны 256 вариантов:

$$\begin{gathered} g(1,1,1,1) = \varepsilon_1(1,1,1,1), \\ g(1,1,-1,-1) = \varepsilon_2(1,1,1,-1), \\ g(1,-1,-1,1) = \varepsilon_3(1,1,-1,1), \\ g(1,-1,1,-1) = \varepsilon_4(1,1,-1,-1), \\ g(-1,1,1,1) = \varepsilon_5(1,-1,1,1), \\ g(1,-1,1,1) = \varepsilon_6(1,-1,1,-1), \\ g(1,1,-1,1) = \varepsilon_7(1,-1,-1,1), \\ g(1,1,1,-1) = \varepsilon_8(1,-1,-1,-1), \end{gathered}$$

при $\varepsilon_1, \ldots, \varepsilon_8 \in \{\pm 1\}$. В каждом из 256 предполагаемых вариантов коэффициенты линейных неравенств, определяющих первые две координатные функции указанной подстановки g:

$$\begin{aligned} g_1(x_1,x_2,x_3,x_4) = 1 &\iff a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 \geqslant 0, \\ g_2(x_1,x_2,x_3,x_4) = 1 &\iff b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \geqslant 0, \end{aligned}$$

необходимо должны удовлетворять системе неравенств

$$\begin{cases}
\varepsilon_1(a_1+a_2+a_3+a_4)>0,\\
\varepsilon_2(a_1+a_2-a_3-a_4)>0,\\
\varepsilon_3(a_1-a_2-a_3+a_4)>0,\\
\varepsilon_4(a_1-a_2+a_3-a_4)>0,\\
\varepsilon_5(-a_1+a_2+a_3+a_4)>0,\\
\varepsilon_6(a_1-a_2+a_3+a_4)>0,\\
\varepsilon_7(a_1+a_2-a_3+a_4)>0,\\
\varepsilon_8(a_1+a_2+a_3-a_4)>0,\\
\varepsilon_1(b_1+b_2+b_3+b_4)>0,\\
\varepsilon_2(b_1+b_2-b_3-b_4)>0,\\
\varepsilon_3(b_1-b_2-b_3+b_4)>0,\\
\varepsilon_4(b_1-b_2+b_3-b_4)>0,\\
\varepsilon_5(-b_1+b_2+b_3+b_4)<0,\\
\varepsilon_6(b_1-b_2+b_3+b_4)<0,\\
\varepsilon_7(b_1+b_2-b_3+b_4)<0,\\
\varepsilon_8(b_1+b_2+b_3-b_4)<0,
\end{cases}$$

которая не имеет решений при любых $\varepsilon_1,\ldots,\varepsilon_8\in\{\pm1\}$ (проверка выполнена в системе Mathematica).

2. Пороговые подстановки, реализуемые $\{0,1\}$-матрицами

Согласно теореме 1, пороговые подстановки порождают достаточно обширный класс подстановок $\mathfrak{S}_n$. Однако результаты теоремы 1, выраженные в следствии 2, обескураживают ожидания относительно простоты получения произвольной подстановки из множества $\mathfrak{S}_n$ в виде композиции пороговых подстановок. Между тем практический интерес заключается в построении классов эффективно реализуемых пороговых подстановок, которые также обладают цикловой структурой, пригодной для использования в узлах защиты информации. Поэтому вполне естественно намерение изучить пороговые подстановки, которые могут быть определены максимально простыми матрицами, состоящими только из 0 и 1.

Теорема 2. $\{0,1\}$-матрица $A_{n\times n}$ задаёт пороговую подстановку $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ в том и только в том случае, когда A — подстановочная матрица.

Доказательство. Достаточность очевидна, докажем необходимость методом «от противного». Предположим, что матрица A не является подстановочной и, не ограничивая общности, её первые две строки имеют следующий вид:

$$(\underbrace{1,\ldots,1}_{k},\underbrace{1,\ldots,1}_{s\geqslant1},\underbrace{0,\ldots,0}_{m},0,\ldots,0),$$
$$(\underbrace{0,\ldots,0}_{k},\underbrace{1,\ldots,1}_{s\geqslant1},\underbrace{1,\ldots,1}_{m},0,\ldots,0).$$

Указанные строки определяют две первые координатные функции порогового отображения $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$:

$$f_1(x_1,\ldots,x_n)=\operatorname{sgn}(x_1+\ldots+x_k+x_{k+1}+\ldots+x_{k+s}),$$
$$f_2(x_1,\ldots,x_n)=\operatorname{sgn}(x_{k+1}+\ldots+x_{k+s}+x_{k+s+1}+\ldots+x_{k+s+m}).$$

Здесь стоит отметить, что сбалансированность координатных функций f_1 и f_2 необходимо влечёт условие нечётности $k+s$ и $s+m$.

Для получения противоречия рассчитаем и сравним мощности множеств

$$M_{1,1} = \big\{\mathbf{x} \in \{\pm 1\}^n : (f_1(\mathbf{x}), f_2(\mathbf{x})) = (1,1)\big\},$$
$$M_{1,-1} = \big\{\mathbf{x} \in \{\pm 1\}^n : (f_1(\mathbf{x}), f_2(\mathbf{x})) = (1,-1)\big\}.$$

При перечислении векторов из $M_{1,1}$ нетрудно установить равенство

$$|M_{1,1}| = 2^{n-k-s-m} \sum_{t=0}^{s} \binom{s}{t} \sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} \sum_{j \geqslant (m+s+1)/2-t} \binom{m}{j},$$

где в перечисляемых наборах t обозначает количество единиц, расположенных на местах с $k+1$ по $k+s$; i — количество единиц на местах с 1 по k; j — количество единиц на местах с $k+s+1$ по $k+s+m$ (здесь и далее значение биномиального коэффициента $\binom{x}{y}$ стандартно полагается равным нулю при $x < y$, а также при $y < 0$). Аналогично

$$|M_{1,-1}| = 2^{n-k-s-m} \sum_{t=0}^{s} \binom{s}{t} \sum_{i \geqslant (k+s+1)/2-t}^{k} \binom{k}{i} \sum_{j \geqslant (m-s+1)/2+t} \binom{m}{j}.$$

Теперь рассмотрим величину

$$\frac{|M_{1,1}| - |M_{1,-1}|}{2^{n-k-s-m}} = \sum_{t=0}^{s} \binom{s}{t} \sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} \underbrace{\left[\sum_{j \geqslant (m+s+1)/2-t} \binom{m}{j} - \sum_{j \geqslant (m-s+1)/2+t} \binom{m}{j} \right]}_{N_{m,s,t}}$$

и заметим, что $N_{m,s,t} = -N_{m,s,s-t}$ при любых m и $t \leqslant s/2$ (в частности, $N_{m,s,s/2} = 0$ при чётном s). Продолжим:

$$\frac{|M_{1,1}| - |M_{1,-1}|}{2^{n-k-s-m}} = \sum_{t<s/2} \binom{s}{t} \sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} N_{m,s,t} + \sum_{t>s/2} \binom{s}{t} \sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} N_{m,s,t} =$$
$$= \sum_{t<s/2} \binom{s}{t} N_{m,s,t} \sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} - \sum_{t<s/2} \binom{s}{t} N_{m,s,t} \sum_{i \geqslant (k-s+1)/2+t} \binom{k}{i} =$$
$$= \sum_{t<s/2} \binom{s}{t} N_{m,s,t} \underbrace{\left[\sum_{i \geqslant (k+s+1)/2-t} \binom{k}{i} - \sum_{i \geqslant (k-s+1)/2+t} \binom{k}{i} \right]}_{N_{k,s,t}} = \sum_{t<s/2} \binom{s}{t} N_{m,s,t} N_{k,s,t}.$$

Заметим, что при любых $m \in \mathbb{N}_0$, $s \in \mathbb{N}$ и $t < s/2$ имеет место

$$\begin{cases} (m-s+1)/2 + t < (m+s+1)/2 - t, \\ (m-s+1)/2 + t \leqslant m, \\ (m+s+1)/2 - t > 0 \end{cases} \implies N_{m,s,t} = -\sum_{j=(m-s+1)/2+t}^{(m+s+1)/2-t-1} \binom{m}{j} < 0.$$

Итак, показали, что при $s \geqslant 1$ всегда выполняется строгое неравенство

$$\frac{|M_{1,1}| - |M_{1,-1}|}{2^{n-k-s-m}} > 0,$$

что противоречит независимости сбалансированных координатных функций f_1 и f_2. ■

Следствие 4. Если для матрицы $A \in \mathbb{R}_{n,n}$ линейное отображение $\mathbf{x}^{\downarrow} \mapsto A\mathbf{x}^{\downarrow}$ задаёт перестановку на множестве $\{\pm 1\}^n$, то A является $\{\pm 1\}$-подстановочной матрицей.

Доказательство. Не ограничивая общности, для удобства будем полагать, что $A\mathbf{1}^{\downarrow} = \mathbf{1}^{\downarrow}$ (при необходимости можно домножить соответствующие строки матрицы A на -1). В таком случае легко видеть, что столбец

$$2A_1^{\downarrow} = A\left(\mathbf{1}^{\downarrow} - (-1, 1, \ldots, 1)^{\mathrm{T}}\right) = A\mathbf{1}^{\downarrow} - A(-1, 1, \ldots, 1)^{\mathrm{T}} = \mathbf{1}^{\downarrow} - (\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}}$$

состоит только из 0 и 2 и, следовательно, $A_1^{\downarrow}$ — $\{0, 1\}$-столбец. Аналогично показывается, что все остальные столбцы матрицы A также являются $\{0, 1\}$-столбцами.

Таким образом, A — $\{0, 1\}$-матрица, которая задаёт пороговую подстановку (на самом деле, действует точным образом), и, согласно теореме 2, A — подстановочная матрица. ■

Замечание 2. Из следствия 4 вытекает интересное наблюдение. Оказывается, что $\{\pm 1\}$-подстановочные матрицы описывают все изометрии множества $\{\pm 1\}^n$ не только относительно метрики Хэмминга, но и относительно метрики Евклида, поскольку в последнем случае изометрия обязательно является линейным отображением и попадает под условие следствия 4. Это вполне закономерно, поскольку на множестве $\{\pm 1\}^n$ метрики Евклида и Хэмминга эквивалентны.

В заключение данного пункта отметим, что описание $\{0, \pm 1\}$-матриц, которые определяют пороговые подстановки, представляется весьма сложной задачей. В настоящий момент не существует никаких решений для построения бесконечных классов $\{0, \pm 1\}$-матриц, определяющих пороговые подстановки. Так, например, в работах [4–6] приведены лишь частные примеры псевдоадамаровых матриц и конференц-матриц размеров 4×4, 6×6, 8×8 и 10×10, которые задают пороговые подстановки; при этом обоснование биективности рассматриваемых пороговых отображений также носит частный характер и не позволяет делать каких-либо выводов о существовании/построении псевдоадамаровых матриц и конференц-матриц, определяющих пороговые подстановки произвольной размерности.

3. Рекурсивное построение полноцикловых пороговых подстановок

Результаты п. 2 носят отрицательный характер в том смысле, что не удалось обнаружить вообще никаких новых пороговых подстановок, не говоря уже о классах пороговых подстановок. Здесь в терминах определяющих матриц исследуем возможность рекурсивного построения пороговых подстановок. Начнём с простейшего варианта рекурсивного продолжения существующих матриц.

Утверждение 3. Если матрица

$$B = \begin{pmatrix} b_{11} & * \\ 0^{\downarrow} & A_{n\times n} \end{pmatrix}_{(n+1)\times(n+1)}$$

задаёт пороговую подстановку $\operatorname{sgn}(B\mathbf{x}^{\downarrow})$ на множестве $\{\pm 1\}^{n+1}$, то данная пороговая подстановка также может быть задана матрицей

$$\begin{pmatrix} b_{11} & 0\ldots 0 \\ 0^{\downarrow} & A_{n\times n} \end{pmatrix}_{(n+1)\times(n+1)},$$

и при этом матрица A необходимо задаёт пороговую подстановку на множестве $\{\pm 1\}^n$.

Доказательство. Если система линейных неравенств

$$\begin{cases} b_{11}x_1 + b_{12}x_2 + \ldots + b_{1n+1}x_{n+1} > 0, \\ a_{11}x_2 + \ldots + a_{1n}x_{n+1} > 0, \\ \ldots \\ a_{n1}x_2 + \ldots + a_{nn}x_{n+1} > 0 \end{cases}$$

задаёт пороговую подстановку, то при любых $\varepsilon_2, \ldots, \varepsilon_{n+1} \in \{\pm 1\}$ неравенство

$$b_{11}x_1 + (b_{12}\varepsilon_2 + \ldots + b_{1n+1}\varepsilon_{n+1}) > 0$$

необходимо задаёт сбалансированную псевдобулеву функцию от переменной x_1 и, как нетрудно видеть, может быть заменено более простым неравенством $b_{11}x_1 > 0$ или вовсе тривиальным $\operatorname{sgn}(b_{11})x_1 > 0$. Остаётся заметить, что система

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n > 0, \\ \ldots \\ a_{n1}x_1 + \ldots + a_{nn}x_n > 0 \end{cases}$$

необходимо задаёт пороговую подстановку на множестве $\{\pm 1\}^n$. ■

Из утверждения 3 следует, что простейший способ продолжения существующей пороговой подстановки множества $\{\pm 1\}^n$ до пороговой подстановки множества $\{\pm 1\}^{n+1}$ посредством добавления ещё одной пороговой координатной функции допускает лишь тривиальные варианты.

Следствие 5. Не существует нетривиальных «треугольных» пороговых подстановок, отличных от подстановок вида $(x_1, \ldots, x_n) \mapsto (\varepsilon_1 x_1, \ldots, \varepsilon_n x_n)$.

Теперь приступим к изложению более продвинутого (по сравнению с утверждением 3) способа рекурсивного продолжения пороговой подстановки множества $\{\pm 1\}^n$ до пороговой подстановки множества $\{\pm 1\}^{n+1}$. В конечном итоге этот метод позволит построить целое семейство полноцикловых пороговых подстановок. Для удобства изложения здесь и далее наборы $\mathbf{a} \in \{\pm 1\}^n$ будем записывать в виде столбцов $\mathbf{a}^{\downarrow}$, при этом набор $-\mathbf{a}^{\downarrow}$ будем обозначать через $\overline{\mathbf{a}}^{\downarrow}$.

Утверждение 4. Если пороговая подстановка $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ на множестве $\{\pm 1\}^n$ реализует полный цикл, то указанный цикл имеет вид

$$\left(\mathbf{a}_1^{\downarrow}, \mathbf{a}_2^{\downarrow}, \ldots, \mathbf{a}_{2^{n-1}}^{\downarrow}, \overline{\mathbf{a}}_1^{\downarrow}, \overline{\mathbf{a}}_2^{\downarrow}, \ldots, \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\right). \tag{2}$$

Доказательство. Рассмотрим полный цикл пороговой подстановки $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$, начиная с произвольного элемента $\mathbf{a}_1^{\downarrow}$:

$$\left(\mathbf{a}_1^{\downarrow}, \mathbf{a}_2^{\downarrow}, \ldots\right).$$

Так как подстановка $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ является полноцикловой, то цикл содержит набор $\overline{\mathbf{a}}_1^{\downarrow}$:

$$\left(\mathbf{a}_1^{\downarrow}, \mathbf{a}_2^{\downarrow}, \ldots, \mathbf{a}_t^{\downarrow}, \overline{\mathbf{a}}_1^{\downarrow}, \ldots\right),$$

а поскольку она является самодвойственной, то её полный цикл имеет вид

$$\left(\mathbf{a}_1^{\downarrow}, \mathbf{a}_2^{\downarrow}, \ldots, \mathbf{a}_t^{\downarrow}, \overline{\mathbf{a}}_1^{\downarrow}, \overline{\mathbf{a}}_2^{\downarrow}, \ldots, \overline{\mathbf{a}}_t^{\downarrow}, \mathbf{a}_1^{\downarrow}, \ldots\right).$$

Легко видеть, что полный цикл подстановки $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ на самом деле имеет вид (2). ■

Следующее утверждение раскрывает детали одного возможного рекурсивного построения полноцикловой пороговой подстановки.

Утверждение 5. Пусть матрица $A_{n\times n}$ определяет полноцикловую пороговую подстановку $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ на множестве $\{\pm1\}^n$:

$$\left(\mathbf{a}_1^{\downarrow},\ldots,\mathbf{a}_{2^{n-1}}^{\downarrow},\overline{\mathbf{a}}_1^{\downarrow},\ldots,\overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\right).$$

Тогда матрица

$$B=\begin{pmatrix} b & b_{11} & \ldots & b_{1n}\\ b_{11} & & & \\ \vdots & & A_{n\times n} & \\ b_{n1} & & & \end{pmatrix}_{(n+1)\times(n+1)}$$

может реализовать полноцикловую пороговую подстановку $\operatorname{sgn}(B\mathbf{x}^{\downarrow})$ вида

$$\left(\begin{pmatrix}*\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}*\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}*\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}*\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}*\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}*\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}*\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}*\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$$

только одного из следующих четырёх типов:

1) $\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

2) $\left(\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

3) $\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

4) $\left(\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$.

Доказательство. Ввиду утверждения 4, полный цикл подстановки $\operatorname{sgn}(B\mathbf{x}^{\downarrow})$ должен иметь вид

$$\left(\begin{pmatrix}a_1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}a_{2^{n-1}}\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}b_1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}b_{2^{n-1}}\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}\overline{a}_1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}\overline{a}_{2^{n-1}}\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}\overline{b}_1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}\overline{b}_{2^{n-1}}\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right).$$

Поскольку все наборы в этом цикле должны быть различны, то на самом деле $a_1=b_1$, $\ldots$, $a_{2^{n-1}}=b_{2^{n-1}}$:

$$\left(\begin{pmatrix}a_1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}a_{2^{n-1}}\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}a_1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}a_{2^{n-1}}\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}\overline{a}_1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}\overline{a}_{2^{n-1}}\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}\overline{a}_1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\ldots,\begin{pmatrix}\overline{a}_{2^{n-1}}\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right).$$

Обозначим $\mathbf{b}=(b_{11},\ldots,b_{1n})$. Наблюдая первую координату в наборах, составляющих полный цикл пороговой подстановки $\operatorname{sgn}(B\mathbf{x}^{\downarrow})$, нетрудно заметить равенства

$$\operatorname{sgn}(ba_i+\mathbf{b}\mathbf{a}_i^{\downarrow})=a_{i+1}=\operatorname{sgn}(ba_i-\mathbf{b}\mathbf{a}_i^{\downarrow}),\quad 1\leqslant i<2^{n-1}.$$

Из этих равенств следует, что $a_{i+1}=\operatorname{sgn}(ba_i)$ при всех $1\leqslant i<2^{n-1}$, и значит, возможны два варианта при $b>0$:

1) $a_1 = a_2 = \ldots = a_{2^{n-1}} = 1$;
2) $a_1 = a_2 = \ldots = a_{2^{n-1}} = -1$,

а также два варианта при $b < 0$:

1) $a_1 = -a_2 = a_3 = \ldots = -a_{2^{n-1}} = 1$;
2) $a_1 = -a_2 = a_3 = \ldots = -a_{2^{n-1}} = -1$.

Утверждение 5 доказано. ■

Покажем, что при некоторых ограничениях на полноцикловое пороговое преобразование $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ множества $\{\pm 1\}^n$ возможно построить продолжение $\operatorname{sgn}(B\mathbf{x}^{\downarrow})$ каждого из четырёх возможных типов, перечисленных в утверждении 5.

Определение 2. Будем называть набор $\mathbf{c} \in \{\pm 1\}^n$ точкой *абсолютного минимума* для порогового отображения $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$, если выполняется условие

$$\forall \mathbf{a} \in \{\pm 1\}^n \quad |A\mathbf{c}^{\downarrow}| \leqslant |A\mathbf{a}^{\downarrow}|,$$

где знак $\leqslant$ означает покоординатное неравенство; $|\mathbf{x}| = \big(|x_1|, \ldots, |x_n|\big)$ — покоординатное вычисление абсолютных значений.

Нетрудно видеть, что для любого $\mathbf{c} \in \{\pm 1\}^n$ выполняется равенство

$$|A\mathbf{c}^{\downarrow}| = |A\overline{\mathbf{c}}^{\downarrow}|.$$

Таким образом, $\mathbf{c} \in \{\pm 1\}^n$ является точкой абсолютного минимума для порогового отображения $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$ в том и только в том случае, когда $\overline{\mathbf{c}}$ является точкой абсолютного минимимума для порогового отображения $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$.

Наборы $\mathbf{c}, \overline{\mathbf{c}} \in \{\pm 1\}^n$ будем называть точками *строгого абсолютного минимума* для порогового отображения $\operatorname{sgn}(A\mathbf{x}^{\downarrow})$, если выполняется условие

$$\forall \mathbf{a} \in \{\pm 1\}^n \setminus \{\mathbf{c}, \overline{\mathbf{c}}\} \quad |A\mathbf{c}^{\downarrow}| < |A\mathbf{a}^{\downarrow}|,$$

где знак $<$ означает строгое неравенство в каждой координате сравниваемых наборов.

Теорема 3. Пусть A_n — целочисленная матрица размера $n \times n$, которая определяет полноцикловую пороговую подстановку $\operatorname{sgn}(A_n\mathbf{x}^{\downarrow})$ на множестве $\{\pm 1\}^n$

$$\big(\mathbf{a}_1, \ldots, \mathbf{a}_{2^{n-1}}, \overline{\mathbf{a}}_1, \ldots, \overline{\mathbf{a}}_{2^{n-1}}\big)$$

с двумя строгими абсолютными минимумами в точках $\mathbf{a}_{2^{n-1}}$ и $\overline{\mathbf{a}}_{2^{n-1}}$:

$$\forall \mathbf{a} \notin \{\mathbf{a}_{2^{n-1}}, \overline{\mathbf{a}}_{2^{n-1}}\} \quad |A_n\mathbf{a}^{\downarrow}| > |A_n\mathbf{a}^{\downarrow}_{2^{n-1}}| = \mathbf{1}^{\downarrow}.$$

Тогда выполняются следующие свойства:

1) матрица $A_{n+1} = \begin{pmatrix} 2n-1 & 2\mathbf{a}_{2^{n-1}} \\ 4\overline{\mathbf{a}}_1^{\downarrow} & 3A_n \end{pmatrix}$ задаёт полноцикловую пороговую подстановку $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$

$$\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$$

с двумя строгими абсолютными минимумами в точках $\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}$ и $\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}$, при которых $\left|A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \left|A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \mathbf{1}^{\downarrow}$;

2) матрица $A_{n+1} = \begin{pmatrix} 2n-1 & 2\overline{\mathbf{a}}_{2^{n-1}} \\ 4\mathbf{a}_1^{\downarrow} & 3A_n \end{pmatrix}$ задаёт полноцикловую пороговую подстановку $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$

$$\left(\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$$

с двумя строгими абсолютными минимумами в точках $\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}$ и $\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}$, при которых $\left|A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \left|A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \mathbf{1}^{\downarrow}$;

3) матрица $A_{n+1} = \begin{pmatrix} -2n+1 & 2\mathbf{a}_{2^{n-1}} \\ 4\mathbf{a}_1^{\downarrow} & 3A_n \end{pmatrix}$ задаёт полноцикловую пороговую подстановку $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$

$$\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$$

с двумя строгими абсолютными минимумами в точках $\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}$ и $\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}$, при которых $\left|A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \left|A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \mathbf{1}^{\downarrow}$;

4) матрица $A_{n+1} = \begin{pmatrix} -2n+1 & 2\overline{\mathbf{a}}_{2^{n-1}} \\ 4\overline{\mathbf{a}}_1^{\downarrow} & 3A_n \end{pmatrix}$ задаёт полноцикловую пороговую подстановку $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$

$$\left(\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix},\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix},\dots,\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$$

с двумя строгими абсолютными минимумами в точках $\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}$ и $\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}$, при которых $\left|A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \left|A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right| = \mathbf{1}^{\downarrow}$.

Доказательство. Заметим, что из целочисленности матрицы A_n и условия

$$\forall \mathbf{a} \notin \{\mathbf{a}_{2^{n-1}}, \overline{\mathbf{a}}_{2^{n-1}}\} \quad |A_n\mathbf{a}^{\downarrow}| > |A_n\mathbf{a}_{2^{n-1}}^{\downarrow}| = (1,\dots,1)^{\mathrm{T}}$$

следует, что

$$\forall \mathbf{a} \notin \{\mathbf{a}_{2^{n-1}}, \overline{\mathbf{a}}_{2^{n-1}}\} \quad |A_n\mathbf{a}^{\downarrow}| \geqslant (2,\dots,2)^{\mathrm{T}}, \quad A_n\mathbf{a}_{2^{n-1}}^{\downarrow} = \overline{\mathbf{a}}_1^{\downarrow}.$$

Теперь доказательство всех свойств проводится непосредственной проверкой:
1)

$$A_{n+1}\begin{pmatrix}\pm 1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix} = \begin{pmatrix}\pm(2n-1) + 2\mathbf{a}_{2^{n-1}}\mathbf{a}_i^{\downarrow}\\ \pm 4\overline{\mathbf{a}}_1^{\downarrow} + 3A_n\mathbf{a}_i^{\downarrow}\end{pmatrix} \Rightarrow \begin{cases} \operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm 1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right) = \begin{pmatrix}\pm 1\\ \mathbf{a}_{i+1}^{\downarrow}\end{pmatrix}, \\ \left|A_{n+1}\begin{pmatrix}\pm 1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right| \geqslant (3,2,\dots,2)^{\mathrm{T}}, \end{cases} \quad 1 \leqslant i < 2^{n-1},$$

$$A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\pm(2n-1)+2\mathbf{a}_{2^{n-1}}\overline{\mathbf{a}}_i^{\downarrow}\\ \pm4\overline{\mathbf{a}}_1^{\downarrow}+3A_n\overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}4n-1\\ 7\overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-4n+1\\ 7\mathbf{a}_1^{\downarrow}\end{pmatrix},$$

$$A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}.$$

2)

$$A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\pm(2n-1)+2\overline{\mathbf{a}}_{2^{n-1}}\mathbf{a}_i^{\downarrow}\\ \pm4\mathbf{a}_1^{\downarrow}+3A_n\mathbf{a}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\pm1\\ \mathbf{a}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\pm(2n-1)+2\overline{\mathbf{a}}_{2^{n-1}}\overline{\mathbf{a}}_i^{\downarrow}\\ \pm4\mathbf{a}_1^{\downarrow}+3A_n\overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-4n+1\\ 7\overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}4n-1\\ 7\mathbf{a}_1^{\downarrow}\end{pmatrix},$$

$$A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}.$$

3)

$$A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\mp(2n-1)+2\mathbf{a}_{2^{n-1}}\mathbf{a}_i^{\downarrow}\\ \pm4\mathbf{a}_1^{\downarrow}+3A_n\mathbf{a}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\mp1\\ \mathbf{a}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\mp(2n-1)+2\mathbf{a}_{2^{n-1}}\overline{\mathbf{a}}_i^{\downarrow}\\ \pm4\mathbf{a}_1^{\downarrow}+3A_n\overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\mp1\\ \overline{\mathbf{a}}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}4n-1\\ 7\overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-4n+1\\ 7\mathbf{a}_1^{\downarrow}\end{pmatrix},$$

$$A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}.$$

4)

$$A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\mp(2n-1)+2\overline{\mathbf{a}}_{2^{n-1}}\mathbf{a}_i^{\downarrow}\\ \pm4\overline{\mathbf{a}}_1^{\downarrow}+3A_n\mathbf{a}_i^{\downarrow}\end{pmatrix}\Rightarrow\begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\mp1\\ \mathbf{a}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm1\\ \mathbf{a}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}\pm 1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}=\begin{pmatrix}\mp(2n-1)+2\overline{\mathbf{a}}_{2^{n-1}}\overline{\mathbf{a}}_i^{\downarrow}\\ \pm 4\overline{\mathbf{a}}_1^{\downarrow}+3A_n\overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\ \Rightarrow\ \begin{cases}\operatorname{sgn}\left(A_{n+1}\begin{pmatrix}\pm 1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right)=\begin{pmatrix}\mp 1\\ \overline{\mathbf{a}}_{i+1}^{\downarrow}\end{pmatrix},\\ \left|A_{n+1}\begin{pmatrix}\pm 1\\ \overline{\mathbf{a}}_i^{\downarrow}\end{pmatrix}\right|\geqslant(3,2,\ldots,2)^{\mathrm{T}},\end{cases}\quad 1\leqslant i<2^{n-1},$$

$$A_{n+1}\begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-4n+1\\ 7\overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}-1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}4n-1\\ 7\mathbf{a}_1^{\downarrow}\end{pmatrix},$$

$$A_{n+1}\begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix},\qquad A_{n+1}\begin{pmatrix}-1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}=\begin{pmatrix}-1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}.$$

Теорема 3 доказана. ■

Замечание 3. В условии теоремы 3 целочисленность матрицы A_n, а также нормированное значение абсолютного минимума $\mathbf{1}^{\downarrow}=(1,\ldots,1)^{\mathrm{T}}$ используются исключительно для лаконичности формулировки и простоты доказательства. Так, например, в общем случае для произвольной матрицы A_n размера $n\times n$, что определяет полноцикловую пороговую подстановку $\operatorname{sgn}(A_n\mathbf{x}^{\downarrow})$ на множестве $\{\pm 1\}^n$

$$\left(\mathbf{a}_1,\ldots,\mathbf{a}_{2^{n-1}},\overline{\mathbf{a}}_1,\ldots,\overline{\mathbf{a}}_{2^{n-1}}\right)$$

с двумя строгими абсолютными минимумами в точках $\mathbf{a}_{2^{n-1}}$ и $\overline{\mathbf{a}}_{2^{n-1}}$:

$$\exists\varepsilon>0\quad\forall\mathbf{a}\notin\{\mathbf{a}_{2^{n-1}},\overline{\mathbf{a}}_{2^{n-1}}\}\quad|A_n\mathbf{a}^{\downarrow}|>(1+\varepsilon)|A_n\mathbf{a}_{2^{n-1}}^{\downarrow}|,$$

при построении полноциклового продолжения $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$ первого типа можно использовать матрицу

$$A_{n+1}=\begin{pmatrix}2n-1 & 2\mathbf{a}_{2^{n-1}}\\ -(1+\varepsilon/2)A_n\overline{\mathbf{a}}_{2^{n-1}}^{\downarrow} & A_n\end{pmatrix}.$$

Нетрудно видеть, что предложенная в теореме 3 конструкция построения полноцикловой пороговой подстановки из полноцикловой пороговой подстановки меньшей размерности в любом из четырёх перечисленных вариантов допускает рекурсивное применение.

Следствие 6. Для любого натурального n существует полноцикловая пороговая подстановка множества $\{\pm 1\}^n$.

Доказательство. Матрица $A_1=(-1)_{1\times 1}$ задаёт полноцикловую пороговую подстановку с двумя точками строгого абсолютного минимума. Применяя к A_1 рекурсивную процедуру, описанную в теореме 3, можно для любого n построить матрицу A_n размера $n\times n$, которая задаёт полноцикловую пороговую подстановку $\operatorname{sgn}(A_n\mathbf{x}^{\downarrow})$. ■

На практике при использовании полноцикловых подстановок часто требуется, чтобы обращение данной подстановки также допускало эффективное вычисление. Оказывается, для пороговых подстановок, которые построены рекурсивным образом в соответствии с теоремой 3, обратные подстановки тоже являются пороговыми и могут быть построены рекурсивным образом.

Следствие 7. В условиях теоремы 3 пусть $\hat{A}_n$ — целочисленная матрица размера $n\times n$, которая определяет обратную к $\operatorname{sgn}(A_n\mathbf{x}^{\downarrow})$ полноцикловую пороговую подстановку

$$\left(\overline{\mathbf{a}}_{2^{n-1}}^{\downarrow},\ldots,\overline{\mathbf{a}}_1^{\downarrow},\mathbf{a}_{2^{n-1}}^{\downarrow},\ldots,\mathbf{a}_1^{\downarrow}\right)$$

с двумя строгими абсолютными минимумами в точках $\overline{\mathbf{a}}_1$ и $\mathbf{a}_1$:

$$\forall \mathbf{a} \notin \{\mathbf{a}_1, \overline{\mathbf{a}}_1\} \quad |\hat{A}_n \mathbf{a}^{\downarrow}| > |\hat{A}_n \mathbf{a}_1^{\downarrow}| = \mathbf{1}^{\downarrow}.$$

Тогда подстановка, обратная к пороговой подстановке $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$, также является пороговой и в каждом из четырёх перечисленных в теореме 3 случаев может быть задана соответствующей матрицей $\hat{A}_{n+1}$:

1) матрица $\hat{A}_{n+1}=\begin{pmatrix} 2n-1 & 2\overline{\mathbf{a}}_1 \\ 4\mathbf{a}^{\downarrow}_{2^{n-1}} & 3\hat{A}_n \end{pmatrix}$ задаёт подстановку $\operatorname{sgn}(\hat{A}_{n+1}\mathbf{x}^{\downarrow})=\big(\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})\big)^{-1}$ с двумя строгими абсолютными минимумами в точках $\begin{pmatrix} 1 \\ \mathbf{a}_1^{\downarrow} \end{pmatrix}$ и $\begin{pmatrix} -1 \\ \overline{\mathbf{a}}_1^{\downarrow} \end{pmatrix}$;

2) матрица $\hat{A}_{n+1}=\begin{pmatrix} 2n-1 & 2\mathbf{a}_1 \\ 4\overline{\mathbf{a}}^{\downarrow}_{2^{n-1}} & 3\hat{A}_n \end{pmatrix}$ задаёт подстановку $\operatorname{sgn}(\hat{A}_{n+1}\mathbf{x}^{\downarrow})=\big(\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})\big)^{-1}$ с двумя строгими абсолютными минимумами в точках $\begin{pmatrix} -1 \\ \mathbf{a}_1^{\downarrow} \end{pmatrix}$ и $\begin{pmatrix} 1 \\ \overline{\mathbf{a}}_1^{\downarrow} \end{pmatrix}$;

3) матрица $\hat{A}_{n+1}=\begin{pmatrix} -2n+1 & 2\mathbf{a}_1 \\ 4\mathbf{a}^{\downarrow}_{2^{n-1}} & 3\hat{A}_n \end{pmatrix}$ задаёт подстановку $\operatorname{sgn}(\hat{A}_{n+1}\mathbf{x}^{\downarrow})=\big(\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})\big)^{-1}$ с двумя строгими абсолютными минимумами в точках $\begin{pmatrix} 1 \\ \mathbf{a}_1^{\downarrow} \end{pmatrix}$ и $\begin{pmatrix} -1 \\ \overline{\mathbf{a}}_1^{\downarrow} \end{pmatrix}$;

4) матрица $\hat{A}_{n+1}=\begin{pmatrix} -2n+1 & 2\overline{\mathbf{a}}_1 \\ 4\overline{\mathbf{a}}^{\downarrow}_{2^{n-1}} & 3\hat{A}_n \end{pmatrix}$ задаёт подстановку $\operatorname{sgn}(\hat{A}_{n+1}\mathbf{x}^{\downarrow})=\big(\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})\big)^{-1}$ с двумя строгими абсолютными минимумами в точках $\begin{pmatrix} -1 \\ \mathbf{a}_1^{\downarrow} \end{pmatrix}$ и $\begin{pmatrix} 1 \\ \overline{\mathbf{a}}_1^{\downarrow} \end{pmatrix}$.

Доказательство. Согласно теореме 3, продолжение матрицы $\hat{A}_n$ до целочисленной матрицы $\hat{A}_{n+1}$ позволяет реализовать полный цикл каждого из четырёх возможных типов с двумя строгими абсолютными минимумами.

Нетрудно видеть, что для полного цикла $\operatorname{sgn}(A_{n+1}\mathbf{x}^{\downarrow})$ типа 1 (2, 3 или 4) обратной будет подстановка $\operatorname{sgn}(\hat{A}_{n+1}\mathbf{x}^{\downarrow})$, полученная в результате продолжения типа 1 (2, 4 и 3 соответственно). ■

Проиллюстрируем результаты теоремы 3 и следствий 6, 7 на примере.

Пример 3. Рассмотрим пороговую подстановку с матрицей $A_1 = (-1)_{1\times 1}$. Очевидно, подстановка $\operatorname{sgn}(A_1\mathbf{x}^{\downarrow})$ является полноцикловой с двумя точками строго абсолютного минимума, а матрица $\hat{A}_1 = (-1)_{1\times 1}$ задаёт обратную подстановку.

Для матриц A_1 и $\hat{A}_1$ выполним рекурсивное продолжение типа 1:

$$A_2 = \begin{pmatrix} 1 & 2 \\ -4 & -3 \end{pmatrix}, \qquad \hat{A}_2 = \begin{pmatrix} 1 & -2 \\ 4 & -3 \end{pmatrix}.$$

Полученные матрицы реализуют полные циклы:

$$A_2\colon \begin{pmatrix} 1 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ -1 \end{pmatrix} \mapsto \begin{pmatrix} -1 \\ -1 \end{pmatrix} \mapsto \begin{pmatrix} -1 \\ 1 \end{pmatrix},$$
$$\hat{A}_2\colon \begin{pmatrix} -1 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} -1 \\ -1 \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ -1 \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Теперь для матриц A_2 и $\hat{A}_2$ выполним рекурсивное продолжение типа 2:

$$A_3 = \begin{pmatrix} 3 & -2 & 2 \\ 4 & 3 & 6 \\ 4 & -12 & -9 \end{pmatrix}, \qquad \hat{A}_3 = \begin{pmatrix} 3 & 2 & 2 \\ -4 & 3 & -6 \\ 4 & 12 & -9 \end{pmatrix}.$$

Построенные матрицы реализуют полные циклы:

$$A_3\colon \begin{pmatrix}-1\\1\\1\end{pmatrix}\mapsto\begin{pmatrix}-1\\1\\-1\end{pmatrix}\mapsto\begin{pmatrix}-1\\-1\\-1\end{pmatrix}\mapsto\begin{pmatrix}-1\\-1\\1\end{pmatrix}\mapsto\begin{pmatrix}1\\-1\\-1\end{pmatrix}\mapsto\begin{pmatrix}1\\-1\\1\end{pmatrix}\mapsto\begin{pmatrix}1\\1\\1\end{pmatrix}\mapsto\begin{pmatrix}1\\1\\-1\end{pmatrix},$$

$$\hat{A}_3\colon \begin{pmatrix}1\\1\\-1\end{pmatrix}\mapsto\begin{pmatrix}1\\1\\1\end{pmatrix}\mapsto\begin{pmatrix}1\\-1\\1\end{pmatrix}\mapsto\begin{pmatrix}1\\-1\\-1\end{pmatrix}\mapsto\begin{pmatrix}-1\\-1\\1\end{pmatrix}\mapsto\begin{pmatrix}-1\\-1\\-1\end{pmatrix}\mapsto\begin{pmatrix}-1\\1\\-1\end{pmatrix}\mapsto\begin{pmatrix}-1\\1\\1\end{pmatrix}.$$

Для матриц A_3 и $\hat{A}_3$ выполним рекурсивное продолжение типа 3:

$$A_4=\begin{pmatrix}-5&-2&-2&2\\-4&9&-6&6\\4&12&9&18\\4&12&-36&-27\end{pmatrix},\qquad \hat{A}_4=\begin{pmatrix}-5&-2&2&2\\-4&9&6&6\\-4&-12&9&-18\\4&12&36&-27\end{pmatrix}.$$

И так далее.

4. Применение построенных полноцикловых пороговых подстановок

Исследуем строение координатных функций всех возможных полноцикловых пороговых подстановок, которые могут быть построены рекурсивным образом посредством многократного применения конструкции, предложенной в теореме 3, с тривиальным начальным условием, а также сделаем вывод о возможности применения указанных подстановок на практике. Описание координатных функций проведём в хорошо развитой терминологии линейных рекуррентных последовательностей над полем (подробно ознакомиться с этим разделом математики можно в монографии [14], ставшей мировым стандартом в данной области, а также в [15]).

Напомним, что над произвольным полем последовательность со знаками

$$\underbrace{0,\dots,0}_{k},\binom{k}{k}\alpha^0,\binom{k+1}{k}\alpha^1,\binom{k+2}{k}\alpha^2,\dots$$

называют *биномиальной последовательностью порядка k с корнем α* и обозначают через $\alpha^{[k]}$. Для знаков последовательности $\alpha^{[k]}$ удобно использовать универсальную формулу $\alpha^{[k]}(i)=\binom{i}{k}\alpha^{i-k}$, $i\in\mathbb{N}_0$, полагая $\alpha^{[k]}(i)=0$ при $i<k$. Биномиальная последовательность $\alpha^{[k]}$ является линейной рекуррентой с минимальным многочленом $(x-\alpha)^{k+1}$.

В последовательности двоичных наборов, упорядоченных лексикографически:

$$\begin{matrix}v_1\colon\\v_2\colon\\\vdots\\v_{n-1}\colon\\v_n\colon\end{matrix}\quad\begin{pmatrix}0\\0\\\vdots\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\\vdots\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\\vdots\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\\vdots\\1\\1\end{pmatrix},\dots\begin{pmatrix}0\\1\\\vdots\\1\\1\end{pmatrix},\begin{pmatrix}1\\0\\\vdots\\0\\0\end{pmatrix},\begin{pmatrix}1\\0\\\vdots\\0\\1\end{pmatrix},\begin{pmatrix}1\\0\\\vdots\\1\\0\end{pmatrix},\dots\begin{pmatrix}1\\1\\\vdots\\1\\1\end{pmatrix},\qquad(3)$$

координатные последовательности являются начальными отрезками биномиальных последовательностей над полем $\mathbb{F}_2$:

$$v_1=1^{[2^{n-1}]}(\overline{0,2^n-1}),v_2=1^{[2^{n-2}]}(\overline{0,2^n-1}),\dots,v_{n-1}=1^{[2^1]}(\overline{0,2^n-1}),v_n=1^{[2^0]}(\overline{0,2^n-1}).$$

Заметим, что указанные отрезки явным образом содержат всю информацию о последовательностях $1^{[2^0]}, 1^{[2^1]}, \ldots, 1^{[2^{n-2}]}, 1^{[2^{n-1}]}$ (2^n — общий период для данных последовательностей), и потому вполне естественно использовать для отрезков

$$1^{[2^0]}(\overline{0, 2^n-1}),\ 1^{[2^1]}(\overline{0, 2^n-1}),\ \ldots,\ 1^{[2^{n-2}]}(\overline{0, 2^n-1}),\ 1^{[2^{n-1}]}(\overline{0, 2^n-1})$$

лаконичные обозначения $1^{[2^0]}, 1^{[2^1]}, \ldots, 1^{[2^{n-2}]}, 1^{[2^{n-1}]}$ — это не приведёт к путанице, поскольку в данной работе рассматриваются только конечные последовательности.

Для представления координатных функций исследуемых полноцикловых пороговых подстановок в виде линейных рекуррентных последовательностей необходимо перейти к стандартному булеву представлению — для этого (-1) заменим на 0, а 1 оставим без изменений. Таким образом, предложенные в теореме 3 варианты рекурсивного продолжения полноцикловых подстановок в булевом представлении будут иметь вид

1) $\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

2) $\left(\begin{pmatrix}0\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

3) $\left(\begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$;

4) $\left(\begin{pmatrix}0\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}0\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}1\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \overline{\mathbf{a}}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\end{pmatrix}, \begin{pmatrix}1\\ \mathbf{a}_1^{\downarrow}\end{pmatrix}, \ldots, \begin{pmatrix}0\\ \mathbf{a}_{2^{n-1}}^{\downarrow}\end{pmatrix}\right)$

(черта сверху в данном случае означает стандартное отрицание в булевой логике).

Теорема 4. Пусть $u_1, \ldots, u_n$ — координатные последовательности булевого представления полноцикловой подстановки $\operatorname{sgn}(A_n\mathbf{x}^{\downarrow})$, построенной рекурсивным образом посредством $(n-1)$-кратного применения конструкций, предложенных в теореме 3. Тогда над полем $\mathbb{F}_2$ справедливы разложения

$$\begin{array}{rccccccccc}
u_1 = & 1^{[2^{n-1}]} & & + & & & + a_1 1^{[2^0]} & + b_1 1^{[0]},\\
u_2 = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & & & + a_2 1^{[2^0]} & + b_2 1^{[0]},\\
\ldots & & & & & & & & \\
u_{n-2} = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \ldots & + 1^{[2^2]} & + a_{n-2} 1^{[2^0]} & + b_{n-2} 1^{[0]},\\
u_{n-1} = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \ldots & + 1^{[2^2]} + 1^{[2^1]} & & + b_{n-1} 1^{[0]},\\
u_n = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \ldots & + 1^{[2^2]} + 1^{[2^1]} & + 1^{[2^0]} & + b_n 1^{[0]},
\end{array}$$

в которых коэффициенты $a_1, \ldots, a_{n-2}, b_1, \ldots, b_n \in \mathbb{F}_2$ зависят от выбора типа рекурсивного продолжения на соответствующем шаге построения.

Доказательство. Проведём индукцию по n.

Б а з а при $n = 1$ очевидна: существует единственная полноцикловая подстановка на множестве $\{0, 1\}$ и её цикл может быть записан двумя способами: $(0, 1)$ и $(1, 0)$ — последовательности с разложениями $1^{[2^0]}$ и $1^{[2^0]} + 1^{[0]}$ соответственно.

Ш а г и н д у к ц и и при $n \geqslant 2$. Предположим, что имеется полноцикловая пороговая подстановка $\left(\mathbf{a}_1^{\downarrow}, \ldots, \mathbf{a}_{2^{n-2}}^{\downarrow}, \overline{\mathbf{a}}_1^{\downarrow}, \ldots, \overline{\mathbf{a}}_{2^{n-2}}^{\downarrow}\right)$ в булевом представлении с координатными последовательностями $u'_2, \ldots, u'_n$. Нетрудно видеть, что применение к данной подстановке какого-либо из вариантов рекурсивного продолжения, предложенных в теореме 3, приведёт к полноцикловой подстановке с координатными последовательностями

$$u_1 = 1^{[2^{n-1}]} + a\,1^{[2^0]} + b\,1^{[0]}, \quad u_2 = 1^{[2^{n-1}]} + u'_2, \quad \ldots, \quad u_n = 1^{[2^{n-1}]} + u'_n,$$

где коэффициенты $a, b \in \mathbb{F}_2$ зависят от выбранного способа рекурсивного продолжения. Здесь стоит отметить, что при $n = 2$ совпадают рекурсивные способы продолжения 1 и 3 (а также 2 и 4), поэтому коэффициент a обязательно равен 0. Для завершения шага индукции остаётся воспользоваться предположением индукции о строении координатных последовательностей $u'_2, \ldots, u'_n$. ■

Следствие 8. Рекурсивное $(n-1)$-кратное применение конструкций, предложенных в теореме 3, позволяет построить 2^{2n-3} различных полноцикловых пороговых подстановок множества $\{0,1\}^n$.

Доказательство. Существует единственная полноцикловая пороговая подстановка на множестве $\{0,1\}$. Применим к ней все возможные $(n-1)$-кратные комбинации рекурсивных продолжений, рассмотренных в теореме 3. В результате получим полноцикловые пороговые подстановки на множестве $\{0,1\}^n$, координатные функции которых обладают всеми возможными биномиальными разложениями, указанными в условии теоремы 4 — всего 2^{2n-2} различных вариантов. Однако необходимо отметить, что каждая полноцикловая пороговая подстановка, построенная в результате рекурсивных продолжений из теоремы 3, обладает двумя точками строгого абсолютного минимума и потому допускает два способа записи в соответствии с представлением из условия теоремы 3: $\left(\mathbf{a}_1^{\downarrow}, \ldots, \mathbf{a}_{2^{n-1}}^{\downarrow}, \overline{\mathbf{a}}_1^{\downarrow}, \ldots, \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}\right)$ и $\left(\overline{\mathbf{a}}_1^{\downarrow}, \ldots, \overline{\mathbf{a}}_{2^{n-1}}^{\downarrow}, \mathbf{a}_1^{\downarrow}, \ldots, \mathbf{a}_{2^{n-1}}^{\downarrow}\right)$. Значит, каждая такая подстановка может быть получена в результате применения двух $(n-1)$-кратных комбинаций рекурсивных продолжений, поэтому общее количество полноцикловых пороговых подстановок, которые возможно построить с применением результата теоремы 3, равно 2^{2n-3}. ■

В дополнение к результату следствия 8 заметим, что построенные с использованием теоремы 3 полноцикловые пороговые подстановки можно сопрягать подстановками из группы Джевонса — результатом такого действия будут полноцикловые пороговые подстановки. При этом из описания координатных функций, полученного в теореме 4, легко видеть, что перестановка координат обладает точным действием, в то время как инверсии не добавляют ни одного нового варианта (инверсия i-й координаты приводит к замене слагаемого $b_i 1^{[0]}$ в разложении i-й координатной последовательности на слагаемое $\overline{b}_i 1^{[0]}$). Значит, $n!2^{2n-3}$ — общее количество различных полноцикловых пороговых подстановок, которые можно получить в результате сопряжений элементами группы Джевонса множества всех полноцикловых пороговых подстановок, построенных в соответствии с теоремой 4. Однако, как показывает следующий результат, вся эта «арифметика» не имеет никакого смысла с точки зрения практического применения построенных подстановок.

Следствие 9. Произвольная полноцикловая пороговая подстановка, построенная рекурсивным образом посредством $(n-1)$-кратного применения конструкций, предложенных в теореме 3, в булевом представлении афинно эквивалентна полноцикловой подстановке, определяемой лексикографическим упорядочением векторов (3).

Доказательство. Пусть $u_1, \ldots, u_n$ — координатные последовательности булевого представления полноцикловой пороговой подстановки, построенной рекурсивным образом посредством $(n-1)$-кратного применения конструкций, предложенных в теореме 3. Нетрудно видеть, что система равенств

$$
\begin{array}{rllllllll}
u_1 = & 1^{[2^{n-1}]} & & + & & & + a_1 1^{[2^0]} & + b_1 1^{[0]}, \\
u_2 = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & & & + a_2 1^{[2^0]} & + b_2 1^{[0]}, \\
\dots & & & & & & & & \\
u_{n-2} = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \dots & + 1^{[2^2]} & + a_{n-2} 1^{[2^0]} & + b_{n-2} 1^{[0]}, \\
u_{n-1} = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \dots & + 1^{[2^2]} + 1^{[2^1]} & & + b_{n-1} 1^{[0]}, \\
u_n = & 1^{[2^{n-1}]} & + 1^{[2^{n-2}]} & + & \dots & + 1^{[2^2]} + 1^{[2^1]} & + 1^{[2^0]} & + b_n 1^{[0]}
\end{array}
$$

линейно эквивалентна над полем $\mathbb{F}_2$ системе

$$
\begin{array}{rlllll}
u'_1 = & 1^{[2^{n-1}]} & + & a_1 1^{[2^0]} & + & b_1 1^{[0]}, \\
u'_2 = & 1^{[2^{n-2}]} & + & (a_1 + a_2) 1^{[2^0]} & + & (b_1 + b_2) 1^{[0]}, \\
\dots & & & & & \\
u'_{n-2} = & 1^{[2^2]} & + & (a_{n-3} + a_{n-2}) 1^{[2^0]} & + & (b_{n-3} + b_{n-2}) 1^{[0]}, \\
u'_{n-1} = & 1^{[2^1]} & + & a_{n-2} 1^{[2^0]} & + & (b_{n-2} + b_{n-1}) 1^{[0]}, \\
u'_n = & & & 1^{[2^0]} & + & (b_{n-1} + b_n) 1^{[0]},
\end{array}
$$

которая, в свою очередь, линейно эквивалентна системе соотношений

$$
\begin{array}{rlll}
u''_1 = & 1^{[2^{n-1}]} & + & (b_1 + a_1(b_{n-1} + b_n)) 1^{[0]}, \\
u''_2 = & 1^{[2^{n-2}]} & + & (b_1 + b_2 + (a_1 + a_2)(b_{n-1} + b_n)) 1^{[0]}, \\
\dots & & & \\
u''_{n-2} = & 1^{[2^2]} & + & (b_{n-3} + b_{n-2} + (a_{n-3} + b_{n-2})(b_{n-1} + b_n)) 1^{[0]}, \\
u''_{n-1} = & 1^{[2^1]} & + & (b_{n-2} + b_{n-1} + a_{n-2}(b_{n-1} + b_n)) 1^{[0]}, \\
u''_n = & 1^{[2^0]} & + & (b_{n-1} + b_n) 1^{[0]}.
\end{array}
$$

Для завершения доказательства остаётся заметить, что биномиальная последовательность $1^{[0]}$ по существу является константой 1. ■

Итак, согласно следствию 9, все полноцикловые пороговые подстановки, которые можно получить рекурсивным образом посредством $(n-1)$-кратного применения конструкций, предложенных в теореме 3, аффинно эквивалентны счётчиковой последовательности (3) и соответственно также «провальны» по ряду важных криптографических характеристик (алгебраическая степень, нелинейность и др.) [11]. Следовательно, в криптографических приложениях указанные полноцикловые подстановки имеет смысл использовать исключительно для генерации первичных или управляющих последовательностей, к которым обязательно будет применено усложнение.

ЛИТЕРАТУРА

1. *Дертоузос М.* Пороговая логика. М.: Мир, 1967.
2. *Бутаков Е. А.* Методы синтеза релейных устройств из пороговых элементов. М.: Энергия, 1970.
3. *Зуев Ю. А.* Пороговые функции и пороговые представления булевых функций // Математические вопросы кибернетики. Вып. 5. М.: Наука, 1994.
4. *Никонов В. Г., Сидоров Е. С.* О способе построения взаимно однозначных отображений при помощи квазиадамаровых матриц // Лесной вестник. 2009. № 2. С. 155–158.
5. *Никонов В. Г., Литвиненко В. С.* Геометрический подход к доказательству биективности одного координатно-порогового отображения // Comp. Nanotechnol. 2015. No. 1. P. 26–31.
6. *Никонов В. Г., Литвиненко В. С.* О биективности преобразований, задаваемых квазиадамаровыми матрицами // Comp. Nanotechnol. 2016. No. 1. P. 6–13.

7. *Никонов В. Г., Кононов С. А.* О некоторых свойствах квазиадамаровых матриц, задающих биективные преобразования // Comp. Nanotechnol. 2022. V. 9. No. 1. P. 32–38.
8. *Никонов В. Г., Зобов А. И.* Построение обратимого полноциклового преобразования в пороговом базисе // Comp. Nanotechnol. 2023. V. 10. No. 2. P. 36–41.
9. *Шурупов А. Н.* Критерии функциональной разделимости квадратичных булевых пороговых функций // Прикладная дискретная математика. 2015. № 2(28). С. 37–45.
10. *Шурупов А. Н.* Некоторые структурные свойства квадратичных булевых пороговых функций // Прикладная дискретная математика. Приложение. 2015. № 8. С. 48–51.
11. *Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В.* Булевы функции в теории кодирования и криптологии. 2-е изд. М.: МЦНМО, 2012. 584 с.
12. *Garey M. R. and Johnson D. S.* Computers and Intractability: A Guide to the Theory of NP-Completeness. N.Y.: W. H. Freeman and Company, 1979.
13. *MacWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. Amsterdam: Elsevier, 1977.
14. *Лидл Р., Нидерайтер Г.* Конечные поля. Т. 1. М.: Мир, 1988. 430 с.
15. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра: учебник для вузов. 5-е изд. СПб.: Лань, 2024. 608 с.

REFERENCES

1. *Dertouzos M. L.* Threshold Logic. Cambridge, MIT Press, 1965.
2. *Butakov E. A.* Metody sinteza releynykh ustroystv iz porogovykh elementov [Methods of Synthesis of Relay Devices from Threshold Elements]. Moscow, Energiya, 1970. (in Russian)
3. *Zuev Yu. A.* Porogovye funktsii i porogovye predstavleniya bulevykh funktsiy [Threshold functions and threshold representations of Boolean functions]. Matematicheskie Voprosy Kibernetiki, iss. 5. Moscow, Nauka, 1994. (in Russian)
4. *Nikonov V. G. and Sidorov E. S.* O sposobe postroeniya vzaimno odnoznachnykh otobrazheniy pri pomoshchi kvaziadamarovykh matrits [On a method for constructing bijective mappings using quasi-Hadamard matrices]. Lesnoy Vestnik, 2009, no. 2, pp. 155–158. (in Russian)
5. *Nikonov V. G. and Litvinenko V. S.* Geometricheskiy podkhod k dokazatel'stvu biektivnosti odnogo koordinatno-porogovogo otobrazheniya [Geometrical approach to the argumentum of bijection of one coordinate-threshold reflection]. Comp. Nanotechnol., 2015, no. 1, pp. 26–31. (in Russian)
6. *Nikonov V. G. and Litvinenko V. S.* O biektivnosti preobrazovaniy, zadavaemykh kvaziadamarovymi matritsami [About bijectivity of transformations determined by quasi-Hadamard matrixes]. Comp. Nanotechnol., 2016, no. 1, pp. 6–13. (in Russian)
7. *Nikonov V. G. and Kononov S. A.* O nekotorykh svoystvakh kvaziadamarovykh matrits, zadayushchikh biektivnye preobrazovaniya [About some properties of quasi-hadamard matrices defining bijective transformations]. Comp. Nanotechnol., 2022, vol. 9, no. 1, pp. 32–38. (in Russian)
8. *Nikonov V. G. and Zobov A. I.* Postroenie obratimogo polnotsiklovogo preobrazovaniya v porogovom bazise [Construction of a reversible full-cycle transformation in a threshold basis]. Comp. Nanotechnol., 2023, vol. 10, no. 2, pp. 36–41. (in Russian)
9. *Shurupov A. N.* Kriterii funktsional'noy razdelimosti kvadratichnykh bulevykh porogovykh funktsiy [Functional decomposability criteria for quadratic threshold Boolean functions]. Prikladnaya Diskretnaya Matematika, 2015, no. 2(28), pp. 37–45. (in Russian)
10. *Shurupov A. N.* Nekotorye strukturnye svoystva kvadratichnykh bulevykh porogovykh funktsiy [Some structural properties of quadratic Boolean threshold functions]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2015, no. 8, pp. 48–51. (in Russian)

11. *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)
12. *Garey M. R. and Johnson D. S.* Computers and Intractability: A Guide to the Theory of NP-Completeness. N.Y., W. H. Freeman and Company, 1979.
13. *MacWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. Amsterdam, Elsevier, 1977.
14. *Lidl R. and Niederreiter H.* Finite Fields, 2nd ed. Cambridge, Cambridge University Press, 1996.
15. *Glukhov M. M., Elizarov V. P., and Nechaev A. A.* Algebra [Algebra]. Saint Petersburg, Lan Publ., 2024. 608 p. (in Russian)

УДК 519.716.5 DOI 10.17223/20710410/71/2

ОБ УРОВНЕ СИЛЬНОЙ АФФИННОСТИ БУЛЕВЫХ ФУНКЦИЙ

А. В. Кулагин*, А. В. Тарасов**

*ООО «Центр сертификационных исследований», г. Москва, Россия
**АО «ИТМиВТ», г. Москва, Россия

E-mail: artemcoolag@yandex.ru, alextar1@mail.ru

Изучается такой параметр булевых функций, как уровень сильной аффинности, равный минимальному числу переменных, фиксация которых любыми значениями даёт аффинную функцию. Исследованы основные свойства уровня сильной аффинности и его связь с другими параметрами булевых функций. Доказана асимптотическая максимальность уровня сильной аффинности булевых функций.

Ключевые слова: *булева функция, уровень сильной аффинности, спектральные характеристики, алгебраическая степень, вес, алгебраическая иммунность.*

ON THE STRONG AFFINITY LEVEL OF BOOLEAN FUNCTIONS

A. V. Kulagin*, A. V. Tarasov**

*LLC "Center of Certification Research", Moscow, Russia
**JSC "IPMCE", Moscow, Russia

The paper is devoted to the study of such a parameter of a Boolean function as the strong affinity level $\mathrm{la}_s(f)$, which is equal to the minimum number of variables whose fixation by any values gives an affine function. A criterion for finding the exact value of a strong affinity level has been obtained by searching for the emptiest subgraph in the graph of a Boolean function. A correlation has been found between the level of strong affinity and other parameters of Boolean functions, such as algebraic degree $\deg(f)$, weight $||f||$, and algebraic immunity $\mathrm{Al}(f)$, which are as follows: $\mathrm{la}_s(f) \geqslant \mathrm{Al}(f)$ if the function f is not balanced or does not contain first-degree monomials; $\mathrm{la}_s(f) \geqslant \deg(f) - 1$; $2^{n-\mathrm{la}_s(f)-1} \leqslant ||f|| \leqslant 2^n - 2^{n-\mathrm{la}_s(f)-1}$. It has been proven that symmetric Boolean functions and monotonic Boolean functions that significantly depend on all their variables have the highest possible strong affinity level. Asymptotic maximality of the strong affinity level of Boolean functions has been proven too.

Keywords: *Boolean function, strong affinity level, spectral characteristics, algebraic degree, weight, algebraic immunity.*

Введение

Среди методов решения систем булевых уравнений, возникающих, в том числе, в задачах криптографии, важное место занимают методы, осуществляющие сведение исходной системы к системе линейных булевых уравнений. К ним можно отнести как методы, использующие операции в полиномиальных идеалах [1], так и методы, осуществляющие опробование значений части переменных: метод Жу — Витсе [2], метод локальных аффинностей [3] и т. д.

Очевидно, что эффективность этих методов зависит от свойств булевых функций, встретившихся в исходной системе. В данной работе рассматривается вопрос о свойствах функций, преобразуемых в аффинные путём произвольной фиксации значений некоторого набора переменных. Вводимое понятие уровня сильной аффинности сильнее, чем известное понятие уровня аффинности, ранее изучавшееся в работах В. Г. Рябова, О. А. Логачёва, М. Л. Бурякова и др. [4–7].

Введём необходимые определения и обозначения:

- $\mathbb{F}_2$ — поле из двух элементов;
- V_n — n-мерное векторное пространство над полем $\mathbb{F}_2$;
- $\mathcal{F}_n$ — класс булевых функций от n переменных;
- $||x||$ — вес Хэмминга двоичного вектора $x \in V_n$;
- $||f||$ — вес функции f. Булева функция $f \in \mathcal{F}_n$ называется *сбалансированной*, если $||f|| = 2^{n-1}$;
- $\deg(f)$ — алгебраическая степень булевой функциии f. Булеву функцию f будем называть *квадратичной*, если $\deg(f) \leqslant 2$, и *аффинной*, если $\deg(f) \leqslant 1$;
- $\langle u, x\rangle$ — скалярное произведение векторов из V_n;
- $\mathrm{ev}(f)$ — число существенных переменных функции f;
- $\mathrm{Al}(f)$ — порядок алгебраической иммунности булевой функции f;
- $\mathcal{A}_n$ — класс аффинных булевых функций от n переменных;
- $\mathcal{S}_n$ — класс симметрических булевых функций от n переменных;
- $\mathcal{M}_n$ — класс монотонных булевых функций от n переменных;
- $\mathcal{B}_n$ — класс бент-функций от n переменных;
- $\mathcal{M}_{2k}$ — класс функций Елисеева — Майорана — Макфарланда от $2k$ переменных, то есть функций вида $f(x, y) = \langle x, s(y)\rangle \oplus h(y)$, где s — подстановка на V_k; $h(y)$ — произвольная функция из V_k. Известно, что $\mathcal{M}_{2k} \subset \mathcal{B}_{2k}$.

Для произвольной функции $f \in \mathcal{F}_n$ и произвольного $u \in V_n$ преобразование Уолша — Адамара функции f определяется выражением

$$W_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus \langle u, x\rangle}.$$

Набор целых чисел $\{W_f(u) : u \in V_n\}$ называется *спектром* функции f, а каждое число $W_f(u)$ — *спектральным коэффициентом* Уолша — Адамара функции f.

Для наборов $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$, $\mathbf{b} = (b_1, \ldots, b_k) \in V_k$ при $k \leqslant n$ обозначим через $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}$ булеву функцию из $\mathcal{F}_{n-k}$, получаемую из f фиксацией переменных $x_{i_1} = b_1, \ldots, x_{i_k} = b_k$ и называемую подфункцией функции f.

Определение 1. Булева функция $f \in \mathcal{F}_n$ называется *k-аффинной*, $1 \leqslant k \leqslant n-1$, если существуют такие наборы $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$, $\mathbf{b} = (b_1, \ldots, b_k) \in V_k$, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}$ является аффинной, то есть $\deg(f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}) \leqslant 1$.

Определение 2. *Уровнем аффинности* $\mathrm{la}(f)$ булевой функции $f \in \mathcal{F}_n$ называется минимальное неотрицательное целое число k, для которого функция f является k-аффинной.

Определение 3. Булева функция $f \in \mathcal{F}_n$ называется *сильно k-аффинной*, $1 \leqslant k \leqslant n-1$, если существует такой набор $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$, что подфункция $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}$ является аффинной для любого $\mathbf{b} = (b_1, \ldots, b_k) \in V_k$, то есть $\deg(f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}) \leqslant 1$.

Определение 4. *Уровнем сильной аффинности* $\mathrm{la}_s(f)$ булевой функции $f \in \mathcal{F}_n$ будем называть минимальное неотрицательное целое число k, для которого функция f является сильно k-аффинной.

Замечание 1. Очевидно, что для любой булевой функции $f \in \mathcal{F}_n$ справедливо

$$\mathrm{la}(f) \leqslant \mathrm{la}_s(f) \leqslant \mathrm{ev}(f) - 1 \leqslant n - 1$$

и для любой квадратичной функции $f \in \mathcal{F}_n$

$$\mathrm{la}(f) = \mathrm{la}_s(f).$$

Обозначим через $\mathcal{SA}_n^k$ класс булевых функций от n переменных с уровнем сильной аффинности, равным k, а через $\mathcal{SA}_n^{\leqslant k}$ — класс булевых функций от n переменных с уровнем сильной аффинности, не превосходящим k, где $k \in \{0, \ldots, n-1\}$.

Замечание 2. Для любой аффинной функции как уровень аффинности, так и уровень сильной аффинности равны нулю.

Приведём несколько вспомогательных утверждений, которые пригодятся в дальнейшем.

Утверждение 1 [5]. Для булевой функции $f \in \mathcal{F}_n$ справедливо неравенство

$$\mathrm{la}(f) \geqslant \mathrm{Al}(f) - 1.$$

Утверждение 2 [5]. Для $f \in \mathcal{B}_{2k}$ справедливо соотношение $\mathrm{la}(f) \geqslant k$.

Для $f \in \mathcal{F}_n$ справедливо, что

$$||f|| - \text{нечётен} \Leftrightarrow \deg(f) = n. \tag{1}$$

Для $f \in \mathcal{B}_{2k}$, где $k \geqslant 2$, справедливо соотношение

$$\deg(f) \leqslant k. \tag{2}$$

При $\deg(f) \geqslant 1$ справедливо соотношение

$$2^{n-\deg(f)} \leqslant ||f|| \leqslant 2^n - 2^{n-\deg(f)}. \tag{3}$$

Определение 5. *Обыкновенным графом* называется упорядоченная пара объектов $G = (V, E)$, где V — множество вершин; $E \subseteq V^{[2]}$ — множество рёбер. Под $V^{[2]}$ подразумевается множество всех неупорядоченных пар различных элементов из V.

Если $|V| = n$ и $E = \varnothing$, то такой граф называется *пустым* и обозначается O_n.

Если $|V| = n$ и $E = V^{[2]}$, то такой граф называется *полным* и обозначается K_n.

1. Общие свойства уровня сильной аффинности

Приведём критерии для определения уровня сильной аффинности.

Обозначим через T множество мономов булевой функции $f(x_1, \ldots, x_n)$ при её представлении многочленом Жегалкина.

Теорема 1. Для булевой функции $f(x_1, \ldots, x_n)$ соотношение $\mathrm{la}_s(f) = n - 1$ выполнено тогда и только тогда, когда для любых i, j, $1 \leqslant i < j \leqslant n$, существует такой моном $x_{t_1} \ldots x_{t_{s_{i,j}}} \in T$, $s_{i,j} \in \{2, \ldots, n\}$, что $\{i, j\} \subseteq \{t_1, \ldots, t_{s_{i,j}}\}$.

Доказательство.

Необходимость. Так как $\mathrm{la}_s(f) = n-1$, то $\mathrm{la}_s(f) > n-2$. Значит, для любых $1 \leqslant i_1 < \ldots < i_{n-2} \leqslant n$ существует такой набор $(b_1, \ldots, b_{n-2}) \in V_{n-2}$, что $\deg(f_{i_1,\ldots,i_{n-2}}^{b_1,\ldots,b_{n-2}}) > 1$. Так как это функция от двух переменных, она квадратична и содержит моном $x_{i_{n-1}} x_{i_n}$. Это значит, что исходная функция f содержит некоторый моном $x_{t_1} \ldots x_{t_{s_{i,j}}} \in T$, такой,

что $\{i_{n-1}, i_n\} \subseteq \{t_1, \ldots, t_{s_{i,j}}\}$. Данные рассуждения справедливы для любого набора $1 \leqslant i_1 < \ldots < i_{n-2} \leqslant n$, поэтому любое произведение $x_i x_j$, $1 \leqslant i < j \leqslant n$, содержится в некотором мономе функции f.

Достаточность. Пусть $\mathrm{la}_s(f) \leqslant n-2$. Тогда существует такой набор $1 \leqslant i_1 < \ldots < i_m \leqslant n$, где $m \leqslant n-2$, что для любого вектора $(b_1, \ldots, b_m) \in V_m$ функция $f_{i_1,\ldots,i_m}^{b_1,\ldots,b_m}$ аффинна. Возьмём $i, j \in \{1, \ldots, n\} \setminus \{i_1, \ldots, i_m\}$. Так как ни одна из функций $f_{i_1,\ldots,i_m}^{b_1,\ldots,b_m}$ не содержит произведения $x_i x_j$, то функция f не содержит ни одного монома, содержащего произведение $x_i x_j$, — противоречие. ■

Теорему 1 можно интерпретировать в терминах теории графов для более удобного её понимания и применения на практике.

Для булевой функции $f(x_1, \ldots, x_n)$ построим граф $G_f = (V, E)$, где $V = \{1, \ldots, n\}$, а множество рёбер задается по следующему правилу:

$$\{i, j\} \in E \Leftrightarrow \text{ существует такой моном } x_{i_1} \ldots x_{i_s} \in T, \text{ что } \{i, j\} \subseteq \{i_1, \ldots, i_s\}.$$

Такое сопоставление графа функции является сюръективным, но не биективным, например, функциям $g_1(x_1, x_2, x_3) = x_1 x_2 x_3$ и $g_2(x_1, x_2, x_3) = x_1 x_2 \oplus x_1 x_3 \oplus x_2 x_3$ сопоставляется один и тот же полный граф K_3.

Тогда в терминах теории графов теорема 1 имеет следующую эквивалентную формулировку:

Следствие 1. Для булевой функции $f(x_1, \ldots, x_n)$ соотношение $\mathrm{la}_s(f) = n-1$ выполнено тогда и только тогда, когда граф G_f полный.

Приведём ещё одну эквивалентную формулировку теоремы 1.

Следствие 2. Для булевой функции $f(x_1, \ldots, x_n)$ соотношение $\mathrm{la}_s(f) = n-1$ выполнено тогда и только тогда, когда для любых $1 \leqslant i < j \leqslant n$ выполняется условие

$$f_{i,j}^{1,1} \oplus f_{i,j}^{1,0} \oplus f_{i,j}^{0,1} \oplus f_{i,j}^{0,0} \not\equiv 0.$$

Доказательство. Разложим функцию f по переменным x_i, x_j и преобразуем данное представление:

$$f = x_i x_j f_{i,j}^{1,1} \oplus (x_i \oplus 1) x_j f_{i,j}^{0,1} \oplus x_i (x_j \oplus 1) f_{i,j}^{1,0} \oplus (x_i \oplus 1)(x_j \oplus 1) f_{i,j}^{0,0} =$$
$$= x_i x_j (f_{i,j}^{1,1} \oplus f_{i,j}^{0,1} \oplus f_{i,j}^{1,0} \oplus f_{i,j}^{0,0}) \oplus x_i (f_{i,j}^{1,0} \oplus f_{i,j}^{0,0}) \oplus x_j (f_{i,j}^{0,1} \oplus f_{i,j}^{0,0}) \oplus f_{i,j}^{0,0}.$$

По теореме 1 любое произведение $x_i x_j$, $1 \leqslant i < j \leqslant n$, встречается в некотором мономе функции f при её представлении многочленом Жегалкина, а значит, функция $f_{i,j}^{1,1} \oplus f_{i,j}^{0,1} \oplus f_{i,j}^{1,0} \oplus f_{i,j}^{0,0}$ никогда не равна тождественному нулю. ■

Пример 1. Функция $f_1(x_1, x_2, x_3, x_4, x_5) = x_1 x_2 x_3 \oplus x_1 x_3 x_4 \oplus x_1 x_3 x_5 \oplus x_2 x_4 x_5$ имеет максимальный уровень сильной аффинности, равный 4, так как любое из 10 произведений вида $x_i x_j$, $1 \leqslant i < j \leqslant 5$, является частью некоторого монома многочлена Жегалкина функции $f_1(x_1, \ldots, x_5)$ и граф функции G_{f_1} полный.

Следствие 3. Если для булевой функции $f \in \mathcal{F}_n$ справедливо, что $\deg(f) = n$, то $\mathrm{la}_s(f) = n-1$.

Доказательство. Функция степени n содержит моном $x_1 \ldots x_n$ в её записи в виде многочлена Жегалкина; любое произведение $x_i x_j$, $1 \leqslant i < j \leqslant n$, содержится в этом мономе, значит, по теореме 1 уровень сильной аффинности данной функции максимален. ■

Лемма 1. Если для функции $f \in \mathcal{F}_n$ верно, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k} = c(b_1,\ldots,b_k)$ для любого $(b_1,\ldots,b_k) \in V_k$, где $c(b_1,\ldots,b_k) \in \mathbb{F}_2$, то любая переменная из множества $\{x_1,\ldots,x_n\} \setminus \{x_{i_1},\ldots,x_{i_k}\}$ фиктивна.

Доказательство. Без ограничения общности будем считать, что $i_1 = 1$, $\ldots$, $i_k = k$. Предположим, что для некоторого $t \in \{k+1,\ldots,n\}$ переменная x_t существенна. Тогда найдётся такой набор $(a_1,\ldots,a_{t-1},a_{t+1},\ldots,a_n)$, что $f(a_1,\ldots,a_{t-1},0,a_{t+1},\ldots,a_n) \neq f(a_1,\ldots,a_{t-1},1,a_{t+1},\ldots,a_n)$. Но по условию $f^{a_1,\ldots,a_k}_{i_1,\ldots,i_k} \equiv$ $\equiv$ const, а значит, $f(a_1,\ldots,a_{t-1},0,a_{t+1},\ldots,a_n) = f(a_1,\ldots,a_{t-1},1,a_{t+1},\ldots,a_n)$ — противоречие. ■

Докажем основной критерий точного значения уровня сильной аффинности.

Теорема 2. Для булевой функции $f(x_1,\ldots,x_n)$ соотношение $\mathrm{la}_s(f) = k$, где $k \leqslant n-2$, выполнено тогда и только тогда, когда одновременно выполняются следующие свойства:

1) для некоторого набора $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$ выполняется следующее условие: ни один моном многочлена Жегалкина функции f не содержит произведение вида $x_{i_j}x_{i_l}$, $1 \leqslant j < l \leqslant n-k$;
2) набора $1 \leqslant i_1 < \ldots < i_t \leqslant n$ длины $t > n-k$ с таким же свойством не существует.

Доказательство.

Достаточность докажем индукцией по параметру k.

Б а з а: $k = 0$. Тогда функция f аффинна и $\mathrm{la}_s(f) = 0$, так как для любых $1 \leqslant i < j \leqslant n$ моном x_ix_j не содержится ни в каком мономе функции f.

Пусть $k = 1$. Тогда существует такой набор $1 \leqslant i_1 < i_2 < \ldots < i_{n-1} \leqslant n$, что никакое произведение x_ix_j, $\{i,j\} \subseteq \{i_1,\ldots,i_{n-1}\}$, не входит в мономы функции f. Без ограничения общности положим $i_1 = 1$, $\ldots$, $i_{n-1} = n-1$. Функция f в этом случае, кроме своей аффинной части, может содержать мономы x_1x_n, x_2x_n, $\ldots$, $x_{n-1}x_n$. Очевидно, что при фиксации переменной x_n любым значением $b \in \mathbb{F}_2$ получим аффинную подфункцию, поэтому $\mathrm{la}_s(f) \leqslant 1$. Случай $\mathrm{la}_s(f) = 0$ невозможен ввиду условия 2, поэтому $\mathrm{la}_s(f) = 1$.

Ш а г: из справедливости утверждения при $k = d-1 \geqslant 1$ докажем его справедливость при $k = d$.

Имеем такой набор $1 \leqslant i_1 < i_2 < \ldots < i_{n-d} \leqslant n$, что ни один моном многочлена Жегалкина функции f не содержит произведения вида $x_{i_j}x_{i_l}$, $1 \leqslant j < l \leqslant n-d$. Без ограничения общности положим $i_1 = 1$, $\ldots$, $i_{n-d} = n-d$. Представим функцию f в виде

$$f(x_1,\ldots,x_n) = x_1x_2g_0(x_3,\ldots,x_n) \oplus x_1g_1(x_3,\ldots,x_n) \oplus x_2g_2(x_3,\ldots,x_n) \oplus g_3(x_3,\ldots,x_n).$$

Так как функция f не содержит мономов, включающих произведение x_1x_2 (в силу того, что $n-d \geqslant 2$), то $g_0 \equiv 0$. По этой же причине функции g_1 и g_2 не содержат переменных $x_3,\ldots,x_{n-d}$ в многочлене Жегалкина и функция $g_3(x_3,\ldots,x_n)$ содержит переменные $x_3,\ldots,x_{n-d}$ только в своей аффинной части, то есть

$$\begin{gathered} g_1(x_3,\ldots,x_n) \equiv h_1(x_{n-d+1},\ldots,x_n), \\ g_2(x_3,\ldots,x_n) \equiv h_2(x_{n-d+1},\ldots,x_n), \\ g_3(x_3,\ldots,x_n) = h_3(x_{n-d+1},\ldots,x_n) \oplus a_3x_3 \oplus \ldots \oplus a_{n-d}x_{n-d}, \end{gathered}$$

где $a_3,\ldots,a_{n-d} \in \mathbb{F}_2$. Таким образом,

$$\begin{gathered} f(x_1,\ldots,x_n) \equiv x_1h_1(x_{n-d+1},\ldots,x_n) \oplus x_2h_2(x_{n-d+1},\ldots,x_n) \oplus \\ \oplus h_3(x_{n-d+1},\ldots,x_n) \oplus a_3x_3 \oplus \ldots \oplus a_{n-d}x_{n-d}. \end{gathered} \tag{4}$$

Теперь очевидно, что, зафиксировав переменные $x_{n-d+1}, \ldots, x_n$ любыми значениями, получим аффинную функцию, то есть $\mathrm{la}_s(f) \leqslant d$.

Предположим, что $\mathrm{la}_s(f) = p < d$. Тогда существует такой набор переменных $\{y_1, \ldots, y_p\} \subsetneq \{x_{n-d+1}, \ldots, x_n\}$, что $f^{b_1,\ldots,b_p}_{y_1,\ldots,y_p}$ аффинна при любых $b_1, \ldots, b_p \in \mathbb{F}_2$. Это означает, что $h_1{}^{b_1,\ldots,b_p}_{y_1,\ldots,y_p}$ и $h_2{}^{b_1,\ldots,b_p}_{y_1,\ldots,y_p}$ — константы при любых $b_1, \ldots, b_p \in \mathbb{F}_2$.

Тогда по лемме 1 все переменные из множества $\{x_{n-d+1}, \ldots, x_n\} \setminus \{y_1, \ldots, y_p\}$ являются фиктивными для функций h_1 и h_2. Рассмотрим произвольный $x_l \in \{x_{n-d+1}, \ldots, x_n\} \setminus \{y_1, \ldots, y_p\} \neq \varnothing$ и набор индексов $1 < 2 < \ldots < n-d < l$ длины $n-d+1$. У функции f в многочлене Жегалкина нет мономов, содержащих произведения x_1x_l, x_2x_l, …, $x_{n-d}x_l$, а значит, для набора $1 < 2 < \ldots < n-d < l$ длины $n-d+1$ верно, что для любых $\{i, j\} \subseteq \{1, \ldots, n-d, l\}$ не существует монома многочлена Жегалкина функции f, содержащего произведение вида x_ix_j. Таким образом, $\mathrm{la}_s(f) = d$.

Необходимость. Докажем первое свойство от противного. Пусть не существует такого набора длины $n-k$, что для любых i, j из этого набора не существует монома многочлена Жегалкина функции f, содержащего произведение вида x_ix_j, то есть данное свойство может выполняться лишь для какого-то набора меньшей длины. Тогда по доказательству достаточности этой теоремы следует, что $\mathrm{la}_s(f) < k$, — противоречие.

Докажем второе свойство также от противного. Пусть существует набор длины $t > n-k$, для которого выполняются вышеуказанные свойства. Тогда по доказательству достаточности этой теоремы следует, что $\mathrm{la}_s(f) > k$, — противоречие.

Таким образом, оба свойства верны. ■

Теорему 2 можно сформулировать в терминах теории графов следующим образом:

Следствие 4. Для булевой функции $f(x_1, \ldots, x_n)$ соотношение $\mathrm{la}_s(f) = k$, где $k \leqslant n-2$, выполнено тогда и только тогда, когда граф G_f содержит подграф O_{n-k} и не содержит подграфа O_t, где $t > n-k$.

Замечание 3. Таким образом, задача определения уровня сильной аффинности сводится к задаче поиска максимально пустого подграфа, которая более известна как задача поиска максимально независимого множества. В общем случае данная задача относится к классу труднорешаемых, вместе с тем существуют приближённые алгоритмы её решения.

Следствие 5. Для булевой функции $f \in \mathcal{F}_n$ соотношение $\mathrm{la}_s(f) = k$, где $k \leqslant n-2$, выполнено тогда и только тогда, когда одновременно выполняются следующие свойства:

1) существует такой набор $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$, что для любых $\{i, j\} \subseteq \{i_1, \ldots, i_{n-k}\}$ выполняется
$$f^{1,1}_{i,j} \oplus f^{1,0}_{i,j} \oplus f^{0,1}_{i,j} \oplus f^{0,0}_{i,j} \equiv 0;$$
2) набора $1 \leqslant i_1 < \ldots < i_t \leqslant n$ длины $t > n-k$ с таким же свойством не существует.

Доказательство. Достаточно воспользоваться разложением функции f из доказательства следствия 2 и заметить его связь с разложением (4). ■

Следует также отметить, что следствия 2 и 5 являются частными случаями теоремы 7 из работы [8].

Пример 2. Функция $f_2(x_1, \ldots, x_8) = x_2x_3x_4x_7x_8 \oplus x_2x_5x_6x_7x_8 \oplus x_3x_4x_5x_7x_8 \oplus \oplus x_1x_2x_7x_8 \oplus x_1x_3x_4 \oplus x_2x_6x_7 \oplus x_4x_5x_7 \oplus x_1x_7 \oplus x_2x_6 \oplus x_5x_8 \oplus x_6x_7$ имеет уровень сильной аффинности $\mathrm{la}_s(f_2) = 5$, так как граф G_{f_2} содержит подграф O_3.

Замечание 4. Из доказательства теоремы 2 следует, что условие 1 из её формулировки используется для оценки уровня сильной аффинности сверху, а условие 2 — для его оценки снизу, и пересечение этих условий даёт точное значение уровня аффинности. Поэтому укажем важное следствие:

Следствие 6.

1) Если для функции $f(x_1, \ldots, x_n)$ и фиксированного k, $k \leqslant n-2$, существует такой набор $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$, что ни один моном многочлена Жегалкина функции f не содержит произведения вида $x_{i_j}x_{i_l}$, $1 \leqslant j < l \leqslant n-k$, то $\mathrm{la}_s(f) \leqslant k$.
2) Если для функции $f(x_1, \ldots, x_n)$ и фиксированного k, $k \leqslant n-2$, не существует такого набора $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$, что ни один моном многочлена Жегалкина функции f не содержит произведения вида $x_{i_j}x_{i_l}$, $1 \leqslant j < l \leqslant n-k$, то $\mathrm{la}_s(f) > k$.

Данное следствие также можно сформулировать в терминах теории графов:

Следствие 7.

1) Если граф G_f функции $f(x_1, \ldots, x_n)$ содержит подграф O_{n-k}, где $k \leqslant n-2$, то $\mathrm{la}_s(f) \leqslant k$.
2) Если граф G_f функции $f(x_1, \ldots, x_n)$ не содержит подграфа O_{n-k}, где $k \leqslant n-2$, то $\mathrm{la}_s(f) > k$.

2. Связь уровня сильной аффинности с другими характеристиками булевых функций

В работе [5] исследована связь уровня аффинности булевой функции с рядом её криптографических параметров.

Теорема 3 [5]. Для коэффициентов Уолша — Адамара функции $f \in \mathcal{F}_n$ выполняется неравенство

$$\max_{u \in V_n} \left|W_f(u)\right| \geqslant 2^{n-\mathrm{la}(f)}.$$

Рассмотрим связь между уровнем сильной аффинности булевой функции и её спектральными характеристиками.

Пусть для булевой функции $f \in \mathcal{F}_n$ справедливо равенство $\mathrm{la}_s(f) = k$ и для набора $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$ подфункция $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}$ аффинна для любого $\mathbf{b} = (b_1, \ldots, b_k) \in V_k$. Положим

$$R = \left|\left\{v \in V_{n-k} : f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k} = \langle v, x\rangle \oplus w, (b_1, \ldots, b_k) \in V_k\right\}\right|,$$

где $w \in \mathbb{F}_2$. Очевидно, что $1 \leqslant R \leqslant \min\{2^k, 2^{n-k}\} \leqslant 2^{n/2}$. Известен следующий факт:

Лемма 2 [9]. Для любой функции $f \in \mathcal{F}_n$, произвольного подпространства $L \subseteq V_n$ и произвольных векторов $a, c \in V_n$ справедливо равенство

$$2^{\dim L - n}(-1)^{\langle a,c\rangle} \sum_{x \in L^\perp \oplus c} W_f(x)(-1)^{\langle a,x\rangle} = \sum_{x \in L \oplus a} (-1)^{f(x)\oplus\langle c,x\rangle}. \tag{5}$$

Теорема 4. Существует по крайней мере R различных векторов $u \in V_n$, для каждого из которых выполняется неравенство

$$\left|W_f(u)\right| \geqslant 2^{n-\mathrm{la}_s(f)}.$$

Доказательство. Пусть $\mathrm{la}_s(f) = k$. Тогда найдётся такой набор $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}(x) = \langle v, x\rangle \oplus w$ для любого $(b_1,\ldots,b_k) \in V_k$, причём $v = v(b_1,\ldots,b_k) \in V_{n-k}$, $w = w(b_1,\ldots,b_k) \in \mathbb{F}_2$.

Аналогично доказательству теоремы 3, для фиксированного $(b_1,\ldots,b_k) \in V_k$ положим:

— $L = \{u \in V_n : u_{i_1} = \ldots = u_{i_k} = 0\}$ — подпространство размерности $\dim L = n - k$ пространства V_n;
— вектор a, такой, что $a_{i_j} = b_j$ для $j \in \{1,\ldots,k\}$ и $a_{i_j} = 0$ для $j \notin \{1,\ldots,k\}$;
— вектор c, такой, что $c_{i_j} = v_{i_j}$ для $j \notin \{1,\ldots,k\}$ и $c_{i_j} = 0$ для $j \in \{1,\ldots,k\}$.

Правая часть равенства (5) в этом случае есть

$$\sum_{x\in L\oplus a} (-1)^{f(x)\oplus\langle c,x\rangle} = \sum_{x\in V_{n-k}} (-1)^{f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}(x)\oplus\langle v,x\rangle} =$$
$$= \sum_{x\in V_{n-k}} (-1)^{\langle v,x\rangle\oplus w\oplus\langle v,x\rangle} = \sum_{x\in V_{n-k}} (-1)^w = 2^{n-k}(-1)^w.$$

Таким образом, имеем $(-1)^{\langle a,c\rangle} \sum\limits_{x\in L^\perp\oplus c} W_f(x)(-1)^{\langle a,x\rangle} = 2^n(-1)^w$.

Переходя к абсолютным величинам, получим

$$\Big|\sum_{x\in L^\perp\oplus c} W_f(x)(-1)^{\langle a,x\rangle}\Big| = 2^n.$$

Так как число слагаемых в последней сумме равно 2^k, в плоскости $L^\perp \oplus c$ есть такой вектор $u \in V_n$, для которого $|W_f(u)| \geqslant 2^{n-k}$. Так как подпространство $L^\perp$ фиксировано, а вектор $c \in V_n$ пробегает различные R значений, множество $\{L^\perp \oplus c\}$ пробегает R различных непересекающихся плоскостей и в каждой из них найдётся такой вектор $u \in V_n$, что $\big|W_f(u)\big| \geqslant 2^{n-k}$. ■

Утверждение 3 [5]. Для любых $n \geqslant 3$, $2 \leqslant d \leqslant n$, $1 \leqslant k \leqslant n-2$ существует такая функция $f \in \mathcal{F}_n$, что $\deg(f) = d$, $\mathrm{la}(f) = k$.

Следующее утверждение устанавливает связь между уровнем сильной аффинности булевой функции и её алгебраической степенью, что отличает исследуемую характеристику от уровня аффинности (см. утверждение 3).

Утверждение 4. Для булевой функции $f \in \mathcal{F}_n$ справедливо соотношение

$$\mathrm{la}_s(f) \geqslant \deg(f) - 1.$$

Если $f \in \mathcal{B}_n$ и $n \geqslant 4$, то

$$\mathrm{la}_s(f) > \deg(f) - 1.$$

Доказательство. Пусть $\deg(f) = t$. Тогда функция f содержит моном $x_{s_1}\ldots x_{s_t}$, где $1 \leqslant s_1 < \ldots < s_t \leqslant n$, в её представлении в виде многочлена Жегалкина. По теореме 2 ни одна из пар этих индексов не может находиться в наборе $1 \leqslant i_1 < i_2 < \ldots < i_{n-\mathrm{la}_s(f)} \leqslant n$, поэтому максимальная длина этого набора не превышает $n - t + 1$. Таким образом, $n - \mathrm{la}_s(f) \leqslant n - t + 1$, откуда $\mathrm{la}_s(f) \geqslant t - 1 = \deg(f) - 1$.

Второе соотношение следует из (2), утверждения 2 и замечания 1. ■

Следствие 8. Если вес булевой функции $f \in \mathcal{F}_n$ нечётен, то $\mathrm{la}_s(f) = n - 1$.

Доказательство. Следует из (1) и утверждения 4. ■

Отсюда, ввиду утверждения 3, можно вывести связь между уровнем сильной аффинности и весом булевой функции.

Утверждение 5. Для булевой функции $f \in \mathcal{F}_n$, $\deg(f) \geqslant 1$, справедлива оценка

$$2^{n-\mathrm{la}_s(f)-1} \leqslant ||f|| \leqslant 2^n - 2^{n-\mathrm{la}_s(f)-1}, \tag{6}$$

при этом:

1) нижняя оценка достигается только на функциях f вида

$$f(x) = (x_{i_1} \oplus c_1)(x_{i_2} \oplus c_2) \dots (x_{i_k} \oplus c_k) l(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n});$$

2) верхняя оценка достигается только на функциях f вида

$$f(x) = (x_{i_1} \oplus c_1)(x_{i_2} \oplus c_2) \dots (x_{i_k} \oplus c_k) l(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n}) \oplus 1,$$

где $k = \mathrm{la}_s(f)$, $c_1, \dots, c_k \in \mathbb{F}_2$, $\deg(l) = 1$.

Доказательство. Справедливость оценки следует из (3) и Утверждения 4.

Докажем критерий достижимости нижней оценки в (6).

Достаточность. Для функций такого вида справедливо

$$||f|| = ||x_{i_{k+1}} \oplus x_{i_{k+2}} \oplus \dots \oplus x_{i_n}|| = 2^{n-k-1} = 2^{n-\mathrm{la}_s(f)-1}.$$

Необходимость. По условию $\mathrm{la}_s(f) = k$, значит, существует такой набор $1 \leqslant i_1 < \dots < i_k \leqslant n$, что $f^{b_1,\dots,b_k}_{i_1,\dots,i_k} \in \mathcal{A}_n$ для всех $(b_1, \dots, b_k) \in V_k$. Отсюда

$$||f^{b_1,\dots,b_k}_{i_1,\dots,i_k}|| \in \{0, 2^{n-k-1}, 2^{n-k}\}$$

для всех $(b_1, \dots, b_k) \in V_k$. Очевидно, что

$$||f|| = \sum_{(b_1,\dots,b_k) \in V_k} ||f^{b_1,\dots,b_k}_{i_1,\dots,i_k}|| = 2^{n-k-1}.$$

Отсюда следует, что существует единственный набор $(a_1, \dots, a_k) \in V_k$, такой, что $||f^{a_1,\dots,a_k}_{i_1,\dots,i_k}|| = 2^{n-k-1}$, и $||f^{b_1,\dots,b_k}_{i_1,\dots,i_k}|| = 0$ для всех $(b_1, \dots, b_k) \in V_k \setminus \{(a_1, \dots, a_k)\}$, то есть $f^{b_1,\dots,b_k}_{i_1,\dots,i_k} \equiv 0$ для всех $(b_1, \dots, b_k) \in V_k \setminus \{(a_1, \dots, a_k)\}$ и $\deg(f^{a_1,\dots,a_k}_{i_1,\dots,i_k}) = 1$.

Раскладывая функцию f по переменным $x_{i_1}, \dots, x_{i_k}$, получаем

$$f(x_1, \dots, x_n) = \bigoplus_{(b_1,\dots,b_k) \in V_k} (x_{i_1} \oplus b_1 \oplus 1) \dots (x_{i_k} \oplus b_k \oplus 1) f^{b_1,\dots,b_k}_{i_1,\dots,i_k}(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n}) =$$
$$= (x_{i_1} \oplus a_1 \oplus 1) \dots (x_{i_k} \oplus a_k \oplus 1) f^{a_1,\dots,a_k}_{i_1,\dots,i_k}(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n}).$$

Для доказательства необходимости остаётся положить $c_1 = a_1 \oplus 1$, …, $c_k = a_k \oplus 1$ и $l(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n}) = f^{a_1,\dots,a_k}_{i_1,\dots,i_k}(x_{i_{k+1}}, x_{i_{k+2}}, \dots, x_{i_n})$.

Доказательство верхней оценки аналогично; в этом случае $f^{b_1,\dots,b_k}_{i_1,\dots,i_k} \equiv 1$ для всех $(b_1, \dots, b_k) \in V_k \setminus \{(a_1, \dots, a_k)\}$ и $\deg(f^{a_1,\dots,a_k}_{i_1,\dots,i_k}) = 1$. ■

Из утверждения 1 и замечания 1 следует, что для булевой функции $f \in \mathcal{F}_n$ справедливо соотношение

$$\mathrm{la}_s(f) \geqslant \mathrm{Al}(f) - 1.$$

Данную оценку можно усилить для определённых классов булевых функций.

Утверждение 6 [5]. Для любой булевой функции $f \in \mathcal{F}_n$ справедливо

$$\mathrm{Al}(f) \leqslant \min_{\substack{0 \leqslant s \leqslant \lceil n/2 \rceil \\ 1 \leqslant i_1 < i_2 < \dots < i_s \leqslant n \\ (b_1,\dots,b_s) \in V_s}} \left\{ \deg\left(f^{b_1,\dots,b_s}_{i_1,\dots,i_s}\right) + s \right\}.$$

Утверждение 7. Если булева функция $f \in \mathcal{F}_n$ не является сбалансированной, то

$$\mathrm{la}_s(f) \geqslant \mathrm{Al}(f).$$

Доказательство. Пусть $\mathrm{la}_s(f) = k$. Тогда существует такой набор $1 \leqslant i_1 < \ldots < i_k \leqslant n$, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k} \in \mathcal{A}_n$ для всех $(b_1,\ldots,b_k) \in V_k$. Отсюда

$$||f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}|| \in \{0, 2^{n-k-1}, 2^{n-k}\}$$

для всех $(b_1,\ldots,b_k) \in V_k$. Если $||f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}|| = 2^{n-k-1}$ для всех $(b_1,\ldots,b_k) \in V_k$, то

$$||f|| = \sum_{(b_1,\ldots,b_k)\in V_k} ||f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k}|| = 2^k \cdot 2^{n-k-1} = 2^{n-1}$$

— противоречие с несбалансированностью функции f. Поэтому существует такой набор $(a_1,\ldots,a_k) \in V_k$, что $||f^{a_1,\ldots,a_k}_{i_1,\ldots,i_k}|| \in \{0, 2^{n-k}\}$, то есть $f^{a_1,\ldots,a_k}_{i_1,\ldots,i_k} \equiv \mathrm{const}$, а значит, ввиду утверждения 6, получаем

$$\mathrm{Al}(f) \leqslant \deg\left(f^{a_1,\ldots,a_k}_{i_1,\ldots,i_k}\right) + k = k.$$

Утверждение 7 доказано. ■

Представим булеву функцию $f \in \mathcal{F}_n$ в виде

$$f(x_1,\ldots,x_n) = g(x_1,\ldots,x_n) \oplus l(x_1,\ldots,x_n) \oplus a_0,$$

где $a_0 = f(0,\ldots,0)$, функция g содержит только мономы степени 2 и выше, а функция l — только мономы степени 1.

Утверждение 8. Если для булевой функции $f \in \mathcal{F}_n$ справедливо, что $l \equiv 0$, то

$$\mathrm{la}_s(f) \geqslant \mathrm{Al}(f).$$

Доказательство. Пусть $\mathrm{la}_s(f) = k$. Тогда существует такой набор $1 \leqslant i_1 < \ldots < i_k \leqslant n$, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k} \in \mathcal{A}_n$ для всех $(b_1,\ldots,b_k) \in V_k$. Из теоремы 2 следует, что в любом мономе функции f найдётся переменная из множества $\{x_{i_1},\ldots,x_{i_k}\}$. Тогда, зафиксировав каждую переменную из этого множества нулём, получим $f^{0,\ldots,0}_{i_1,\ldots,i_k} = a_0$, а значит, ввиду утверждения 6, $\mathrm{Al}(f) \leqslant \deg\left(f^{0,\ldots,0}_{i_1,\ldots,i_k}\right) + k = k$. ■

3. Уровень сильной аффинности булевых функций из некоторых классов

Обсудим задачу нахождения уровня сильной аффинности для некоторых классов функций. Рассмотрим класс симметрических булевых функций. В работе [10] получена оценка уровня аффинности для функций из этого класса:

Утверждение 9 [10]. Для функции $f \in \mathcal{S}_n$, такой, что $\deg(f) > 1$, справедлива оценка

$$\mathrm{la}(f) > n - \deg(f).$$

Найдём значение уровня сильной аффинности для функций этого класса.

Утверждение 10. Для функции $f \in \mathcal{S}_n$, такой, что $\deg(f) > 1$, справедлива следующая оценка:

$$\mathrm{la}_s(f) = n - 1.$$

Доказательство. Так как $\deg(f) > 1$, то функция f содержит моном степени выше 1, а значит, существует произведение $x_j x_k$, которое является частью некоторого монома функции f при её представлении многочленом Жегалкина. Так как по определению симметрической функции

$$f(x_1, \ldots, x_j, \ldots, x_k, \ldots, x_n) = f(x_{i_1}, \ldots, x_{i_j}, \ldots, x_{i_k}, \ldots, x_{i_n})$$

для любой перестановки $(i_1, \ldots, i_n)$ элементов множества $\{1, \ldots, n\}$, то произведение $x_{i_j} x_{i_k}$ также является частью некоторого монома функции f. Данные рассуждения справедливы для любой перестановки элементов множества $\{1, \ldots, n\}$, значит, для любых $1 \leqslant i < j \leqslant n$ существуют такие $s_{i,j} \in \{2, \ldots, n\}$ и моном $x_{t_1} \ldots x_{t_{s_{i,j}}} \in T$, что $\{i, j\} \subseteq \{t_1, \ldots, t_{s_{i,j}}\}$. Таким образом, из теоремы 1 следует справедливость утверждения 10. ■

Из утверждения 2 и замечания 1 следует, что для булевой функции $f \in \mathcal{B}_{2k}$ справедливо соотношение

$$\mathrm{la}_s(f) \geqslant k. \tag{7}$$

Следствие 9. Для булевой функции $f \in \mathcal{M}_{2k}$ справедливо равенство $\mathrm{la}_s(f) = k$.

Доказательство. При фиксации всех переменных y любыми значениями функция f становится аффинной, т. е. $\mathrm{la}_s(f) \leqslant k$. Ввиду (7) получаем искомое равенство. ■

Рассмотрим класс монотонных булевых функций $\mathcal{M}_n$.

Утверждение 11. Если функция $f \in \mathcal{M}_n$ существенно зависит от всех своих переменных, то

$$\mathrm{la}_s(f) = n - 1.$$

Доказательство. Предположим противное: $\mathrm{la}_s(f) = k \leqslant n-2$. Тогда существует такой набор $1 \leqslant i_1 < \ldots < i_k \leqslant n$, что $f^{b_1,\ldots,b_k}_{i_1,\ldots,i_k} \in \mathcal{A}_{n-k}$. Без ограничения общности положим $i_1 = 1, \ldots, i_k = k$. Тогда для некоторого фиксированного набора $(b_1, \ldots, b_k) \in V_k$ имеем

$$f^{b_1,\ldots,b_k}_{1,\ldots,k}(x_{k+1}, \ldots, x_n) = \langle v, x \rangle \oplus w = v_{k+1} x_{k+1} \oplus \ldots \oplus v_n x_n \oplus w,$$

где $v_{k+1}, \ldots, v_n, w \in \mathbb{F}_2$. Ввиду леммы 1 $v \neq 0$, иначе у функции f были бы фиктивные переменные. Без ограничения общности положим $v_{k+1} \neq 0$. Тогда ввиду монотонности функции f справедливо

$$\begin{gathered} v_{k+2} c_{k+2} \oplus \ldots \oplus v_n c_n \oplus w = f(b_1, \ldots, b_k, 0, c_{k+2}, \ldots, c_n) \leqslant \\ \leqslant f(b_1, \ldots, b_k, 1, c_{k+2}, \ldots, c_n) = 1 \oplus v_{k+2} c_{k+2} \oplus \ldots \oplus v_n c_n \oplus w \end{gathered} \tag{8}$$

для любых $c_{k+2}, \ldots, c_n \in \mathbb{F}_2$. Данное неравенство справедливо лишь в случае, когда $v_{k+2} = \ldots = v_n = w = 0$. Значит, $f^{b_1,\ldots,b_k}_{1,\ldots,k}(x_{k+1}, \ldots, x_n) = v_{k+1} x_{k+1}$, и тогда $f^{b_1,\ldots,b_k,b_{k+1}}_{1,\ldots,k,k+1} = \mathrm{const}$ для любого фиксированного набора $(b_1, \ldots, b_k, b_{k+1}) \in V_{k+1}$. По лемме 1 переменные $x_{k+2}, \ldots, x_n$ фиктивны — противоречие с условием. ■

Следствие 10. Если функция $f \in \mathcal{F}_n$ антимонотонна и существенно зависит от всех своих переменных, то

$$\mathrm{la}_s(f) = n - 1.$$

Доказательство. Аналогично доказательству утверждения 11, за исключением того, что вместо (8) имеет место неравенство

$$\begin{gathered} v_{k+2} c_{k+2} \oplus \ldots \oplus v_n c_n \oplus w = f(b_1, \ldots, b_k, 0, c_{k+2}, \ldots, c_n) \geqslant \\ \geqslant f(b_1, \ldots, b_k, 1, c_{k+2}, \ldots, c_n) = 1 \oplus v_{k+2} c_{k+2} \oplus \ldots \oplus v_n c_n \oplus w \end{gathered}$$

для всех $c_{k+2}, \ldots, c_n \in \mathbb{F}_2$; оно справедливо лишь в случае, когда $w = 1$, $v_{k+2} = \ldots = v_n = 0$. Далее доказательство аналогично. ■

Обобщением утверждения 11 является

Следствие 11. Для функции $f \in \mathcal{M}_n$ справедливо равенство

$$\mathrm{la}_s(f) = \mathrm{ev}(f) - 1.$$

Доказательство. Случай $\mathrm{ev}(f) = n$ доказан в утверждении 11.

Пусть функция f содержит $d < n$ существенных переменных. Без ограничения общности считаем переменные $x_{d+1}, \ldots, x_n$ фиктивными, т. е. в многочлене Жегалкина функции f присутствуют только мономы, содержащие переменные $x_1, \ldots, x_d$. Тогда $\mathrm{la}_s(f) \leqslant d - 1$. Если $\mathrm{la}_s(f) < d - 1$, то аналогично доказательству утверждения 11 получим, что функция f содержит более чем $n - d$ фиктивных переменных, т. е. существенных переменных меньше чем d, — противоречие. Значит, $\mathrm{la}_s(f) = d - 1$. ■

Для антимонотонных функций справедливо аналогичное утверждение, доказательство которого аналогично.

Следствие 12. Для антимонотонной функции $f \in \mathcal{F}_n$ справедливо равенство

$$\mathrm{la}_s(f) = \mathrm{ev}(f) - 1.$$

4. Асимптотические оценки уровня сильной аффинности

Будем говорить, что некоторое свойство асимптотически выполняется для почти всех булевых функций, если доля функций в $\mathcal{F}_n$, для которых это свойство выполняется, при $n \to \infty$, стремится к единице.

Асимптотическое поведение уровня аффинности описано В. Г. Рябовым в начале 1980-х и в более общем виде опубликовано в [11]. Для квадратичных форм асимптотическое поведение уровня аффинности изучено в [12].

В работах [6, 7, 13] также исследовано асимптотическое поведение уровня аффинности, в частности, доказана следующая

Теорема 5 [13]. Асимптотически при $n \to \infty$ для почти всех булевых функций $f \in \mathcal{F}_n$ справедливо

$$n - \lfloor \log_2 n \rfloor \leqslant \mathrm{la}(f) \leqslant n - \lceil \log_2 n \rceil + 1.$$

Изучим асимптотическое поведение уровня сильной аффинности. Для этого сначала оценим мощность множества $\mathcal{SA}_n^{\leqslant k}$ для произвольного $k \in \{0, \ldots, n-2\}$.

Утверждение 12. Для произвольного $k \in \{0, \ldots, n-2\}$ справедливо следующее неравенство:

$$2^{(n-k+1)2^k} \leqslant |\mathcal{SA}_n^{\leqslant k}| \leqslant \binom{n}{k} 2^{(n-k+1)2^k}.$$

Доказательство. По определению для $f \in \mathcal{SA}_n^{\leqslant k}$ выполнено $\mathrm{la}_s(f) \leqslant k$. Тогда, воспользовавшись логическим отрицанием п. 2 следствия 6, получим, что существует такой набор $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$, что ни один моном многочлена Жегалкина функции f не содержит произведения вида $x_{i_j} x_{i_l}$, $1 \leqslant j < l \leqslant n - k$.

Ввиду утверждения 4, $\deg(f) \leqslant k + 1$. Ясно, что никакой моном, содержащий две и более переменных из множества $\{x_{i_1}, \ldots, x_{i_{n-k}}\}$, не содержится в многочлене Жегалкина функции $f \in \mathcal{SA}_n^{\leqslant k}$, т. е. среди мономов этого многочлена могут быть мономы, содержащие переменные только из множества $\{x_1, \ldots, x_n\} \setminus \{x_{i_1}, \ldots, x_{i_{n-k}}\} = \{y_1, \ldots, y_k\}$

либо одну переменную из $\{x_{i_1}, \ldots, x_{i_{n-k}}\}$, а остальные из $\{y_1, \ldots, y_k\}$. Такие мономы назовём допустимыми. Посчитав количество допустимых мономов и возведя двойку в степень этого числа, получим количество функций из класса $\mathcal{SA}_n^{\leqslant k}$ с фиксированным набором $1 \leqslant i_1 < i_2 < \ldots < i_{n-k} \leqslant n$, это и будет искомой оценкой снизу:

— число допустимых мономов степени 0: 1;
— число допустимых мономов степени 1: n;
— число допустимых мономов степени 2: $\binom{k}{2} + (n-k)\binom{k}{1}$;
— число допустимых мономов степени 3: $\binom{k}{3} + (n-k)\binom{k}{2}$;
$\ldots$
— число допустимых мономов степени $k-1$: $\binom{k}{k-1} + (n-k)\binom{k}{k-2}$;
— число допустимых мономов степени k: $\binom{k}{k} + (n-k)\binom{k}{k-1}$;
— число допустимых мономов степени $k+1$: $(n-k)\binom{k}{k}$.

Всего допустимых мономов:

$$1 + n + \binom{k}{2} + (n-k)\binom{k}{1} + \binom{k}{3} + (n-k)\binom{k}{2} + \ldots + \binom{k}{k-1} + (n-k)\binom{k}{k-2} +$$
$$+ \binom{k}{k} + (n-k)\binom{k}{k-1} + (n-k)\binom{k}{k} = 1 + n + \sum_{i=2}^{k}\binom{k}{i} + (n-k)\sum_{i=1}^{k}\binom{k}{i} =$$
$$= 1 + n + (2^k - 1 - k) + (n-k)(2^k - 1) = (n-k+1)2^k.$$

Таким образом, доказана оценка снизу. Набор длины $n-k$ можем выбрать $\binom{n}{k}$ способами, откуда следует искомая оценка сверху. ■

Замечание 5. Данная оценка сверху не является достижимой, так как различным наборам длины $n-k$ может соответствовать одна и та же функция. Например, для функции $g(x_1, \ldots, x_6) = x_1x_2x_5x_6 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_5x_6$ справедливо $\mathrm{la}_s(g) = 3$ и ей соответствуют два различных набора длины 3: $1 < 2 < 5$ и $1 < 2 < 6$.

Теорема 6. Асимптотически при $n \to \infty$ для почти всех булевых функций $f \in \mathcal{F}_n$ справедливо равенство

$$\mathrm{la}_s(f) = n - 1.$$

Доказательство. Оценим число функций с уровнем сильной аффинности, равным $n-1$.

Так как $|\mathcal{SA}_n^{\leqslant n-1}| = 2^{2^n}$ и по утверждению 12 $|\mathcal{SA}_n^{\leqslant n-2}| \leqslant \binom{n}{2} 2^{3 \cdot 2^{n-2}}$, то

$$|\mathcal{SA}_n^{n-1}| = |\mathcal{SA}_n^{\leqslant n-1}| - |\mathcal{SA}_n^{\leqslant n-2}| \geqslant 2^{2^n} - \binom{n}{2} 2^{3 \cdot 2^{n-2}}.$$

Оценим долю таких функций при $n \to \infty$:

$$\frac{|\mathcal{SA}_n^{n-1}|}{2^{2^n}} \geqslant 1 - \binom{n}{2} 2^{3 \cdot 2^{n-2}} / 2^{2^n} = 1 - \binom{n}{2} 2^{3 \cdot 2^{n-2} - 2^n} = 1 - \frac{n(n-1)}{2^{2^{n-2}+1}} \to 1.$$

Отсюда следует искомое утверждение. ■

ЛИТЕРАТУРА

1. *Ars G., Faugere J.-C., Imai H., et al.* Comparsion between XL and Grobner basis algorithms // LNCS. 2004. V. 3329. P. 148–172.
2. *Joux A. and Vitse V.* A crossbred algorithm for solving Boolean polynomial systems // LNCS. 2018. V. 10737. P. 3–21.
3. *Логачев О. А., Сукаев А. А., Федоров С. Н.* Об одном методе решения систем квадратичных булевых уравнений, использующем локальные аффинности булевых функций // Информ. и её примен. 2019. Т. 13. № 2. С. 37–46.
4. *Буряков М. Л., Логачев О. А.* Об уровне аффинности булевых функций // Дискретная математика. 2005. Т. 17. № 4. С. 98–107.
5. *Буряков М. Л.* О связи уровня аффинности с криптографическими параметрами булевых функций // Дискретная математика. 2008. Т. 20. № 2. С. 3–14.
6. *Буряков М. Л.* Асимптотические оценки уровня аффинности для почти всех булевых функций // Дискретная математика. 2008. Т. 20. № 3. С. 73–79.
7. *Логачев О. А.* Нижняя граница уровня аффинности для почти всех булевых функций // Дискретная математика. 2008. Т. 20. № 4. С. 85–88.
8. *Бабуева А. А., Логачев О. А., Ященко В. В.* О связи локальных аффинностей булевой функции с некоторыми видами ее вырожденности // Дискретная математика. 2022. Т. 34. № 2. С. 7–25.
9. *Логачев О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.
10. *Logachev O. A., Yashchenko V. V., and Denisenko M. P.* Local affinity of Boolean mappings // Boolean Functions in Cryptology and Information Security. V. 18. IOS Press, 2008. P. 148–172.
11. *Рябов В. Г.* О степени ограничений функций q-значной логики на линейные многообразия // Прикладная дискретная математика. 2019. № 45. С. 13–25.
12. *Черемушкин А. В.* Об оценке уровня аффинности квадратичных форм // Дискретная математика. 2017. Т. 29. № 1. С. 114–125.
13. *Логачев О. А.* О значениях уровня аффинности для почти всех булевых функций // Прикладная дискретная математика. 2010. № 3. С. 17–21.

REFERENCES

1. *Ars G., Faugere J.-C., Imai H., et al.* Comparsion between XL and Grobner basis algorithms. LNCS, 2004, vol. 3329, pp. 148–172.
2. *Joux A. and Vitse V.* A crossbred algorithm for solving Boolean polynomial systems. LNCS, 2018, vol. 10737, pp. 3–21.
3. *Logachev O. A., Sukaev A. A., and Fedorov S. N.* Ob odnom metode resheniya sistem kvadratichnykh bulevykh uravneniy, ispol'zuyushchem lokal'nye affinnosti bulevykh funktsiy [On local affinity based method of solving systems of quadratic Boolean equations]. Informatika i Ee Primeneniya, 2019, vol. 13, no. 2, pp. 37–46. (in Russian)
4. *Buryakov M. L. and Logachev O. A.* On the affinity level of Boolean functions. Discrete Math. Appl., 2005, vol. 15, no. 5, pp. 479–488.
5. *Buryakov M. L.* The relationship between the level of affinity and cryptographic parameters of Boolean functions. Discrete Math. Appl., 2008, vol. 18, no. 3, pp. 227–238.
6. *Buryakov M. L.* Asymptotic bounds for the affinity level for almost all Boolean functions. Discrete Math. Appl., 2008, vol. 18, no. 5, pp. 545–551.
7. *Logachev O. A.* A lower bound for the affinity level for almost all Boolean functions. Discrete Math. Appl., 2008, vol. 18, no. 5, pp. 553–556.

8. *Babyeva A. A., Logachev O. A., and Yashchenko V. V.* On the relationship between local affinities of a Boolean function and some types of its degeneracy. Discrete Math. Appl., 2023, vol. 33, no. 6, pp. 339–353.
9. *Logachev O. A., Sal'nikov A. A., and Yashchenko V. V.* Bulevy funktsii v teorii kodirovaniya i kriptologii [Boolean Functions in Coding Theory and Cryptology]. Moscow, MCCME Publ., 2004. (in Russian)
10. *Logachev O. A., Yashchenko V. V., and Denisenko M. P.* Local affinity of Boolean mappings. Boolean Functions in Cryptology and Information Security, vol. 18, IOS Press, 2008, pp. 148–172.
11. *Ryabov V. G.* O stepeni ogranicheniy funktsiy q-znachnoy logiki na lineynye mnogoobraziya [On the degree of restrictions of q-valued logic functions to linear manifolds]. Prikladnaya Diskretnaya Matematika, 2019, no. 45, pp. 13–25. (in Russian)
12. *Cheremushkin A. V.* Estimating the level of affinity of a quadratic form. Discrete Math. Appl., 2017, vol. 27, no. 6, pp. 339–347.
13. *Logachev O. A.* O znacheniyakh urovnya affinnosti dlya pochti vsekh bulevykh funktsiy [On values of affinity level for almost all Boolean functions]. Prikladnaya Diskretnaya Matematika, 2010, no. 3, pp. 17–21. (in Russian)

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7 DOI 10.17223/20710410/71/3

ВИЗАНТИЙСКОЕ СОГЛАШЕНИЕ И ШИРОКОВЕЩАТЕЛЬНАЯ ПЕРЕДАЧА

С. М. Рацеев

Ульяновский государственный университет, г. Ульяновск, Россия

E-mail: ratseevsm@mail.ru

Византийское соглашение и широковещательная передача являются двумя фундаментальными протоколами и важнейшими строительными блоками в безопасных многосторонних вычислениях, поэтому повышение их эффективности представляет интерес как для теоретической, так и для практической криптографии. В данной обзорной работе приводятся протоколы византийского соглашения, протоколы широковещательной передачи, а также протоколы расширения для протоколов византийского соглашения и широковещательной передачи.

Ключевые слова: *византийское соглашение, широковещательная передача, криптографический протокол.*

BYZANTINE AGREEMENT AND BYZANTINE BROADCAST

S. M. Ratseev

Ulyanovsk State University, Ulyanovsk, Russia

Byzantine agreement and byzantine broadcast are the two most fundamental problems and essential building blocks in secure multiparty computations, and improving their efficiency is of interest to both theorists and practitioners. In this survey, we describe the most important constructions of Byzantine agreement and Byzantine broadcast, as well as their extension protocols.

Keywords: *Byzantine agreement, Byzantine broadcast, cryptographic protocol.*

Введение

Задача (византийской) широковещательной передачи заключается в том, что некоторый назначенный участник (отправитель) отправляет сообщение всем участникам, причём все (честные) получатели должны получить одинаковое сообщение, несмотря на то, что некоторые нечестные участники могут вести себя произвольным образом. Аналогично, византийское соглашение (Byzantine agreement) позволяет всем (честным) участникам, у каждого из которых имеется входное сообщение, определить одно и то же выходное сообщение.

Наиболее важна задача построения эффективных протоколов византийского соглашения и широковещательной передачи для входных сообщений большой длины, поскольку такие протоколы широко используются в качестве строительных блоков

безопасных многосторонних вычислений [1]. Простым решением для сообщения длины l является применение l протоколов трансляции (византийского соглашения) для каждого бита этого сообщения в отдельности. Такой подход имеет коммуникационную сложность $\Omega(ln^2)$ бит, где n — число участников. Более эффективным решением для сообщения большой длины l является применение протоколов расширения для протоколов византийского соглашения и широковещательной передачи. В этом случае можно достичь коммуникационной сложности $O(ln)$ бит.

В данной работе исследуются как протоколы византийского соглашения и широковещательной передачи, так и протоколы расширения для этих протоколов. Рассматриваются информационно-теоретически и криптографически безопасные протоколы. Криптографическая система (протокол) обладает свойством *информационно-теоретической безопасности*, если ни один противник не может взломать систему, независимо от того, насколько он силён, т. е. он может обладать неограниченными вычислительными возможностями. Если же криптографическая система удовлетворяет требованиям *криптографической* (*вычислительной*) *безопасности*, то это означает, что она безопасна только до тех пор, пока противник располагает ограниченными вычислительными ресурсами. Часто это тот случай, когда используются такие инструменты, как асимметричные шифры, с которыми связана вычислительная проблема, которую противник не должен быть в состоянии решить, чтобы гарантировать безопасность системы.

Оракулом в теории вычислений называется внешнее (по отношению к алгоритмам) устройство, которое в ответ на запрос произвольного алгоритма выдаёт значение некоторой функции на этом запросе. При этом как обращение к оракулу, так и получение от него ответа занимают один такт работы алгоритма.

1. Византийское соглашение

Задачу византийского соглашения можно определить следующим образом. Пусть имеется n участников $P_1, \ldots, P_n$, среди которых t могут быть нечестными и контролироваться противником. Участники соединены попарно защищёнными и аутентифицированными каналами. У каждого участника P_i имеется своё входное значение $x_i \in \{0,1\}^*$. Цель протокола состоит в том, чтобы все честные участники согласовали общее выходное значение y.

Определение 1 (византийское соглашение). Протокол, в котором изначально каждый участник P_i имеет входное значение (сообщение) $x_i \in \{0,1\}^*$ и который завершается при вычислении выходного значения y_i, является протоколом *византийского соглашения*, устойчивого к активному противнику, контролирующему до t участников (т. е. t-безопасное византийское соглашение), если выполняются следующие свойства:

1) *достоверность* (validity): если все честные участники имеют на входе x, то каждый честный участник P_i имеет выходное значение $y_i = x$;
2) *договоренность* (agreement): любые два честных участника P_i и P_j определяют одинаковое выходное значение $y_i = y_j$, т. е. выходные значения всех честных участников одинаковы.

Заметим, что t-безопасное византийское соглашение имеет смысл только при $t < n/2$.

Рассмотрим протокол византийского соглашения BGP (Berman, Garay, Perry) [2] для однобитовых входных данных с полиномиальной коммуникационной сложностью. Под *коммуникационной сложностью* передачи данных протокола понимается общее количество битов, отправленных/полученных честными участниками во время вы-

полнения протокола (при этом учитываются только те биты, которые должны быть получены в соответствии со спецификацией протокола). Под *раундовой сложностью* понимается количество раундов, необходимых протоколу для завершения работы.

Протокол 1 (протокол BGP).
Пусть каждый участник P_i получает на вход бит $x_i \in \{0,1\}$.
Цикл: $k = 1, \ldots, t+1$:

— Р а у н д 1. Каждый участник P_i передает свой бит x_i всем остальным участникам. В конце раунда участник P_i для $b = 0, 1$ определяет

$$C_i^b = \begin{cases} 1, & \text{если } P_i \text{ получил } b \text{ от не менее чем } n-t \text{ участников,} \\ 0 & \text{иначе.} \end{cases}$$

Заметим, что при вычислении C_i^b учитывается и собственное входное значение x_i участника P_i.

— Р а у н д 2. Каждый участник P_i передаёт C_i^0 и C_i^1 всем остальным участникам. Пусть C_{ji}^b — значение, полученное участником P_i от P_j. В конце раунда участник P_i определяет

$$D_i^b = |\{j : C_{ji}^b = 1\}|, \quad b = 0, 1.$$

Участник P_i определяет своё (промежуточное) выходное значение y_i:

$$y_i = \begin{cases} 1, & D_i^1 > t, \\ 0, & D_i^1 \leqslant t. \end{cases}$$

— Р а у н д 3. Участник P_k (k — номер итерации цикла) передаёт значение y_k всем остальным участникам. После этого каждый участник P_i переопределяет своё значение y_i следующим образом: если $D_i^{y_i} < n - t$, то P_i переопределяет $y_i := y_k$; в противном случае y_i остаётся прежним.
Каждый участник P_i определяет $x_i := y_i$ (как входное значение для следующего шага итерации).

Теорема 1 [2]. При $t < n/3$ протокол 1 является t-безопасным протоколом византийского соглашения с коммуникационной сложностью $O(n^3)$ бит.

2. Широковещательная передача

Широковещательная передача позволяет участнику отправлять одно и то же сообщение всем участникам, и все участники уверены, что они получили одинаковые сообщения. Предполагать наличие широковещательного канала разумно только в ограниченных условиях, например, когда участники географически близки и могут использовать радиоволны. В большинстве случаев, особенно при выполнении протокола через Интернет, участники должны реализовывать широковещательный канал по каналам «точка — точка» (point-to-point network). Например, эмуляцию широковещательных протоколов можно создавать, используя каналы типа «точка — точка», при наличии инфраструктуры открытых ключей и электронных подписей. Один из таких протоколов приведён далее.

Отказоустойчивая широковещательная передача позволяет участнику распределять некоторое значение (сообщение) между набором участников, не доверяющих друг другу, которые попарно соединены каналами типа «точка — точка». Формальным требованием широковещательного протокола является то, что в конце протокола

все участники должны договориться о распределённом значении между всеми участниками. При этом должно быть гарантировано выполнение договорённости, даже если участник, который распределяет некоторое значение, является нечестным. Протоколы безопасных многосторонних вычислений обычно разрабатываются с учётом наличия широковещательных каналов. Однако в реальных сетях взаимодействие между участниками обычно происходит только с помощью каналов типа «точка — точка» и широковещательная передача должна эмулироваться с помощью защищённого протокола широковещательной передачи.

Определение 2 (широковещательная передача). Протокол для участников $\mathcal{P} = \{P_1, \ldots, P_n\}$, в котором назначенный участник (дилер) $D \in \mathcal{P}$ имеет некоторое (входное) значение M, называется протоколом *широковещательной передачи*, который является устойчивым к действиям активного противника (т. е. t-безопасной широковещательной передачи), контролирующего до t участников, если выполнены следующие условия:

1) *достоверность* (validity): если дилер честный, то все честные участники в качестве своего итогового (выходного) значения определят значение дилера M;
2) *договоренность* (agreement): даже если дилер нечестный, то итоговые (выходные) значения всех честных участников будут одинаковыми.

Пусть n участников $P_1, \ldots, P_n$ синхронно обменивается данными по защищённым и аутентифицированным каналам в полностью подключенной сети «точка — точка». Если канал, соединяющий двух участников, является защищённым, то противник ничего не может узнать о сообщениях, которыми обмениваются эти два участника. Аутентифицированный канал гарантирует, что никто не сможет изменить сообщение, передаваемое по каналу, во время его передачи. Отметим, что защищённые и аутентифицированные каналы могут быть реализованы с использованием криптографических примитивов, таких, как шифрование и электронная подпись.

Замечание 1. Если $t < n/2$, то на основе протокола t-безопасной широковещательной передачи можно построить протокол t-безопасного византийского соглашения. В этом случае каждый участник транслирует свое входное значение. Тогда каждый участник в качестве выходного значения определяет то значение, которое во время n трансляций повторяется чаще всего.

Обратно, на основе протокола t-безопасного византийского соглашения можно построить протокол t-безопасной широковещательной передачи. В этом случае участник (отправитель) D передаёт сообщение M всем участникам. После этого все участники запускают протокол византийского соглашения, в котором входным значением является значение, полученное от D.

Замечание 2. Наиболее популярным является предположение об аутентифицированной настройке, например, наличие инфраструктуры с открытыми ключами (Public-Key Infrastructure, PKI) для n участников. В этом случае все участники имеют набор из n открытых ключей для схемы подписи, где i-й ключ соответствует i-му участнику. Каждый честный участник имеет секретный ключ, сгенерированный честным путём, связанный с его собственным открытым ключом. Нечестные участники могут генерировать свои ключи произвольно.

При отсутствии аутентифицированной настройки t-безопасная широковещательная передача и t-безопасное византийское соглашение существуют тогда и только тогда, когда $t < n/3$ [2, 3]. Примером такого протокола византийского соглашения явля-

ется протокол 1 (BGP), на основе которого можно построить протокол широковещательной передачи без аутентифицированной настройки с учётом замечания 1.

При наличии аутентифицированной настройки при $t < n$ возможна t-безопасная широковещательная передача (протокол 2), а при $t < n/2$ возможно t-безопасное византийское соглашение (протокол 2 с учётом замечания 1).

Аутентифицированная настройка существует в двух вариантах: *информационно-теоретическая* и *криптографическая*. Информационно-теоретическая безопасность достигается с помощью использования псевдоподписи (pseudo-signature scheme) [4]. Криптографическая (вычислительная) безопасность достигается с помощью использования безопасной схемы электронной подписи, которую (почти) невозможно подделать.

Рассмотрим протокол широковещательной передачи, который устойчив к активному противнику, контролирующему до t участников. Наличие активного противника означает, что любой нечестный участник, находящийся под контролем противника, может произвольно отклоняться от предписанного протокола.

Модифицированный протокол Долева — Стронга (Dolev — Strong).

Пусть $P_1, \ldots, P_n$ — участники протокола широковещательной передачи; $D \in \{P_1, \ldots, P_n\}$ — дилер, который собирается транслировать бит (сообщение) $m \in \{0, 1\}$; Sign — некоторый алгоритм электронной подписи; sk_i — секретный ключ электронной подписи участника P_i, $i = 1, \ldots, n$. Приведём модифицированную версию протокола Долева — Стронга [5] в синхронной настройке для аутентифицированной широковещательной передачи для случая $t < n$.

Протокол 2 (модифицированный протокол Долева — Стронга).

Пусть дилер D обладает сообщением $m \in \{0, 1\}$.

— Э т а п 1 (действия дилера D и каждого участника P_i)
  — Дилер D отправляет всем участникам P_i подписанное сообщение $(m, \mathrm{Sign}_{sk_D}(m))$.
  — Каждый участник P_i определяет и инициализирует три множества: $AS_i = SET_0^i = SET_1^i = \varnothing$, где AS_i — одобренное участником P_i множество значений (Accepted Set); SET_0^i, SET_1^i — множества подписанных разными участниками сообщений $m = 0$ и $m = 1$ соответственно.

— Э т а п 2 (действия каждого участника P_i)
  В раундах $r = 1, \ldots, N + 1$ выполняются следующие шаги (в начале раунда с номером $r = 1$ все участники получили от дилера соответствующее сообщение):
  — Пусть участник P_i в r-м раунде получил от некоторого участника P_j сообщение (x, SET), где $x \in \{0, 1\}$; $SET = \{\mathrm{Sign}_{sk_{i_1}}(x), \ldots, \mathrm{Sign}_{sk_{i_s}}(x)\}$ — множество подписей, поставленных под сообщением x различными участниками $P_{i_1}, \ldots, P_{i_s}$, включая дилера D, причём $s \geqslant r$. Тогда участник P_i переопределяет свои множества следующим образом: $AS_i := AS_i \cup \{x\}$, $SET_x^i := SET_x^i \cup SET$.
  — Если $s < r$ или в сообщении (x, SET) нет подписи дилера D, то полученное сообщение игнорируется.
  — Если добавленного в множество AS_i сообщения x в этом множестве ранее не было (к началу r-го раунда), то участник P_i подписывает это сообщение $\mathrm{Sign}_{sk_i}(x)$, после чего всем остальным участникам передаёт сообщение $(x, SET_x^i \cup \{\mathrm{Sign}_{sk_i}(x)\})$.

— Э т а п 3 (действия каждого участника P_i)
  Если $AS_i = \{1\}$, то участник P_i в качестве своего итогового значения m_i определяет $m_i = 1$, в противном случае $m_i = 0$.

Теорема 2 [5]. Если $t < n$ и $N = t$, то модифицированный протокол Долева — Стронга является протоколом широковещательной передачи даже при наличии активного противника, контролирующего до t участников.

Протокол Долева — Стронга. Обобщим протокол 2 до случая сообщения m произвольной длины l.

Протокол 3 (протокол Долева — Стронга).
Пусть дилер D обладает сообщением $m \in \{0,1\}^l$.

— Э т а п 1 (действия дилера D и каждого участника P_i)
  — Дилер D отправляет всем участникам P_i подписанное сообщение $(m, \mathrm{Sign}_{sk_D}(m))$.
  — Каждый участник P_i определяет и инициализирует множество $AS_i = \varnothing$ — одобренное участником P_i множество сообщений.

— Э т а п 2 (действия каждого участника P_i)
  В раундах $r = 1, \ldots, N+1$ выполняются следующие шаги (в начале раунда с номером $r = 1$ все участники получили от дилера соответствующее сообщение):
  — Пусть участник P_i в r-м раунде получил от участника P_j сообщение $(x, \mathrm{Sign}_{sk_{i_1}}(x), \ldots, \mathrm{Sign}_{sk_{i_s}}(x))$, где $\{\mathrm{Sign}_{sk_{i_1}}(x), \ldots, \mathrm{Sign}_{sk_{i_s}}(x)\}$ — множество подписей, поставленных под сообщением x различными участниками $P_{i_1}, \ldots, P_{i_s}$, включая дилера D, причём $s \geqslant r$. Если $|AS_i| < 2$, то участник P_i переопределяет множество $AS_i := AS_i \cup \{x\}$, подписывает x своей подписью $\mathrm{Sign}_{sk_i}(x)$ и передаёт сообщение $(x, \mathrm{Sign}_{sk_{i_1}}(x), \ldots, \mathrm{Sign}_{sk_{i_s}}(x), \mathrm{Sign}_{sk_i}(x))$ всем остальным участникам.
  — Если $|AS_i| = 2$, или $s < r$, или в сообщении $(x, \mathrm{Sign}_{sk_{i_1}}(x), \ldots, \mathrm{Sign}_{sk_{i_s}}(x))$ нет подписи дилера D, то полученное сообщение игнорируется.

— Э т а п 3 (действия каждого участника P_i)
  Если $AS_i = \{x\}$ — одноэлементное множество, то участник P_i в качестве своего итогового значения m_i определяет $m_i = x$, в противном случае $m_i = 0$.

Теорема 3 [6]. Если $t < n$ и $N = t$, то протокол Долева — Стронга является протоколом широковещательной передачи даже при наличии активного (адаптивного) противника, контролирующего до t участников.

3. Широковещательная передача с прерыванием

В случае нечестного большинства и активного противника существуют протоколы безопасных многосторонних вычислений, которые не обладают свойствами гарантированного получения результатов и справедливости [7]. Следовательно, многие такие протоколы просто прерываются при обнаружении обмана, реализуя безопасность с прерыванием (security with abort) [8]. В частности, это означает, что либо протокол завершается успешно и каждый участник получит свои выходные данные, либо протокол прерывается, причём это может произойти даже после того, как противник узнал результаты вычислений, что может стать серьёзной проблемой в некоторых приложениях.

Учитывая соглашение о безопасности с прерыванием, широковещательный канал может быть эффективно реализован с использованием стандартного 2-раундового протокола *эхо-трансляции* [8], который приведён далее.

Определение 3 (широковещательная передача с прерыванием). Протокол для участников $\mathcal{P} = \{P_1, \ldots, P_n\}$, в котором назначенный участник (дилер) $D \in \mathcal{P}$ имеет

некоторое (входное) значение M, называется протоколом *широковещательной передачи с прерыванием*, который является устойчивым к действиям активного противника, контролирующего до t участников, если выполнены следующие условия:

1) *достоверность* (validity): если дилер честный, то каждый честный участник в качестве своего итогового (выходного) значения определит либо значение дилера M, либо $\perp$;
2) *договоренность* (agreement): если некоторый честный участник в качестве своего итогового значения определит $\widetilde{M}$, то каждый честный участник в качестве своего итогового значения определит либо $\widetilde{M}$, либо $\perp$;
3) *нетривиальность* (non-triviality): если все участники являются честными (включая дилера), то все участники в качестве своих итоговых значений определят M.

Протокол 4 (трансляция с прерыванием).

Пусть дилер D обладает сообщением M, предназначенным для трансляции.

— Дилер D передаёт сообщение M каждому участнику.
— Обозначим через M_i сообщение, полученное участником P_i от дилера D на предыдущем шаге. Если участник P_i не получил ничего от дилера на предыдущем шаге, то полагается $M_i = \perp$.

Каждый участник P_i передаёт сообщение M_i всем остальным участникам.

— Обозначим через M_{ji} сообщение, полученное участником P_i от участника P_j на предыдущем шаге. Участник P_i в качестве выходного значения определяет M_i, если для любого $j = 1, \ldots, n$, $j \neq i$, выполнено $M_{ji} = M_i$. В противном случае участник P_i определяет $\perp$.

Утверждение 1 [8]. Протокол 4 является протоколом широковещательной передачи с прерыванием, который является устойчивым к действиям активного противника, контролирующего до $t < n$ участников.

4. Протокол расширения для случая $t < n/3$

Исследуем протоколы расширения для византийского соглашения и широковещательной передачи. Это связано с задачей эффективной трансляции и византийским соглашением для длинных входных сообщений, поскольку такие протоколы широко используются. Простым решением трансляции l-битного сообщения является трансляция каждого бита в отдельности. Такой подход требует коммуникационной сложности $\Omega(ln^2)$ бит, где n — число участников, так как коммуникационная сложность для однобитного сообщения составляет $\Omega(n^2)$ бит [9]. Протоколы расширения для широковещательной передачи и византийского соглашения представляют собой конструкции для длинных сообщений, построенные на основе соответствующих протоколов для коротких сообщений (в частности, однобитных) и каналов связи типа «точка — точка».

Рассмотрим протокол византийского соглашения для случая $t < n/3$, который является оптимальным как для коммуникационной, так и для раундовой сложности [10]. Данный протокол имеет информационно-теоретическую безопасность, причём вероятность ошибки равна нулю (error-free), т. е. является детерминированным. Он основан на методах теории кодирования и теории графов. В протоколе использован алгоритм поиска (n, t)-звезды и коды Рида — Соломона.

Поиск (n, t)-звезды

Определим структуру данных под названием (n, t)-*звезда*, которую иногда будем называть просто звездой.

Пусть G является неориентированным графом с множеством вершин $\{P_1, \ldots, P_n\}$; $\mathcal{C} \subseteq \mathcal{D} \subseteq \{P_1, \ldots, P_n\}$ — некоторые подмножества вершин графа. Пара $(\mathcal{C}, \mathcal{D})$ называется (n,t)-*звездой*, если $|\mathcal{C}| \geqslant n-2t$, $|\mathcal{D}| \geqslant n-t$ и для любых $P_i \in \mathcal{C}$, $P_j \in \mathcal{D}$ ребро (P_i, P_j) принадлежит графу G.

Кликой в неориентированном графе $G = (V, E)$ называется подмножество вершин $\mathcal{C} \subseteq V$, для которого для любых двух вершин в $\mathcal{C}$ существует ребро, их соединяющее. Из определения (n,t)-звезды $(\mathcal{C}, \mathcal{D})$ следует, что $\mathcal{C}$ является кликой. Понятие звезды обобщает понятие клики. Если надмножество $\mathcal{D}$ множества $\mathcal{C}$ является кликой, то $(\mathcal{C}, \mathcal{D})$ является звездой.

В работе [11] приведён эффективный алгоритм проверки наличия звезды в графе при условии, что граф содержит клику размера не менее $n-t$ (алгоритм 1).

Напомним несколько понятий. Граф $\overline{G} = (V, \overline{E})$ называется дополнительным графом к графу $G = (V, E)$, если

$$\overline{E} = \{(u,v) : u, v \in V, u \neq v, (u,v) \notin E\},$$

т. е. дополнительный граф $\overline{G}$ содержит те же вершины, что и граф G, и любые две различные вершины в нем смежны в том и только в том случае, когда эти вершины не смежны в графе G. Множество вершин $U \subseteq V$ называется независимым, если никакие две его вершины не смежны. Множество рёбер $M \subseteq E$ называется паросочетанием, если никакие два его ребра не имеют общей вершины. Максимальное паросочетание в графе G — это такое паросочетание M, в котором количество входящих в его состав рёбер является максимальным среди всех паросочетаний в графе G. Две вершины называются соседями (или смежными вершинами), если в графе имеется ребро, их соединяющее.

Алгоритм 1. Эффективный алгоритм поиска звезды в графе

Вход: граф $G = (\{1, \ldots, n\}, E)$, параметр t.
Выход: звезда $(\mathcal{C}, \mathcal{D})$ или сообщение об её отсутствии.

1: Пусть $\overline{G} = (\{1, \ldots, n\}, \overline{E})$ — дополнительный граф; M — максимальное паросочетание в $\overline{G}$, найденное, например, с помощью алгоритма Эдмондса; N — множество всех вершин графа G, входящих в паросочетание M; $\overline{N} = \{1, \ldots, n\} \setminus N$.
2: Пусть T — множество таких вершин из $\overline{N}$, для каждой из которых найдутся вершины $j, k \in \{1, \ldots, n\}$, такие, что треугольник $(i,j), (i,k), (j,k)$ принадлежит графу $\overline{G}$ и $(j,k) \in M$: $T = \{i \in \overline{N} : \exists j, k\, (j,k) \in M, (i,j), (i,k) \in \overline{G}\}$. Пусть $\mathcal{C} = \overline{N} \setminus T$.
3: Пусть B — множество вершин из N, для которых найдутся смежные вершины в $\overline{G}$: $B = \{j \in N : \exists i \in \mathcal{C}\, (i,j) \in \overline{G}\}$; $\mathcal{D} = \{1, \ldots, n\} \setminus B$.
4: **Если** $|\mathcal{C}| \geqslant n-2t$ и $|\mathcal{D}| \geqslant n-t$, **то**
5: **Вернуть** звезду $(\mathcal{C}, \mathcal{D})$,
6: **иначе**
7: **Вернуть** сообщение об отсутствии звезды.

Коды Рида — Соломона

В протоколе византийского соглашения используется обобщённый $[n, t+1, n-t]$-код Рида — Соломона [12] над полем $F = \mathrm{GF}(2^m)$, $n \leqslant 2^m$. Каждый элемент поля F можно представить в виде двоичного вектора длины m. С помощью обобщённого кода Рида — Соломона информационное сообщение над полем F длины $t+1$ кодируется в сообщение длины n над полем F.

Протокол византийского соглашения

Приведём протокол расширения для византийского соглашения [10].

Протокол 5 (протокол византийского соглашения для $t < n/3$).
Пусть каждый участник P_i обладает сообщением $m_i \in \{0,1\}^l$.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.
Каждый участник P_i производит следующие действия:

1) l-Битное сообщение m_i разбивается на $t+1$ блоков $m_i = (m_{i0}, m_{i1}, \ldots, m_{it})$, где каждый блок m_{ij} имеет длину $l/(t+1)$ бит. С помощью обобщённого $[n, t+1, n-t]$-кода Рида — Соломона сообщение $(m_{i0}, m_{i1}, \ldots, m_{it})$ кодируется в кодовое сообщение $(s_{i1}, \ldots, s_{in})$. Значение s_{ii} передаётся всем участникам. Значение s_{ij} передаётся участнику P_j, $j = 1, \ldots, n$.
2) Двоичный вектор v_i длины n заполняется следующим образом:
$$v_i[j] = \begin{cases} 1, & s_{ij} = s_{jj} \text{ и } s_{ii} = s_{ji}, \\ 0 & \text{иначе}, \end{cases}$$
где s_{jj} и s_{ji} получено от P_j, $j = 1, \ldots, n$. С помощью оракула вектор v_i транслируется всем участникам.
3) Строится граф G с множеством вершин $\{P_1, \ldots, P_n\}$. Ребро (P_j, P_k) добавляется в граф, если $v_j[k] = 1$ и $v_k[j] = 1$ (j и k могут совпадать). Для графа G вызывается алгоритм 1. Возможны два случая:
   а) алгоритм 1 возвратил звезду $(\mathcal{C}, \mathcal{D})$. Вычисляется множество вершин $\mathcal{F}$ графа G, каждая из которых имеет не менее $t+1$ соседей из множества $\mathcal{C}$. Вычисляется множество вершин $\mathcal{E}$ графа G, каждая из которых имеет не менее $2t+1$ соседей из множества $\mathcal{F}$. Если $|\mathcal{E}| \geqslant 2t+1$, то $\mathcal{P}_{\text{sm}} := \mathcal{E}$ (“sm” означает “same message”). Если $|\mathcal{E}| < 2t+1$, то участник P_i полагает своим выходным значением заранее определённое значение $m^* \in \{0,1\}^l$ и прерывает протокол;
   б) звезда в графе G не найдена. Участник P_i полагает выходным значением заранее определённое значение $m^* \in \{0,1\}^l$ и прерывает протокол.
4) Пусть s_i — такое значение из множества $\{s_{ji} : P_j \in \mathcal{P}_{\text{sm}}\}$, которое в нём встречается чаще всего. Значение s_i передаётся всем остальным участникам.
5) Пусть $(s_1, \ldots, s_n)$ — вектор, в котором s_j получено от P_j, $j = 1, \ldots, n$. К нему применяется алгоритм декодирования для обобщённого $[n, t+1, n-t]$-кода Рида — Соломона, который возвращает информационное сообщение $(m_0, \ldots, m_t)$. Это сообщение определяется в виде выходного результата участника P_i.

Обозначим через $\mathcal{B}(s)$ коммуникационную сложность трансляции двоичного сообщения длины s. Простой конструкцией является трансляция (короткого) сообщения по одному биту. В этом случае коммуникационная сложность составляет $s\mathcal{B}(1)$ бит. Например, для протокола Долева — Стронга $\mathcal{B}(1) = O(n^2 + kn^3)$. Аналогичным образом через $\mathcal{A}(s)$ будем обозначать коммуникационную сложность византийского соглашения для двоичного сообщения длины s.

Теорема 4 [10]. Протокол 5 является (информационно-теоретически) t-безопасным протоколом византийского соглашения со следующими условиями: число раундов равно 3, коммуникационная сложность — $O\big(ln + n^2\mathcal{B}(1)\big)$ бит.

Замечание 3. Протокол широковещательной передачи получается на основе протокола византийского соглашения. В этом случае сначала отправитель передаёт всем участникам некоторое сообщение, а потом запускается протокол византийского соглашения.

5. Универсальная хеш-функция

Пусть $K = \{0, 1, \ldots, 2^\kappa - 1\}$ — множество ключей. Рассмотрим семейство функций $\mathcal{U} = \{U_k : k \in K\}$, где для каждого $k \in K$ функция U_k отображает элементы некоторого множества X в $\{0,1\}^\kappa$. Семейство $\mathcal{U}$ называется ε-*универсальным*, если для любых двух различных сообщений $m_1, m_2 \in X$ выполнено

$$\frac{|\{k \in K : U_k(m_1) = U_k(m_2)\}|}{|K|} \leqslant \varepsilon.$$

Данное условие (комбинаторного) определения универсальной хеш-функции можно записать на языке вероятностей:

$$\Pr[k \leftarrow K : U_k(m_1) = U_k(m_2)] \leqslant \varepsilon.$$

ε-Универсальную хеш-функцию можно построить следующим образом. Пусть $X = \{0,1\}^l$, $K = \mathrm{GF}(2^\kappa)$. Каждое сообщение $m \in X$ интерпретируется как многочлен f_m над $\mathrm{GF}(2^\kappa)$ степени не более $\lceil l/\kappa \rceil - 1$. Тогда значение хеш-функции U_k определяется следующим образом: $U_k(m) = f_m(k)$. Так как два различных многочлена степени не более $\lceil l/\kappa \rceil - 1$ могут совпадать не более чем в $\lceil l/\kappa \rceil - 1$ точках, для любых $m_1, m_2 \in X$, $m_1 \neq m_2$, выполнено

$$\frac{|\{k \in GF(2^\kappa) \mid U_k(m_1) = U_k(m_2)\}|}{2^\kappa} \leqslant \frac{\lceil l/\kappa \rceil - 1}{2^\kappa} \leqslant 2^{-\kappa} l.$$

Таким образом, построенное семейство $\mathcal{U}$ является $2^{-\kappa}l$-универсальной хеш-функцией.

6. Протоколы расширения для случая $t < n/2$

Рассмотрим протоколы расширения для византийского соглашения и широковещательной передачи для случая $t < n/2$. Коммуникационная сложность для однобитного сообщения составляет $\Omega(n^2)$ бит, поэтому если такой протокол использовать для каждого бита сообщения длины l, то получим коммуникационную сложность $\Omega(ln^2)$. Поэтому для длинных сообщений более практичны протоколы расширения для широковещательной передачи и византийского соглашения, которые имеют меньшую коммуникационную сложность, нежели $\Omega(ln^2)$. Более того, для достаточно больших l коммуникационная сложность таких протоколов составляет $O(ln)$ бит.

6.1. Информационно-теоретически безопасный протокол

Протокол [13] состоит из трёх этапов, каждый из которых «приближает» участников к соглашению. Протокол может быть прерван на любом из первых двух этапов при обнаружении (доказуемых) несоответствий между данными, предоставленными честными участниками. В этом случае каждый участник выбирает выходное сообщение по умолчанию, обозначаемое $\perp$. Если выполнение протокола дошло до третьего этапа, то в конце протокола все участники получат свои выходные значения. В протоколе используется ε-универсальная хеш-функция $\mathcal{U} = \{U_k : k \in K\}$. Пусть $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Протокол 6 (протокол византийского соглашения для $t < n/2$).
Пусть каждый участник P_i обладает сообщением $m_i \in \{0,1\}^l$.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

Этап проверок

1) Каждый участник P_i выбирает случайным образом ключ k_i для хеш-функции U_k и вычисляет $h_i = (k_i, U_{k_i}(m_i))$, которое транслируется всем участникам.

2) Каждый участник P_j создает двоичный вектор v_j длины n следующим образом:

$$v_j[i] = \begin{cases} 1, & U_{k_i}(m_j) = U_{k_i}(m_i), \\ 0 & \text{иначе}, \end{cases} \qquad i = 1, \ldots, n.$$

Должно быть выполнено $v_j[j] = 1$, $j = 1, \ldots, n$. Каждый участник P_j транслирует вектор v_j.

3) Если не менее $n-t$ транслируемых векторов являются одинаковыми и в каждом из таких одинаковых векторов v_j выполнено $v_j[j] = 1$, то этот вектор обозначим через v, а через $\mathcal{P}_{\text{sm}}$ — множество участников, которые транслировали вектор v. Если транслировалось менее $n - t$ одинаковых векторов, то протокол прерывается.

Этап укрупнения

Пусть $\phi : \mathcal{P} \setminus \mathcal{P}_{\text{sm}} \to \mathcal{P}_{\text{sm}}$ — некоторая инъективная функция. С её помощью участник $\phi(P_j) \in \mathcal{P}_{\text{sm}}$ помогает участнику $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ получить корректное сообщение m:

1) Для каждого участника $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ участник $P_i = \phi(P_j)$ передаёт сообщение m_i участнику P_j, который обозначает полученное сообщение через $\widetilde{m}_j$.
2) Каждый участник $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ выбирает случайным образом ключ k_j, вычисляет и транслирует сообщение $(k_j, U_{k_j}(\widetilde{m}_j))$.
3) Каждый участник $P_i \in \mathcal{P}_{\text{sm}}$ создаёт двоичный вектор v_i длины $|\mathcal{P} \setminus \mathcal{P}_{\text{sm}}|$ следующим образом:

$$v_i[j] = \begin{cases} 1, & U_{k_j}(m_i) = U_{k_j}(\widetilde{m}_j), \\ 0 & \text{иначе}, \end{cases} \qquad j = 1, \ldots, |\mathcal{P} \setminus \mathcal{P}_{\text{sm}}|.$$

Каждый участник P_i транслирует вектор v_i.

4) Если не менее $n-t$ транслируемых векторов являются одинаковыми, то обозначим этот вектор через v, а через $\mathcal{P}_{\text{rej}} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\text{sm}})$ — множество всех участников P_j, для которых j-я компонента вектора v равна нулю. Пусть

$$\mathcal{P}_{\text{ok}} = \mathcal{P} \setminus \mathcal{P}_{\text{rej}} \setminus \{\phi(P_j) : P_j \in \mathcal{P}_{\text{rej}}\}.$$

Заметим, что множество

$$\mathcal{P}_{\text{conf}} = \mathcal{P}_{\text{rej}} \cup \{\phi(P_j) : P_j \in \mathcal{P}_{\text{rej}}\} = \{P_j, \phi(P_j) : P_j \in \mathcal{P}_{\text{rej}}\}$$

состоит из участников, не менее половины которых являются нечестными, т. е. из каждой пары участников P_j и $\phi(P_j)$, $P_j \in \mathcal{P}_{\text{rej}}$, хотя бы один нечестный.
Каждый участник $P_i \in \mathcal{P}_{\text{sm}} \cap \mathcal{P}_{\text{ok}}$ определяет имеющееся у него сообщение m_i как выходное, а каждый участник $P_i \in \mathcal{P}_{\text{ok}} \setminus \mathcal{P}_{\text{sm}}$ определяет своё выходное сообщение $m_i = \widetilde{m}_i$. Если транслировалось менее $n - t$ одинаковых векторов, то протокол прерывается.

Этап утверждения сообщений

1) Каждый участник $P_i \in \mathcal{P}_{\text{ok}}$ вычисляет $d = \lceil (|\mathcal{P}_{ok}| + 1)/2 \rceil$, $c = \lceil (l+1)/d \rceil$ и многочлен f_m, где $m = (m_0, m_1, \ldots, m_{d-1})$, $f_m(x) = m_0 + m_1 x + \ldots + m_{d-1} x^{d-1}$. После этого P_i передаёт $y_i = f_m(i)$ (вектор длины c) каждому участнику множества $\mathcal{P} \setminus \mathcal{P}_{\text{ok}}$.
2) Каждый участник $P_i \in \mathcal{P}_{\text{ok}}$ выбирает случайным образом ключ k_i и передаёт набор $\big(k_i, U_{k_i}(f_m(1)), \ldots, U_{k_i}(f_m(n))\big)$ каждому участнику множества $\mathcal{P} \setminus \mathcal{P}_{\text{ok}}$.

3) Каждый участник $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{ok}}$ для каждого y_i, полученного от $P_i \in \mathcal{P}_{\text{ok}}$, делает следующее: участник P_j обладает $|\mathcal{P}_{\text{ok}}|$ наборами вида

$$\big(k_i, U_{k_i}(f_m(1)), \ldots, U_{k_i}(f_m(n))\big), \quad P_i \in \mathcal{P}_{\text{ok}}.$$

Значения $U_{k_i}(y_i)$ сравниваются с $U_{k_i}(f_m(i))$. Если не менее $d = \lceil (|\mathcal{P}_{\text{ok}}| + 1)/2 \rceil$ наборов содержат значения вида $U_{k_i}(y_i)$, то значение y_i принимается, в противном случае оно отвергается. После этого P_j интерполирует многочлен f_m по не менее d значениям y_i, восстанавливая при этом некоторое сообщение $\widetilde{m}_j$.

Опишем этапы протокола 6.

Этап проверок. Участники сравнивают между собой сообщения m_i с помощью сравнения образов хеш-функции и совместно определяют подмножество участников $\mathcal{P}_{\text{sm}}$, чьи сообщения равны между собой (вернее, равны образы хеш-функции). Этот этап может быть прерван при обнаружении несоответствий между сообщениями честных участников. Пусть $\mathcal{P}_{\text{hon}}$ — все честные участники. Более формально, этап проверок удовлетворяет следующим требованиям:

— если все честные участники $P_i \in \mathcal{P}_{\text{hon}}$ обладают одинаковыми входными сообщениями $m_i = m$, то этап проверок не будет прерван;
— все честные участники $P_i \in \mathcal{P}_{\text{hon}} \cap \mathcal{P}_{\text{sm}}$ обладают одинаковыми входными сообщениями $m_i = m$;
— если все честные участники $P_i \in \mathcal{P}_{\text{hon}}$ обладают одинаковыми входными сообщениями $m_i = m$, то все они принадлежат множеству $\mathcal{P}_{\text{sm}}$, т. е. $\mathcal{P}_{\text{hon}} \subseteq \mathcal{P}_{\text{sm}}$.

Этап укрупнения. Участники из множества $\mathcal{P}_{\text{sm}}$ помогают другим участникам (множества $\mathcal{P} \setminus \mathcal{P}_{\text{sm}}$) получить правильное сообщение. Это приводит к множеству участников $\mathcal{P}_{\text{ok}}$, состоящему из участников с одинаковыми сообщениями, причём большинство участников $\mathcal{P}_{\text{ok}}$ являются честными. Данный этап также может быть прерван при обнаружении несоответствий между сообщениями честных участников. Этап укрупнения удовлетворяет следующим требованиям:

— если все честные участники попали в множество $\mathcal{P}_{\text{sm}}$ на этапе проверок (т. е. выполнено $\mathcal{P}_{\text{hon}} \subseteq \mathcal{P}_{\text{sm}}$), то этап укрупнения завершается без прерывания;
— все честные участники $P_i \in \mathcal{P}_{\text{hon}} \cap \mathcal{P}_{\text{ok}}$ обладают одинаковыми выходными сообщениями $m_i = m$;
— выходные сообщения участников множества $\mathcal{P}_{\text{hon}} \cap \mathcal{P}_{\text{sm}}$ совпадают с сообщением m;
— большинство участников множества $\mathcal{P}_{\text{ok}}$ являются честными, т. е. выполнено неравенство $|\mathcal{P}_{\text{ok}} \cap \mathcal{P}_{\text{hon}}| > |\mathcal{P}_{\text{ok}}|/2$. Это следует из того, что не менее половины участников множества $\mathcal{P}_{\text{conf}} = \{P_j, \phi(P_j) : P_j \in \mathcal{P}_{\text{rej}}\}$ являются нечестными.

Этап утверждения сообщений. Участники множества $\mathcal{P} \setminus \mathcal{P}_{\text{ok}}$ должны иметь возможность получить сообщение m, которым владеют участники множества $\mathcal{P}_{\text{ok}}$. Но при этом участники из множества $\mathcal{P} \setminus \mathcal{P}_{\text{ok}}$ не должны запрашивать сообщения от любых участников множества $\mathcal{P}_{\text{ok}}$. Это связано с тем, что нечестные участники могут злоупотребить этой возможностью, запросив сообщения у каждого участника из $\mathcal{P}_{\text{ok}}$, что приведёт к коммуникационной сложности $\Omega(ln^2)$ бит. Для уменьшения этой сложности применён трюк с использованием многочлена.

Пусть $l \approx c \cdot d$ для подходящих c и d. Представим сообщение m в виде $m = (m_0, m_1, \ldots, m_{d-1})$, где m_i имеет длину c бит, $i = 0, 1, \ldots, d-1$. Рассмотрим поле $F = \text{GF}(2^c)$. Сообщению m поставим в соответствие многочлен $f_m(x) = m_0 + m_1 x + \ldots + m_{d-1} x^{d-1} \in \in F[x]$. Для восстановления многочлена $f_m(x)$ (следовательно, сообщения m) необходимо знать d значений многочлена f_m в различных точках. Пусть $d = \lceil (|\mathcal{P}_{\text{ok}}| + 1)/2 \rceil$,

$c = \lceil (l+1)/d \rceil$. В данном случае к сообщению добавляется дополнительный бит. Каждый участник $P_i \in \mathcal{P}_{\rm ok}$ передаёт значение $f_m(i)$ каждому участнику из множества $\mathcal{P} \setminus \mathcal{P}_{\rm ok}$. Каждый участник $P_j \in \mathcal{P} \setminus \mathcal{P}_{\rm ok}$ (с помощью участников из множества $\mathcal{P}_{\rm ok}$) может отличить корректные значения $f_m(i)$ от некорректных и в конечном итоге интерполировать многочлен f_m, получая при этом сообщение m.

Теорема 5 [13]. Протокол 6 является (информационно-теоретически) t-безопасным протоколом византийского соглашения с коммуникационной сложностью $O\big(ln + n^3k + (n^2+nk)\mathcal{B}(1)\big)$ бит, где k — параметр безопасности.

6.2. Криптографически безопасный протокол

Рассмотрим протокол, который является модификацией протокола 6 и имеет немного меньшую коммуникационную сложность, но обладает криптографической (вычислительной) безопасностью [10]. Пусть H — функция хеширования, устойчивая к обнаружению коллизий.

Протокол 7 (протокол византийского соглашения для $t < n/2$).
Пусть каждый участник P_i обладает сообщением $m_i \in \{0,1\}^l$.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

Этап проверок. Каждый участник P_i делает следующее.

1) Вычисляется значение функции хеширования $h_i = H(m_i)$, которое транслируется всем участникам.
2) Происходит проверка, сколько одинаковых свёрток транслируется. Если их не менее $n-t$, то обозначим эту свёртку через h, а через $\mathcal{P}_{\rm sm}$ — множество участников, которые транслировали свертку h. Если транслировалось менее $n-t$ одинаковых свёрток, то протокол прерывается.

Этап договорённости. Пусть $\phi : \mathcal{P} \setminus \mathcal{P}_{\rm sm} \to \mathcal{P}_{\rm sm}$ — некоторая инъективная функция. Например, участники множества $\mathcal{P} \setminus \mathcal{P}_{\rm sm}$ перебираются по порядку следования их индексов, которые отображаются в соответствующих участников множества $\mathcal{P}_{\rm sm}$, которые тоже перебираются по порядку следования индексов. С помощью этой функции участнику $P_j \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$ получить корректное сообщение m помогает участник $\phi(P_j) \in \mathcal{P}_{\rm sm}$.

Каждый участник P_i делает следующее:

1) Если $P_i \in \mathcal{P}_{\rm sm}$, то он определяет своё сообщение m_i как выходное.
2) Если $P_i \in \mathcal{P}_{\rm sm}$ и $P_i = \phi(P_j)$, то P_i передаёт сообщение m_i участнику P_j. Обозначим полученное участником P_j сообщение через $\widetilde{m}_j$.
3) Если $P_i \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$ и он получил на предыдущем шаге от участника P_j сообщение $\widetilde{m}_i = m_j$, где $P_j = \phi(P_i)$, то P_i проверяет, выполнено ли равенство $H(\widetilde{m}_i) = h$. Если равенство выполнено, то P_i транслирует значение 1 и определяет своё выходное сообщение $m_i = \widetilde{m}_i$, в противном случае P_i транслирует значение 0.
4) Строится множество $\mathcal{P}_{\rm conf}$, которое состоит из пар вида $\{P_j, \phi(P_j)\}$, $P_j \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$, причём P_j транслировал 0. Пусть $\mathcal{P}_{\rm ok} = \mathcal{P} \setminus \mathcal{P}_{\rm conf}$, $d = \lceil (|\mathcal{P}_{\rm ok}| + 1)/2 \rceil$, $c = \lceil (l+1)/d \rceil$.
5) Если $P_i \in \mathcal{P}_{\rm ok}$, то сообщение m_i преобразуется в многочлен $f_i \in {\rm GF}(2^c)$ степени не более $d-1$. Вычисляется значение $y_i = f_i(i)$ и вектор $H_i = \big(H(f_i(1)), \ldots, H(f_i(n))\big)$. Набор (y_i, H_i) передаётся каждому $P_j \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$, который транслировал значение 0.
6) Если $P_i \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$ и P_i транслировал 0, то для каждого y_j, полученного от $P_j \in \mathcal{P}_{\rm ok}$, делается следующее: участник P_i обладает $|\mathcal{P}_{\rm ok}|$ наборами вида $\big(H(f_j(1)), \ldots, H(f_j(n))\big)$, $P_j \in \mathcal{P}_{\rm ok}$. Значения $H(y_j)$ сравниваются с $H(f_j(j))$.

Если не менее $d = \lceil(|\mathcal{P}_{\text{ok}}| + 1)/2\rceil$ наборов содержат значения вида $H(y_j)$, то значение y_j принимается, в противном случае оно отвергается. После этого P_i интерполирует многочлен f по не менее d значениям y_j, восстанавливая некоторое сообщение $\widetilde{m}_i$, которое является его выходным значением.

Этап проверок обладает следующими свойствами:

— если все честные участники P_i начнут этап проверок с одинаковым сообщением $m_i = m$, то этап не будет прерван, причём все честные участники попадут в множество $\mathcal{P}_{\text{sm}}$;
— все честные участники множества $\mathcal{P}_{\text{sm}}$ обладают одинаковыми входными сообщениями.

Этап договоренности обладает следующими свойствами:

— большинство участников множества $\mathcal{P}_{\text{ok}}$ являются честными;
— выходные значения честных участников множеств $\mathcal{P}_{\text{sm}}$ и $\mathcal{P}_{\text{ok}}$ одинаковы;
— все честные участники обладают одинаковыми выходными сообщениями.

Теорема 6 [10]. Протокол 7 (за исключением пренебрежимо малой вероятности от параметра k) является (криптографически) t-безопасным протоколом византийского соглашения с коммуникационной сложностью $O\big(ln + n^3k + nk\mathcal{B}(1)\big)$ бит, где k — параметр безопасности.

7. Протоколы расширения широковещательной передачи для случая $t < n$

Рассмотрим криптографически (вычислительно) и информационно-теоретически безопасные протоколы расширения широковещательной передачи в случае $t < n$ [14]. На высоком уровне обе конструкции работают следующим образом: длинное сообщение (которое требуется транслировать) разбивается на блоки, а к каждому блоку применяется специальный протокол для трансляции блока.

В протоколах используется система контроля для разрешения споров между участниками (когда участники обладают разными значениями). Пусть Δ — множество неупорядоченных пар участников, причём $\{P_i, P_j\} \in \Delta$ тогда и только тогда, когда у P_i и P_j происходит спор по поводу того, чьё сообщение является правильным. В начале протокола множество Δ пустое. Во время выполнения протокола при возникновении спора соответствующая пара участников добавляется в это множество. Будем говорить, что множество Δ является *допустимым*, если для каждой пары $\{P_i, P_j\} \in \Delta$ участники P_i и P_j находятся в споре.

7.1. Криптографически безопасный протокол

Сначала рассмотрим протокол CryptoBlockBC, который предназначен для трансляции некоторого блока данных. Он вызывается в протоколе CryptoBC, где сообщение длины l разбивается на блоки и для каждого блока вызывается протокол CryptoBlockBC. В начале протокола CryptoBC множество Δ пусто. Если при вызове протокола CryptoBlockBC некоторая пара $\{P_i, P_j\}$ попадает в Δ, то она там остаётся до окончания протокола CryptoBC, т. е. множество Δ является глобальной переменной относительно протоколов CryptoBlockBC. Это значит, что если у участников P_i и P_j возник спор относительно значения некоторого блока, то этот спор продолжается и при вызове CryptoBlockBC для других блоков некоторого сообщения длины l.

В протоколе CryptoBlockBC используется устойчивая к коллизиям функция хеширования H. Пусть $\mathcal{P} = \{P_1, \ldots, P_n\}$ — множество участников; $\mathcal{P}_{\text{sm}}$ — множество участников, у которых сообщения m_i совпадают.

Протокол 8 (протокол CryptoBlockBC(m)).
Пусть дилер D обладает сообщением m.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

1) Каждый участник P_i инициализирует множество $\mathcal{P}_{\rm sm} = \{D\}$.
2) Дилер D транслирует значение хеш-функции $h = H(m)$.
3) Цикл: пока существует хотя бы одна пара различных участников $P_i, P_j \in \mathcal{P}$ с условием $P_i \in \mathcal{P}_{\rm sm}$, $P_j \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$ и $\{P_i, P_j\} \notin \Delta$, делается следующее:
   а) участник P_i передаёт сообщение m_i участнику P_j; обозначим полученное участником P_j сообщение через m_j;
   б) если $H(m_j) = h$, то участник P_j транслирует значение 1, в противном случае — значение 0;
   в) если P_j транслирует 1, то все участники добавляют участника P_j в множество $\mathcal{P}_{\rm sm}$, в противном случае все участники добавляют пару $\{P_i, P_j\}$ в множество Δ.
4) Каждый участник $P_i \in \mathcal{P}_{\rm sm}$ определяет в качестве выходного значения сообщение m_i. Каждый участник $P_i \in \mathcal{P} \setminus \mathcal{P}_{\rm sm}$ определяет в качестве выходного значения $\perp$.

Лемма 1 [14]. Пусть Δ — допустимое множество пар участников на момент вызова протокола CryptoBlockBC для блока m, $\Delta_{\rm e}$ — соответствующее множество на момент окончания протокола CryptoBlockBC для сообщения m, H — устойчивая к коллизиям функция хеширования с параметром безопасности k (длина свёртки). Тогда протокол CryptoBlockBC является t-безопасным протоколом широковещательной передачи, множество $\Delta_{\rm e}$ является допустимым, число раундов протокола CryptoBlockBC равно $O(n+d)$, коммуникационная сложность составляет $\mathcal{B}(k)+(n+d)(|m|+\mathcal{B}(1))$ бит, где $d = |\Delta_e| - |\Delta|$; $|m|$ — длина блока m.

Теперь рассмотрим протокол CryptoBC. Пусть q — некоторый параметр, $q \leqslant l$.

Протокол 9 (протокол CryptoBC).
Пусть дилер D обладает сообщением $m \in \{0,1\}^l$.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

1) Участники инициализируют множество Δ в виде пустого множества.
2) Дилер D разбивает сообщение m на q блоков $m = (m_1, \ldots, m_q)$.
3) Для $i = 1, \ldots, q$ вызывается протокол CryptoBlockBC(m_i). Пусть m_j^i — выходное сообщение участника P_j после i-го шага, $i = 1, \ldots, q$.
4) Каждый участник P_j в качестве итогового выходного сообщения определяет следующее: если для некоторого i выполнено $m_j^i = \perp$, то $m_j = \perp$, в противном случае $m_j = (m_j^1, \ldots, m_j^q)$.

Ввиду леммы 1 коммуникационная сложность протокола CryptoBC не превышает

$$\sum_{i=1}^{q}\Big(\mathcal{B}(k) + (n+d_i)(l/q + \mathcal{B}(1))\Big) = q\mathcal{B}(k) + \Big(qn + \sum_{i=1}^{q} d_i\Big)(l/q + \mathcal{B}(1)) \text{ бит.}$$

Так как $\sum\limits_{i=1}^{q} d_i \leqslant n^2$, коммуникационная сложность ограничена сверху числом $q\mathcal{B}(k) + (qn + n^2)(l/q + \mathcal{B}(1))$. При $q = n$ коммуникационная сложность ограничена сверху числом $2ln + n\mathcal{B}(k) + 2n^2\mathcal{B}(1)$.

Так как число раундов протокола CryptoBlockBC при i-м вызове равно $O(n + d_i)$, причём $\sum\limits_{i=1}^{q} d_i \leqslant n^2$, то число раундов протокола CryptoBC равно $O(n^2)$.

Таким образом, получаем следующее утверждение:

Теорема 7. При $t < n$ протокол CryptoBC при $q = n$ является (вычислительно) t-безопасным протоколом широковещательной передачи для сообщения длины l бит с числом раундов $O(n^2)$ и коммуникационной сложностью $O\big(ln + (n^2 + nk)\mathcal{B}(1)\big)$ бит, где k — параметр безопасности.

7.2. Информационно-теоретически безопасный протокол

Как и для случая криптографически безопасного протокола, все участники во время протокола ITBlockBC делятся на подмножества $\mathcal{P}_{\text{sm}}$ и $\mathcal{P} \setminus \mathcal{P}_{\text{sm}}$. Отличие от предыдущего случая в том, что множество $\mathcal{P}_{\text{sm}}$ не растёт монотонно, так как во время протокола ITBlockBC один и тот же участник может быть добавлен/удалён из множества $\mathcal{P}_{\text{sm}}$ несколько раз. При очередной итерации цикла происходит попытка перевести участника из $\mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ в $\mathcal{P}_{\text{sm}}$. Пусть $\{P_i, P_j\}$ — некоторая пара участников, для которых $P_i \in \mathcal{P}_{\text{sm}}$, $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$, $\{P_i, P_j\} \notin \Delta$. Участник P_i передаёт участнику P_j сообщение m_i. После этого участник P_j должен проверить, что полученное сообщение совпадает с сообщениями, которые имеются у участников множества $\mathcal{P}_{\text{sm}}$. Для этого P_j транслирует значение ключа k_j для ε-универсальной хеш-функции U_k, а дилер транслирует значение $h_j = U_{k_j}(m)$. Если участник P_j честно выбирает k_j случайным равновероятным образом, то с подавляющей вероятностью честные участники получат разные значения хеш-функции, если у них будут сообщения, отличающиеся от m. Если участник из $\mathcal{P}_{\text{sm}} \cup \{P_j\} \setminus \{D\}$ для своего сообщения получит значение хеш-функции, равное h_j, то он транслирует значение 1, иначе — 0. В данном случае не требуется, чтобы дилер D транслировал результат своей проверки, так как честный дилер всегда транслирует значение 1. Если все участники множества $\mathcal{P}_{\text{sm}} \cup \{P_j\} \setminus \{D\}$ транслируют 1, то участник P_j добавляется в множество $\mathcal{P}_{\text{sm}}$, в противном случае хотя бы один участник множества $\mathcal{P}_{\text{sm}} \cup \{P_j\}$ не обладает корректным сообщением, поэтому происходит поиск новых споров участников.

Важным отличием от криптографического случая является то, что споры могут возникать не только между P_i и P_j, но и между любыми двумя участниками в $\mathcal{P}_{\text{sm}}$. Чтобы найти такие споры, нужно знать историю формирования множества $\mathcal{P}_{\text{sm}}$. Для этого множество T содержит такие (упорядоченные) пары участников (P_i, P_j), для которых P_j получил от участника P_i сообщение m_i.

Протокол 10 (протокол ITBlockBC(m)).
Пусть дилер D обладает сообщением m.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

1) Каждый участник P_i инициализирует множества $\mathcal{P}_{\text{sm}} = \{D\}$ и $T = \varnothing$.
2) Цикл: пока существует хотя бы одна пара участников $P_i, P_j \in \mathcal{P}$ с условием $P_i \in \mathcal{P}_{\text{sm}}$, $P_j \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ и $\{P_i, P_j\} \notin \Delta$, делается следующее:
   а) участник P_i передаёт сообщение m_i участнику P_j. Обозначим полученное участником P_j сообщение через m_j. Пара (P_i, P_j) добавляется в T;
   б) участник P_j выбирает случайным равновероятным образом ключ $k_j \in K$ и транслирует его. После этого дилер D транслирует $h_j = U_{k_j}(m)$;
   в) каждый участник $P_k \in \mathcal{P}_{\text{sm}} \cup \{P_j\} \setminus \{D\}$ проверяет, выполнено ли равенство $U_{k_j}(m_k) = h_j$. Участник P_k транслирует 1, если равенство выполнено, в противном случае транслируется 0;
   г) если все участники множества $\mathcal{P}_{\text{sm}} \cup \{P_j\} \setminus \{D\}$ транслируют значение 1, то участник P_j добавляется в множество $\mathcal{P}_{\text{sm}}$, в противном случае делается следующее:

— для каждой пары вида $(P_k, P_r) \in T$, для которой P_k транслировал значение 1 (P_k может быть равен D), а участник P_r — значение 0, пара $\{P_k, P_r\}$ добавляется в множество Δ;
— $\mathcal{P}_{\text{sm}} := \{D\}$, $T := \varnothing$.

3) Каждый участник $P_i \in \mathcal{P}_{\text{sm}}$ определяет в качестве выходного значения сообщение m_i. Каждый участник $P_i \in \mathcal{P} \setminus \mathcal{P}_{\text{sm}}$ определяет в качестве выходного значения $\perp$.

Лемма 2 [14]. Пусть Δ — допустимое множество пар участников на момент вызова протокола ITBlockBC для сообщения (блока) m, Δ_{e} — соответствующее множество на момент окончания протокола ITBlockBC для сообщения m, $\mathcal{U}$ — универсальная функция хеширования с параметром безопасности κ. Тогда протокол ITBlockBC является t-безопасным протоколом широковещательной передачи, множество Δ_{e} является допустимым, число раундов протокола ITBlockBC равно $O(n + nd)$, коммуникационная сложность составляет $(n + nd)\big(|m| + 2\mathcal{B}(\kappa) + n\mathcal{B}(1)\big)$ бит, где $d = |\Delta_e| - |\Delta|$; $|m|$ — длина блока m.

Приведём протокол ITBC для широковещательной передачи сообщения длины l; он аналогичен протоколу CryptoBC.

Протокол 11 (протокол ITBC).
Пусть дилер D обладает сообщением $m \in \{0,1\}^l$.
Оракул: оракул для широковещательной передачи сообщений небольшой длины.

1) Участники инициализируют множество Δ в виде пустого множества.
2) Дилер D разбивает сообщение m на q блоков $m = (m_1, \ldots, m_q)$.
3) Для $i = 1, \ldots, q$ вызывается протокол ITBlockBC(m_i). Пусть m_j^i — выходное сообщение участника P_j после i-го шага, $i = 1, \ldots, q$.
4) Каждый участник P_j в качестве итогового выходного сообщения определяет следующее: если для некоторого i выполнено $m_j^i = \perp$, то $m_j = \perp$, в противном случае $m_j = (m_j^1, \ldots, m_j^q)$.

Ввиду леммы 2 коммуникационная сложность протокола ITBC не превышает

$$\sum_{i=1}^{q}(n + nd_i)\big(l/q + 2\mathcal{B}(\kappa) + n\mathcal{B}(1)\big) = n\Big(q + \sum_{i=1}^{q} d_i\Big)\big(l/q + 2\mathcal{B}(\kappa) + n\mathcal{B}(1)\big) \quad \text{бит.}$$

Так как $\sum\limits_{i=1}^{q} d_i \leqslant n^2$, то коммуникационная сложность ограничена сверху числом $n(q + n^2)\big(l/q + 2\mathcal{B}(\kappa) + n\mathcal{B}(1)\big)$. При $q = n^2$ коммуникационная сложность ограничена сверху числом $2ln + 2n^3\big(2\mathcal{B}(\kappa) + n\mathcal{B}(1)\big)$.

Так как число раундов протокола ITBlockBC при i-м вызове равно $O(n + nd_i)$, причём $\sum\limits_{i=1}^{q} d_i \leqslant n^2$, то число раундов протокола ITBC равно $O(n^3)$.

Теорема 8 [14]. При $t < n$ протокол ITBC при $q = n^2$ является (информационно-теоретически) t-безопасным протоколом широковещательной передачи для сообщения длины l бит с числом раундов $O(n^3)$ и коммуникационной сложностью $O\big(ln + (n^4 + n^3\kappa)\mathcal{B}(1)\big)$ бит, где κ — параметр безопасности.

Заключение

В таблице приведены параметры протоколов расширения для византийского соглашения и широковещательной передачи в синхронной настройке, где n — число участ-

ников; t — максимальное число нечестных участников; l — длина сообщения; k — параметр безопасности.

Порог	Безопасность	Протокол	**Коммуникационная сложность**	**Литература**
$t < n/3$	Информационно-теоретическая	Виз. соглашение, широковещание	$O\big(ln + n^2\mathcal{B}(1)\big)$	[10]
$t < n/3$	Информационно-теоретическая	Виз. соглашение, широковещание	$O\big(ln + n\mathcal{B}(1) + n^3\big)$	[15]
$t < n/2$	Информационно-теоретическая	Виз. соглашение, широковещание	$O\big(ln + n^3k + (n^2 + nk)\mathcal{B}(1)\big)$	[13]
$t < n/2$	Криптографическая	Виз. соглашение, широковещание	$O\big(ln + nk\mathcal{B}(1) + kn^3\big)$	[10]
$t < n/2$	Криптографическая	Виз. соглашение, широковещание	$O\big(ln + k\mathcal{A}(1) + kn^2\big)$	[15]
$t < (1-\varepsilon)n$	Криптографическая	Широковещание	$O\big(ln + k\mathcal{B}(1) + kn^2 + n^3\big)$	[15]
$t < n$	Криптографическая	Широковещание	$O\big(ln + (nk + n^3\log n)\mathcal{B}(1)\big)$	[10]
$t < n$	Криптографическая	Широковещание	$O\big(ln + (n^2 + nk)\mathcal{B}(1)\big)$	[14]
$t < n$	Информационно-теоретическая	Широковещание	$O\big(ln + (n^4 + n^3k)\mathcal{B}(1)\big)$	[14]

ЛИТЕРАТУРА

1. *Рацеев С. М.* Криптография. Безопасные многосторонние вычисления: учеб. пособие для вузов. 2-е изд., испр. и доп. СПб.: Лань, 2025. 540 с.
2. *Berman P., Garay J. A., and Perry K. J.* Bit-optimal distributed consensus // R. Baeza-Yates and U. Manber (eds.). Computer Science. Boston, MA: Springer, 1992. P. 313–322.
3. *Lamport L., Shostak R., and Pease M.* The Byzantine generals problem // ACM Trans. Program. Lang. Syst. 1982. V. 4. No. 3. P. 382–401.
4. *Pfitzmann B. and Waidner M.* Information-Theoretic Pseudosignatures and Byzantine Agreement for $t < n/3$. Technical Report RZ 2882. IBM Research, 1996.
5. *Kumaresan R.* Broadcast and Verifiable Secret Sharing: New Security Models and Round Optimal Constructions. PhD Thesis. University of Maryland at College Park, 2012.
6. *Dolev D. and Strong H. R.* Authenticated algorithms for Byzantine agreement // SIAM J. Computing. 1983. V. 12. No. 4. P. 656–666.
7. *Cleve R.* Limits on the security of coin flips when half the processors are faulty // Proc. STOC'86. Berkeley, California, USA, 1986. P. 364–369.
8. *Goldwasser S. and Lindell Y.* Secure computation without agreement // J. Cryptology. 2005. V. 18. No. 3. P. 247–287.
9. *Dolev D. and Reischuk R.* Bounds on information exchange for Byzantine agreement // Proc. PODS'82. Ottawa, Canada, 1982. P. 132–140.
10. *Ganesh C. and Patra A.* Optimal extension protocols for Byzantine broadcast and agreement // Distrib. Comput. 2021. V. 34. P. 59–77.
11. *Ben-Or M., Canetti R., and Goldreich O.* Asynchronous secure computation // Proc. STOC'93. N.Y., USA, 1993. P. 52–61.
12. *Рацеев С. М.* Элементы высшей алгебры и теории кодирования: учеб. пособие для вузов. 2-е изд., испр. и доп. СПб.: Лань, 2023. 684 с.
13. *Fitzi M. and Hirt M.* Optimally efficient multi-valued Byzantine agreement // Proc. PODC'06. Denver, Colorado, USA, 2006. P. 163–168.
14. *Hirt M. and Raykov P.* Multi-valued Byzantine broadcast: the $t < n$ case // LNCS. 2014. V. 8874. P. 448–465.
15. *Nayak K., Ren L., Shi E., et al.* Improved Extension Protocols for Byzantine Broadcast and Agreement. arXiv:2002.11321. `https://arxiv.org/abs/2002.11321`. 2020.

REFERENCES

1. *Ratseev S. M.* Kriptografiya. Bezopasnye mnogostoronnie vychisleniya [Cryptography. Secure Multiparty Computation]. St. Petersburg, Lan Publ., 2025. 540 p. (in Russian)
2. *Berman P., Garay J. A., and Perry K. J.* Bit-optimal distributed consensus. R. Baeza-Yates and U. Manber (eds.). Computer Science, Boston, MA, Springer, 1992, pp. 313–322.
3. *Lamport L., Shostak R., and Pease M.* The Byzantine generals problem. ACM Trans. Program. Lang. Syst., 1982, vol. 4, no. 3, pp. 382–401.
4. *Pfitzmann B. and Waidner M.* Information-Theoretic Pseudosignatures and Byzantine Agreement for $t < n/3$. Technical Report RZ 2882, IBM Research, 1996.
5. *Kumaresan R.* Broadcast and Verifiable Secret Sharing: New Security Models and Round Optimal Constructions. PhD Thesis, University of Maryland at College Park, 2012.
6. *Dolev D. and Strong H. R.* Authenticated algorithms for Byzantine agreement. SIAM J. Computing, 1983, vol. 12, no. 4, pp. 656–666.
7. *Cleve R.* Limits on the security of coin flips when half the processors are faulty. Proc. STOC'86, Berkeley, California, USA, 1986, pp. 364–369.
8. *Goldwasser S. and Lindell Y.* Secure computation without agreement. J. Cryptology, 2005, vol. 18, no. 3, pp. 247–287.
9. *Dolev D. and Reischuk R.* Bounds on information exchange for Byzantine agreement. Proc. PODS'82, Ottawa, Canada, 1982, pp. 132–140.
10. *Ganesh C. and Patra A.* Optimal extension protocols for Byzantine broadcast and agreement. Distrib. Comput., 2021, vol. 34, pp. 59–77.
11. *Ben-Or M., Canetti R., and Goldreich O.* Asynchronous secure computation. Proc. STOC'93, N.Y., USA, 1993, pp. 52–61.
12. *Ratseev S. M.* Elementy vysshey algebry i teorii kodirovaniya [Elements of Higher Algebra and Coding Theory]. St. Petersburg, Lan Publ., 2023. 684 p. (in Russian)
13. *Fitzi M. and Hirt M.* Optimally efficient multi-valued Byzantine agreement. Proc. PODC'06, Denver, Colorado, USA, 2006, pp. 163–168.
14. *Hirt M. and Raykov P.* Multi-valued Byzantine broadcast: the $t < n$ case. LNCS, 2014, vol. 8874, pp. 448–465.
15. *Nayak K., Ren L., Shi E., et al.* Improved Extension Protocols for Byzantine Broadcast and Agreement. arXiv:2002.11321. `https://arxiv.org/abs/2002.11321`, 2020.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЁЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

УДК 519.718.7 DOI 10.17223/20710410/71/4

КОРОТКИЕ ЕДИНИЧНЫЕ ПРОВЕРЯЮЩИЕ ТЕСТЫ РАЗМЫКАНИЯ ДЛЯ КОНТАКТНЫХ СХЕМ С ДВУМЯ И БОЛЕЕ ДОПОЛНИТЕЛЬНЫМИ ПОЛЮСАМИ

К. А. Попков

Институт прикладной математики им. М. В. Келдыша РАН, г. Москва, Россия

E-mail: kirill-formulist@mail.ru

Рассматривается задача синтеза многополюсных контактных схем, реализующих заданные булевы функции между полюсами A и B и допускающих короткие единичные проверяющие тесты относительно размыканий контактов. Для каждой булевой функции от n переменных и каждого тестового полюсного множества, содержащего хотя бы две отличных от $\{A, B\}$ и непересекающихся пары полюсов, найдено минимально возможное значение длины такого теста. В частности, доказано, что оно не превосходит 2.

Ключевые слова: *контактная схема, обрыв контакта, дополнительный полюс, единичный проверяющий тест, булева функция.*

SHORT SINGLE FAULT DETECTION TESTS OF CONTACT BREAK FOR CONTACT CIRCUITS WITH TWO OR MORE ADDITIONAL POLES

K. A. Popkov

Keldysh Institute of Applied Mathematics, Moscow, Russia

We consider the problem of synthesizing multi-pole contact circuits that implement given Boolean functions between poles A and B and allow short single fault detection tests related to contact breaks. For each Boolean function on n variables and each test pole set containing at least two disjoint pairs of poles other than $\{A, B\}$, the minimal possible length value of such a test is found. In particular, it is proved that this value does not exceed 2.

Keywords: *contact circuit, contact break, additional pole, single fault detection test, Boolean function.*

Введение

Рассматривается задача синтеза легкотестируемых контактных схем [1], реализующих заданные булевы функции. Логический подход к тестированию контактных схем предложен И. А. Чегис и С. В. Яблонским в [2]. Представим, что имеется двухполюсная контактная схема S, реализующая булеву функцию $f(\tilde{x}^n)$, где $\tilde{x}^n = (x_1, \ldots, x_n)$.

Под воздействием некоторого источника неисправностей один или несколько контактов схемы S могут перейти в неисправное состояние. В качестве неисправностей контактов обычно рассматриваются их обрывы (размыкания) и/или замыкания. При обрыве контакта проводимость между его концами становится тождественно нулевой, а при замыкании — тождественно единичной. В результате схема S вместо исходной функции $f(\tilde{x}^n)$ станет реализовывать некоторую булеву функцию $g(\tilde{x}^n)$, вообще говоря, отличную от f. Все такие функции $g(\tilde{x}^n)$, получающиеся при всевозможных допустимых для рассматриваемой задачи неисправностях контактов схемы S, называются *функциями неисправности* данной схемы.

Введём следующие определения [3, 4]. *Проверяющим тестом* для схемы S называется такое множество T наборов значений переменных $x_1, \ldots, x_n$, что для любой нетривиальной, т. е. отличной от $f(\tilde{x}^n)$, функции неисправности $g(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\sigma}$, на котором $f(\tilde{\sigma}) \neq g(\tilde{\sigma})$. *Диагностическим тестом* для схемы S называется такое множество T наборов значений булевых переменных $x_1, \ldots, x_n$, что T является проверяющим тестом и, кроме того, для любых двух различных функций неисправности $g_1(\tilde{x}^n)$ и $g_2(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\pi}$, на котором $g_1(\tilde{\pi}) \neq g_2(\tilde{\pi})$. Число наборов в T называется *длиной* теста. В качестве тривиального диагностического (и проверяющего) теста длины 2^n для схемы S всегда можно взять множество, состоящее из всех двоичных наборов длины n. Тест называется *полным*, если в схеме могут быть неисправны сколько угодно контактов, и *единичным*, если в схеме может быть неисправен только один контакт. Единичные тесты обычно рассматривают для *неизбыточных схем* [4, с. 110–111], в которых любая допустимая неисправность любого одного контакта приводит к нетривиальной функции неисправности. Если в схеме допускаются только обрывы контактов (или только их замыкания), то говорят о *тестах размыкания* (соответственно о *тестах замыкания*).

Пусть зафиксирован вид неисправностей контактов и T — единичный проверяющий тест (ЕПТ) для некоторой двухполюсной контактной схемы S. Введём следующие обозначения: $D_{\text{ЕП}}(T)$ — длина теста T; $D_{\text{ЕП}}(S) = \min D_{\text{ЕП}}(T)$, где минимум берётся по всем ЕПТ T для схемы S; $D_{\text{ЕП}}(f) = \min D_{\text{ЕП}}(S)$, где минимум берётся по всем неизбыточным двухполюсным контактным схемам S, реализующим функцию f; $D_{\text{ЕП}}(n) = \max D_{\text{ЕП}}(f)$, где максимум берётся по всем булевым функциям f от n переменных. Функция $D_{\text{ЕП}}(n)$ называется *функцией Шеннона* длины ЕПТ. По аналогии с функциями $D_{\text{ЕП}}$ можно ввести функции $D_{\text{ПП}}$, $D_{\text{ЕД}}$ и $D_{\text{ПД}}$ для соответственно полного проверяющего (ПП), единичного диагностического (ЕД) и полного диагностического (ПД) тестов, зависящие от T, от S, от f и от n (в определениях функций $D_{\text{ПП}}(f)$ и $D_{\text{ПД}}(f)$ не предполагается неизбыточности схем). Так, например, $D_{\text{ПД}}(n)$ — функция Шеннона длины полного диагностического теста.

Будем говорить, что некоторое свойство выполняется *для почти всех булевых функций от n переменных*, если отношение числа булевых функций от n переменных, для которых это свойство не выполняется, к числу всех булевых функций от n переменных (т. е. к 2^{2^n}) стремится к нулю при $n \to \infty$.

Далее в качестве неисправностей контактов будем рассматривать только их обрывы. Перечислим основные результаты, касающиеся тестов размыкания для двухполюсных контактных схем. Н. П. Редькиным в [5] получена оценка $D_{\text{ПП}}(n) \leqslant 2^{\lfloor n/2 \rfloor} + 2^{\lceil n/2 \rceil}$. В [6] найдено точное значение величины $D_{\text{ЕП}}(f)$ для любой булевой функции f от n переменных и установлено, что $D_{\text{ЕП}}(n) = n$, а для почти всех булевых функций f от n переменных $D_{\text{ЕП}}(f) = 2$. В силу [6, утверждение 1] вышеупомянутые результаты для величин $D_{\text{ЕП}}(f)$ и $D_{\text{ЕП}}(n)$ остаются справедливыми для $D_{\text{ПП}}(f)$ и $D_{\text{ПП}}(n)$

соответственно; в частности, равенство $D_{\text{ЕП}}(n) = n$ из [6] уточняет неравенство $D_{\text{ПП}}(n) \leqslant 2^{\lfloor n/2 \rfloor} + 2^{\lceil n/2 \rceil}$ из [5]. По аналогии с [4, с. 113, теорема 9] можно показать, что $D_{\text{ЕД}}(n) \lesssim 2^n/n$; в [7] при $n \geqslant 2$ получена существенно более точная оценка $D_{\text{ЕД}}(n) \leqslant 2n - 2$, а также доказано, что $D_{\text{ЕД}}(f) \leqslant 4$ для почти всех булевых функций f от n переменных. Х. А. Мадатян в [8, теорема 1] фактически установил, что $D_{\text{ПД}}(n) \geqslant 2^{n-1}$ при $n \geqslant 1$; Н. П. Редькин в [9] доказал соотношение $D_{\text{ПД}}(n) \leqslant 2^n - 2$ при $n \geqslant 2$. В [10] для любого $n \geqslant 1$ установлено равенство $D_{\text{ПД}}(n) = 2^{n-1}$ и доказано, что число булевых функций f от n переменных, для которых $D_{\text{ПД}}(f) = 2^{n-1}$, асимптотически не меньше $4n^{-2} \cdot 2^{2^n(\log_2(n^2-n+2)-1)/(n^2-n+2)}$.

В настоящей работе будем исследовать возможности реализации булевых функций контактными схемами, содержащими дополнительные полюсы и допускающими короткие проверяющие тесты размыкания, т. е. многополюсными контактными схемами, в которых между какими-то двумя полюсами A и B реализуется заданная булева функция и при этом для тестирования схем относительно обрывов контактов используются заранее выбранные пары полюсов. Опишем формальную постановку задачи. Рассмотрим произвольное $d \in \mathbb{N}$ и всевозможные $(d+2)$-полюсные контактные схемы с полюсами $A, B, V_1, \ldots, V_d$; при этом из рассмотрения не исключаем и такие схемы, в которых некоторые из полюсов $A, B, V_1, \ldots, V_d$ совпадают. Пусть зафиксировано некоторое непустое множество $\mathcal{P}$ неупорядоченных пар полюсов из множества $\{A, B, V_1, \ldots, V_d\}$, обладающее тем свойством, что никакая пара из $\mathcal{P}$ не может состоять из двух полюсов с одним и тем же обозначением (например, $\{A, A\}$). Назовём $\mathcal{P}$ *тестовым полюсным множеством* и занумеруем произвольным образом его элементы различными натуральными числами от 1 до p, где $p = |\mathcal{P}|$; очевидно, что $1 \leqslant p \leqslant (d+2)(d+1)/2$. Пусть некоторая $(d+2)$-полюсная контактная схема S с полюсами $A, B, V_1, \ldots, V_d$ для каждого $i \in \{1, \ldots, p\}$ реализует между i-й парой своих полюсов булеву функцию $f_i(\tilde{x}^n)$. При наличии в схеме S оборванных контактов она для каждого $i \in \{1, \ldots, p\}$ будет реализовывать между i-й парой своих полюсов некоторую булеву функцию $g_i(\tilde{x}^n)$, вообще говоря, отличную от $f_i(\tilde{x}^n)$. Все такие наборы $(g_1(\tilde{x}^n), \ldots, g_p(\tilde{x}^n))$ назовём *наборами функций неисправности* схемы S. *Проверяющим тестом* для схемы S назовём такое множество T наборов значений булевых переменных $x_1, \ldots, x_n$, что для любого нетривиального, т. е. отличного от $(f_1(\tilde{x}^n), \ldots, f_p(\tilde{x}^n))$ набора функций неисправности $(g_1(\tilde{x}^n), \ldots, g_p(\tilde{x}^n))$ схемы S в T найдётся набор $\tilde{\sigma}$, на котором

$$(f_1(\tilde{\sigma}), \ldots, f_p(\tilde{\sigma})) \neq (g_1(\tilde{\sigma}), \ldots, g_p(\tilde{\sigma})).$$

Диагностическим тестом для схемы S назовём такое множество T наборов значений переменных $x_1, \ldots, x_n$, что T является проверяющим тестом и, кроме того, для любых двух различных наборов функций неисправности $(g_{11}(\tilde{x}^n), \ldots, g_{1p}(\tilde{x}^n))$ и $(g_{21}(\tilde{x}^n), \ldots, g_{2p}(\tilde{x}^n))$ схемы S в T найдётся набор $\tilde{\pi}$, на котором

$$(g_{11}(\tilde{\pi}), \ldots, g_{1p}(\tilde{\pi})) \neq (g_{21}(\tilde{\pi}), \ldots, g_{2p}(\tilde{\pi})).$$

Определения полного и единичного тестов и длины теста остаются неизменными. Будем рассматривать единичные тесты только для *неизбыточных схем*, в которых обрыв любого одного контакта приводит к нетривиальному набору функций неисправности.

Легко видеть, что определения проверяющего и диагностического тестов не зависят от того, в каком порядке элементы множества $\mathcal{P}$ занумерованы числами от 1 до p.

Пусть T — ЕПТ размыкания для некоторой $(d+2)$-полюсной контактной схемы S с полюсами $A, B, V_1, \ldots, V_d$ и заданным тестовым полюсным множеством $\mathcal{P}$. Введём

следующие обозначения: $D^{d,\mathcal{P}}_{\text{ЕП}}(T)$ — длина теста T; $D^{d,\mathcal{P}}_{\text{ЕП}}(S) = \min D^{d,\mathcal{P}}_{\text{ЕП}}(T)$, где минимум берётся по всем ЕПТ T для схемы S; $D^{d,\mathcal{P}}_{\text{ЕП}}(f) = \min D^{d,\mathcal{P}}_{\text{ЕП}}(S)$, где минимум берётся по всем неизбыточным $(d+2)$-полюсным контактным схемам S с полюсами $A, B, V_1, \ldots, V_d$, реализующим между полюсами A и B функцию f; $D^{d,\mathcal{P}}_{\text{ЕП}}(n) = \max D^{d,\mathcal{P}}_{\text{ЕП}}(f)$, где максимум берётся по всем булевым функциям f от n переменных. Функцию $D^{d,\mathcal{P}}_{\text{ЕП}}(n)$ назовём *функцией Шеннона* длины ЕПТ. По аналогии с функциями $D^{d,\mathcal{P}}_{\text{ЕП}}$ можно ввести функции $D^{d,\mathcal{P}}_{\text{ПП}}$, $D^{d,\mathcal{P}}_{\text{ЕД}}$ и $D^{d,\mathcal{P}}_{\text{ПД}}$ для соответственно полного проверяющего, единичного диагностического и полного диагностического тестов, зависящие от T, от S, от f и от n (в определениях функций $D^{d,\mathcal{P}}_{\text{ПП}}(f)$ и $D^{d,\mathcal{P}}_{\text{ПД}}(f)$ не предполагается неизбыточности схем).

Утверждение 1. Для любой булевой функции $f(\tilde{x}^n)$, любого $d \in \mathbb{N}$ и любого тестового полюсного множества $\mathcal{P}$ значение $D^{d,\mathcal{P}}_{\text{ЕП}}(f)$ определено.

Доказательство. Как известно (см., например, [6, теорема 1]), значение $D_{\text{ЕП}}(f)$ определено, поэтому существует неизбыточная двухполюсная контактная схема S' с полюсами A и B, реализующая функцию $f(\tilde{x}^n)$. Преобразуем схему S' в $(d+2)$-полюсную контактную схему S с полюсами $A, B, V_1, \ldots, V_d$ путём добавления полюсов $V_1, \ldots, V_d$. Пусть $\{P_1, P_2\}$ — произвольная пара полюсов из множества $\mathcal{P}$. Каждый полюс схемы S, принадлежащий множеству $\{V_1, \ldots, V_d\} \setminus \{P_1, P_2\}$, отождествим с полюсом A. В случае $P_1 = B$ будем считать, что полюс P_2 схемы S совпадает с полюсом A, в случае $P_2 = A$ — что полюс P_1 схемы S совпадает с полюсом B, а в случае $P_1 \neq B$ и $P_2 \neq A$ — что полюсы P_1 и P_2 схемы S совпадают с полюсами A и B соответственно. Тогда пара $\{P_1, P_2\}$ совпадает с парой $\{A, B\}$, а схема S, очевидно, реализует между полюсами A и B функцию $f(\tilde{x}^n)$. Пусть K — произвольный контакт схемы S. Тогда он содержится и в схеме S', и при его обрыве схема S' в силу неизбыточности станет реализовывать некоторую функцию неисправности $g(\tilde{x}^n)$, отличную от $f(\tilde{x}^n)$. Из соотношений $\{A, B\} \in \mathcal{P}$, $f(\tilde{x}^n) \neq g(\tilde{x}^n)$, определения неизбыточной $(d+2)$-полюсной контактной схемы и произвольности контакта K легко получить, что при выборе $\mathcal{P}$ в качестве тестового полюсного множества схема S неизбыточна, откуда следует, что значение $D^{d,\mathcal{P}}_{\text{ЕП}}(f)$ определено. ■

В работе [11] найдено точное значение величины $D^{1,\mathcal{P}}_{\text{ЕП}}(f)$ для каждой булевой функции $f(\tilde{x}^n)$ и каждого тестового полюсного множества $\mathcal{P}$, а также установлено равенство $D^{1,\mathcal{P}}_{\text{ЕП}}(n) = \min(3, n)$ при $\mathcal{P} \neq \{\{A, B\}\}$. В настоящей работе для любой булевой функции $f(\tilde{x}^n)$, любого целого $d \geqslant 2$ и любого тестового полюсного множества $\mathcal{P}$, содержащего хотя бы две отличных от $\{A, B\}$ и непересекающихся пары полюсов, устанавливается точное значение величины $D^{d,\mathcal{P}}_{\text{ЕП}}(f)$ (теорема 1); в частности, доказывается, что оно равно 0, 1 или 2 в зависимости от функции f. В качестве следствия из теоремы 1 при тех же условиях на d и $\mathcal{P}$ для произвольного целого $n \geqslant 0$ получено равенство $D^{d,\mathcal{P}}_{\text{ЕП}}(n) = \min(2, n)$ (следствие 1). Сравнивая последний результат с равенством $D_{\text{ЕП}}(n) = n$ из [6], получаем, что добавление в контактные схемы по крайней мере двух дополнительных полюсов позволяет в ряде случаев уменьшить функцию Шеннона длины ЕПТ размыкания с n до 2.

Формулировка и доказательство основного результата

Двоичный n-разрядный набор $\tilde{\sigma}$ будем называть *единичным* (*нулевым*) набором булевой функции $f(\tilde{x}^n)$, если $f(\tilde{\sigma}) = 1$ (соответственно $f(\tilde{\sigma}) = 0$). Для каждого $n \in \mathbb{N}$ обозначим через K_n множество всех элементарных конъюнкций вида $x_{i_1}^{\sigma_1} \& \ldots \& x_{i_s}^{\sigma_s}$, где $s \in \{1, \ldots, n\}$; $1 \leqslant i_1 < \ldots < i_s \leqslant n$; $\sigma_1, \ldots, \sigma_s \in \{0, 1\}$.

Всюду далее запись вида «контакт x^σ» означает замыкающий (размыкающий) контакт, отвечающий переменной x, в случае $\sigma = 1$ (соответственно $\sigma = 0$). Под цепью (в контактной схеме) будем понимать несамопересекающуюся цепь. *Длиной цепи* будем называть число содержащихся в этой цепи контактов.

Сформулируем основной результат данной работы.

Теорема 1. Для любой булевой функции $f(\tilde{x}^n)$, любого целого $d \geqslant 2$ и любого тестового полюсного множества $\mathcal{P}$, содержащего хотя бы две отличных от $\{A, B\}$ и непересекающихся пары полюсов, справедливо следующее равенство:

$$D_{\text{ЕП}}^{d,\mathcal{P}}(f) = \begin{cases} 0, & \text{если } f \equiv 0 \text{ или } f \equiv 1, \\ 1, & \text{если } n \geqslant 1 \text{ и } f \in \mathsf{K}_n, \\ 2 & \text{в остальных случаях.} \end{cases}$$

Следствие 1. Для любых целых $n \geqslant 0$, $d \geqslant 2$ и любого тестового полюсного множества $\mathcal{P}$, содержащего хотя бы две отличных от $\{A, B\}$ и непересекающихся пары полюсов, справедливо равенство $D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \min(2, n)$.

Сначала покажем, как следствие 1 выводится из теоремы 1. Имеем

$$D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \max_{f \in \{0,1\}} D_{\text{ЕП}}^{d,\mathcal{P}}(f) = \max(0,0) = 0 = \min(2,n) \text{ при } n = 0,$$

$$D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \max_{f \in \{0,1,x_1,\overline{x}_1\}} D_{\text{ЕП}}^{d,\mathcal{P}}(f) = \max(0,0,1,1) = 1 = \min(2,n) \text{ при } n = 1,$$

поскольку $x_1, \overline{x}_1 \in \mathsf{K}_n$;

$$D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \max_{f(\tilde{x}^n)} D_{\text{ЕП}}^{d,\mathcal{P}}(f) \leqslant 2 = \min(2,n) \text{ при } n \geqslant 2,$$

$$D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \max_{f(\tilde{x}^n)} D_{\text{ЕП}}^{d,\mathcal{P}}(f) \geqslant D_{\text{ЕП}}^{d,\mathcal{P}}(x_1 \vee \ldots \vee x_n) = 2 = \min(2,n) \text{ при } n \geqslant 2,$$

поскольку $x_1 \vee \ldots \vee x_n \notin \mathsf{K}_n$ при $n \geqslant 2$. Таким образом, $D_{\text{ЕП}}^{d,\mathcal{P}}(n) = \min(2, n)$.

Доказательство теоремы 1. В случае $f \equiv 0$ или $f \equiv 1$ функцию $f(\tilde{x}^n)$ можно реализовать между полюсами A и B контактной схемы с полюсами $A, B, V_1, \ldots, V_d$, не содержащей ни одного контакта. У такой схемы нет ни одного набора функций неисправности, поэтому она неизбыточна и допускает ЕПТ $\varnothing$ длины 0, откуда следует равенство $D_{\text{ЕП}}^{d,\mathcal{P}}(f) = 0$.

Далее будем считать, что $f \not\equiv 0$ и $f \not\equiv 1$; в частности, $n \geqslant 1$. В любой неизбыточной контактной схеме S_f с полюсами $A, B, V_1, \ldots, V_d$, реализующей между полюсами A и B функцию $f(\tilde{x}^n)$, обязан содержаться хотя бы один контакт. При его обрыве схема в силу неизбыточности станет реализовывать между парами полюсов из множества $\mathcal{P}$ нетривиальный набор функций неисправности, который должен быть отличим от исходного набора функций, реализуемых схемой S_f между этими парами полюсов, хотя бы на одном наборе из произвольного ЕПТ T для данной схемы, откуда следуют неравенства $D_{\text{ЕП}}^{d,\mathcal{P}}(T) \geqslant 1$, $D_{\text{ЕП}}^{d,\mathcal{P}}(S_f) \geqslant 1$ и $D_{\text{ЕП}}^{d,\mathcal{P}}(f) \geqslant 1$.

Пусть $f \in \mathsf{K}_n$. Тогда $f(\tilde{x}^n) = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_s}^{\sigma_s}$, где $s \in \{1, \ldots, n\}$; $1 \leqslant i_1 < \ldots < i_s \leqslant n$; $\sigma_1, \ldots, \sigma_s \in \{0, 1\}$. Реализуем функцию $f(\tilde{x}^n)$ между полюсами A и B контактной схемой S с полюсами $A, B, V_1, \ldots, V_d$, представляющей собой цепь из s контактов $x_{i_1}^{\sigma_1}, \ldots, x_{i_s}^{\sigma_s}$, концами которой являются вершины A и B. Пусть $\{P_1, P_2\}$ — произвольная пара полюсов из множества $\mathcal{P}$. С использованием метода, изложенного в доказательстве утверждения 1, можно добиться того, чтобы пара $\{P_1, P_2\}$ совпала с парой $\{A, B\}$.

Тогда в случае возникновения в схеме S обрыва какого-то контакта проводимость между полюсами P_1 и P_2 схемы на любом единичном наборе $\tilde{\sigma}$ функции $f(\tilde{x}^n)$ изменится с 1 на 0 и неисправность будет обнаружена. Тем самым показано, что схема S неизбыточна и допускает ЕПТ $\{\tilde{\sigma}\}$ длины 1, откуда следует неравенство $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \leqslant 1$. Равенство $D^{d,\mathcal{P}}_{\text{ЕП}}(f) = 1$ доказано.

Далее рассмотрим случай $f \notin \mathsf{K}_n$. Тогда, в частности, $n \geqslant 2$, поскольку $x_1, \overline{x}_1 \in \mathsf{K}_n$. Сначала установим справедливость неравенства $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \geqslant 2$. Воспользуемся идеями, изложенными в доказательстве теоремы 2 работы [11] при получении оценки $D^{\mathcal{P}_2}_{\text{ЕП}}(f) \geqslant 2$ [11, с. 132–133]. Предположим, что $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \leqslant 1$. Тогда $D^{d,\mathcal{P}}_{\text{ЕП}}(f) = 1$. Это означает, что при выборе в качестве тестового полюсного множества $\mathcal{P}$ существует неизбыточная контактная схема S с полюсами $A, B, V_1, \ldots, V_d$, реализующая между полюсами A и B функцию $f(\tilde{x}^n)$ и допускающая ЕПТ, который состоит из какого-то одного набора $\tilde{\sigma}$. Полюсы A и B различны, так как $f \not\equiv 1$. Если при отсутствии неисправностей в схеме S хотя бы один содержащийся в ней контакт K_0 не проводит на наборе $\tilde{\sigma}$, то обрыв контакта K_0 никак не отразится на значениях, возникающих между полюсами схемы на указанном наборе; следовательно, данный обрыв нельзя обнаружить на наборе $\tilde{\sigma}$. Однако это противоречит тому, что $\{\tilde{\sigma}\}$ — ЕПТ для неизбыточной схемы S. Таким образом, при отсутствии неисправностей в схеме каждый её контакт обязан проводить на наборе $\tilde{\sigma}$.

Если между полюсами A и B схемы S нет ни одной цепи (есть ровно одна цепь), то $f \equiv 0$ (соответственно $f \in \mathsf{K}_n \cup \{0\}$), что невозможно. Значит, между полюсами A и B схемы S есть по крайней мере две различных цепи C и C'. Очевидно, что в цепи C' содержится хотя бы один контакт K, не содержащийся в цепи C. Обозначим тот конец контакта K, который расположен при движении по цепи C' от вершины A к вершине B ближе к вершине A (вершине B), через a (соответственно b). Подцепь произвольной цепи $\hat{C}$ в схеме S, ограниченную какими-то вершинами v и v', для краткости будем обозначать через $\hat{C}_{v,v'}$.

В рамках этого абзаца считаем, что вместо набора $(x_1, \ldots, x_n)$ переменных схемы S подан набор $\tilde{\sigma}$. При обрыве контакта K в схеме S есть проводимость между вершинами a и A (по цепи $C'_{a,A}$), A и B (по цепи C), а также B и b (по цепи $C'_{B,b}$); следовательно, при указанном обрыве в ней есть проводимость и между вершинами a и b. Обрыв контакта K должен обнаруживаться между какой-то парой полюсов $\{P_1, P_2\} \in \mathcal{P}$, поскольку $\{\tilde{\sigma}\}$ — ЕПТ для неизбыточной схемы S. При рассматриваемой неисправности проводимость между полюсами P_1 и P_2 схемы не может увеличиться, поэтому она обязана измениться с 1 на 0. Отсюда, в частности, следует, что в схеме S при отсутствии в ней неисправностей существует проводящая цепь C'' между вершинами P_1 и P_2. Если контакт K не содержится в цепи C'', то при его обрыве проводимость между полюсами P_1 и P_2 схемы остаётся равной 1, что невозможно. Значит, данная цепь проходит через контакт K и, как следствие, через его концы a и b. Найдётся такое $i \in \{1, 2\}$, что вершина a расположена в цепи C'' между вершинами P_i и b (возможно, при этом вершины a и P_i совпадают). Тогда при обрыве контакта K в схеме S есть проводимость между вершинами P_i и a (по цепи $C''_{P_i,a}$), a и b (см. выше), а также b и P_{3-i} (по цепи $C''_{b,P_{3-i}}$). Следовательно, при указанном обрыве проводимость между полюсами P_1 и P_2 схемы остаётся равной 1; противоречие. Тем самым установлено, что исходное предположение было неверно и $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \geqslant 2$.

Наконец, докажем неравенство $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \leqslant 2$. Для этого построим неизбыточную контактную схему S с полюсами $A, B, V_1, \ldots, V_d$, реализующую между полюсами A и B функцию $f(\tilde{x}^n)$ и допускающую ЕПТ из двух наборов при выборе $\mathcal{P}$ в качестве

тестового полюсного множества. По условию теоремы 1 в множестве $\mathcal{P}$ содержатся две отличных от $\{A, B\}$ и непересекающихся пары полюсов; обозначим эти пары через U_1 и U_2. При построении схемы S переименуем и/или отождествим некоторые её полюсы в зависимости от того, какой из следующих случаев имеет место:

С л у ч а й 1: полюс A принадлежит паре U_i для некоторого $i \in \{1, 2\}$. Второй полюс из этой пары обозначим через A'. Отметим, что $B \notin U_i$, так как $U_i \neq \{A, B\}$, и $A \notin U_{3-i}$, поскольку пары U_1 и U_2 не пересекаются. Рассмотрим два подслучая:

П о д с л у ч а й 1.1: полюс B принадлежит паре U_{3-i}. Второй полюс из этой пары обозначим через B'.

П о д с л у ч а й 1.2: полюс B не принадлежит паре U_{3-i}. Отождествим произвольный полюс из этой пары с полюсом B, а второй полюс из неё обозначим через B'.

С л у ч а й 2: полюс A не принадлежит ни одной из пар U_1, U_2. Рассмотрим два подслучая:

П о д с л у ч а й 2.1: полюс B принадлежит паре U_i для некоторого $i \in \{1, 2\}$. Второй полюс из этой пары обозначим через B'. Отметим, что $A \notin U_i$ и $A \notin U_{3-i}$ по предположению случая 2; кроме того, $B \notin U_{3-i}$, поскольку пары U_1 и U_2 не пересекаются. Отождествим произвольный полюс из пары U_{3-i} с полюсом A, а второй полюс из этой пары обозначим через A'.

П о д с л у ч а й 2.2: полюс B не принадлежит ни одной из пар U_1, U_2. Отождествим произвольный полюс из пары U_1 с полюсом A, а второй полюс из этой пары обозначим через A'. Также отождествим произвольный полюс из пары U_2 с полюсом B, а второй полюс из неё обозначим через B'.

В результате указанных переименований и/или отождествлений получаем, что одна из пар U_1, U_2 теперь является парой $\{A, A'\}$, а другая — парой $\{B, B'\}$, где полюсы A, A', B, B' попарно различны и $A', B' \in \{V_1, \ldots, V_d\}$.

Пусть $\sigma_1, \ldots, \sigma_n$ — произвольные булевы константы. Положим $\tilde{\sigma}^{(1)} = (\sigma_1, \ldots, \sigma_n)$ и $\tilde{\sigma}^{(2)} = (\overline{\sigma}_1, \overline{\sigma}_2, \ldots, \overline{\sigma}_n)$.

Дальнейшее построение схемы S будем проводить по аналогии с тем, как это делается в доказательстве [11, теорема 2, с. 125–130]. Возьмём контактное дерево $\hat{D}$, реализующее систему всех 2^n элементарных конъюнкций вида $x_1^{\beta_1} \& \ldots \& x_n^{\beta_n}$, где $\beta_1, \ldots, \beta_n \in \in \{0, 1\}$, и содержащее $2 \cdot 2^n - 2$ контактов [1, с. 39], в котором корню a_0 инцидентны контакты x_1 и $\overline{x}_1$, далее идут контакты переменной x_2 и т. д. Для каждого нулевого набора $(\tau_1, \ldots, \tau_n)$ функции $f(\tilde{x}^n)$ удалим из дерева $\hat{D}$ концевую вершину вместе с инцидентным ей ребром, в которой реализуется элементарная конъюнкция $x_1^{\tau_1} \& \ldots \& x_n^{\tau_n}$. Если после всех этих операций в дереве возникли новые концевые вершины, удалим и их вместе с инцидентными им рёбрами, и т. д. Легко проверить, что полученное дерево D обладает следующими свойствами:

(i) любая вершина дерева инцидентна не более чем трём контактам, и если трём, то два из них противоположны (а именно: x_t и $\overline{x}_t$ для некоторого $t \in \{2, \ldots, n\}$);

(ii) корень дерева инцидентен не более чем двум контактам, и если двум, то это противоположные контакты (а именно: x_1 и $\overline{x}_1$).

(iii) в каждой концевой вершине дерева реализуется элементарная конъюнкция вида $x_1^{\beta_1} \& \ldots \& x_n^{\beta_n}$, причём единственный двоичный n-разрядный набор, на котором эта конъюнкция обращается в единицу, является единичным для функции $f(\tilde{x}^n)$;

(iv) для каждого единичного набора $(\beta_1, \ldots, \beta_n)$ функции $f(\tilde{x}^n)$ в дереве найдётся такая концевая вершина, что единственная цепь, связывающая корень дерева с этой вершиной, содержит n контактов: $x_1^{\beta_1}, \ldots, x_n^{\beta_n}$;

(v) в дереве нет ни одной цепи, соединяющей какие-либо две различные концевые вершины и проводящей хотя бы на одном двоичном n-разрядном наборе.

Например, для функции $f(x_1, x_2, x_3, x_4)$, принимающей значение 1 только на наборах $(0,1,0,0), (0,1,1,1), (1,0,0,1), (1,1,1,0), (1,1,1,1)$, дерево D имеет вид, показанный на рис. 1.

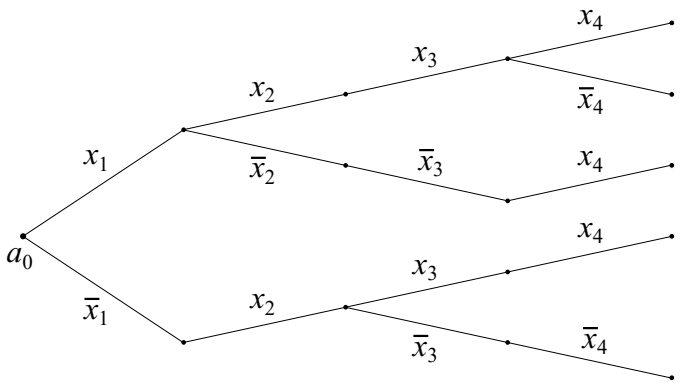

Рис. 1. Дерево D

Для $i = 1, 2$ обозначим через $D_{\tilde{\sigma}^{(i)}}$ граф, состоящий из всех вершин и всех проводящих на наборе $\tilde{\sigma}^{(i)}$ контактов дерева D.

Лемма 1. Для любого $i \in \{1, 2\}$ каждая компонента связности графа $D_{\tilde{\sigma}^{(i)}}$ представляет собой цепь (возможно, нулевой длины), в которой ни одна внутренняя вершина не совпадает с вершиной a_0 и хотя бы одна концевая вершина отлична от всех концевых вершин дерева D.

Доказательство почти дословно повторяет доказательство [11, лемма 1]. Дерево D, а вместе с ним и граф $D_{\tilde{\sigma}^{(i)}}$ не содержат циклов. Из свойств (i), (ii) и того, что противоположные контакты не могут оба проводить на наборе $\tilde{\sigma}^{(i)}$, следует, что в графе $D_{\tilde{\sigma}^{(i)}}$ любая вершина инцидентна не более чем двум контактам, а вершина a_0 — не более чем одному контакту. Отсюда и из свойства (v), применяемого для набора $\tilde{\sigma}^{(i)}$, вытекает утверждение леммы 1. ■

Возьмём две копии дерева D и соединим каждую концевую вершину одной из них с той же концевой вершиной другой. Корни этих копий будем считать полюсами A и B полученной «симметричной» двухполюсной контактной схемы, которую обозначим через S'. Из свойств (iii), (iv) нетрудно получить, что схема S' реализует функцию $f(\tilde{x}^n)$ (каждая цепь между полюсами схемы S', проводящая хотя бы на одном наборе, представляет собой «удвоенную» цепь из дерева D, соединяющую его корень с какой-то концевой вершиной), а из леммы 1 — что для любого $i \in \{1, 2\}$ в подсхеме $S'_{\tilde{\sigma}^{(i)}}$, состоящей из всех вершин и всех проводящих на наборе $\tilde{\sigma}^{(i)}$ контактов схемы S', каждая компонента связности представляет собой цепь, в которой ни одна внутренняя вершина не совпадает ни с одним из полюсов схемы S'.

Схема S' и её подсхемы $S'_{\tilde{\sigma}^{(1)}}, S'_{\tilde{\sigma}^{(2)}}$ для функции $f(x_1, x_2, x_3, x_4)$, заданной выше, и наборов $\tilde{\sigma}^{(1)} = (1,1,1,1)$, $\tilde{\sigma}^{(2)} = (0,0,0,0)$ приведены на рис. 2.

Преобразуем схему S' в $(d+2)$-полюсную контактную схему S с полюсами A, B, V_1, …, V_d. Рассмотрим произвольное $i \in \{1, 2\}$. Пусть в подсхеме $S'_{\tilde{\sigma}^{(i)}}$ имеется ровно k_i таких компонент связности, каждая из которых, за исключением, быть может, компоненты связности, содержащей полюс A схемы S', содержит хотя бы один контакт

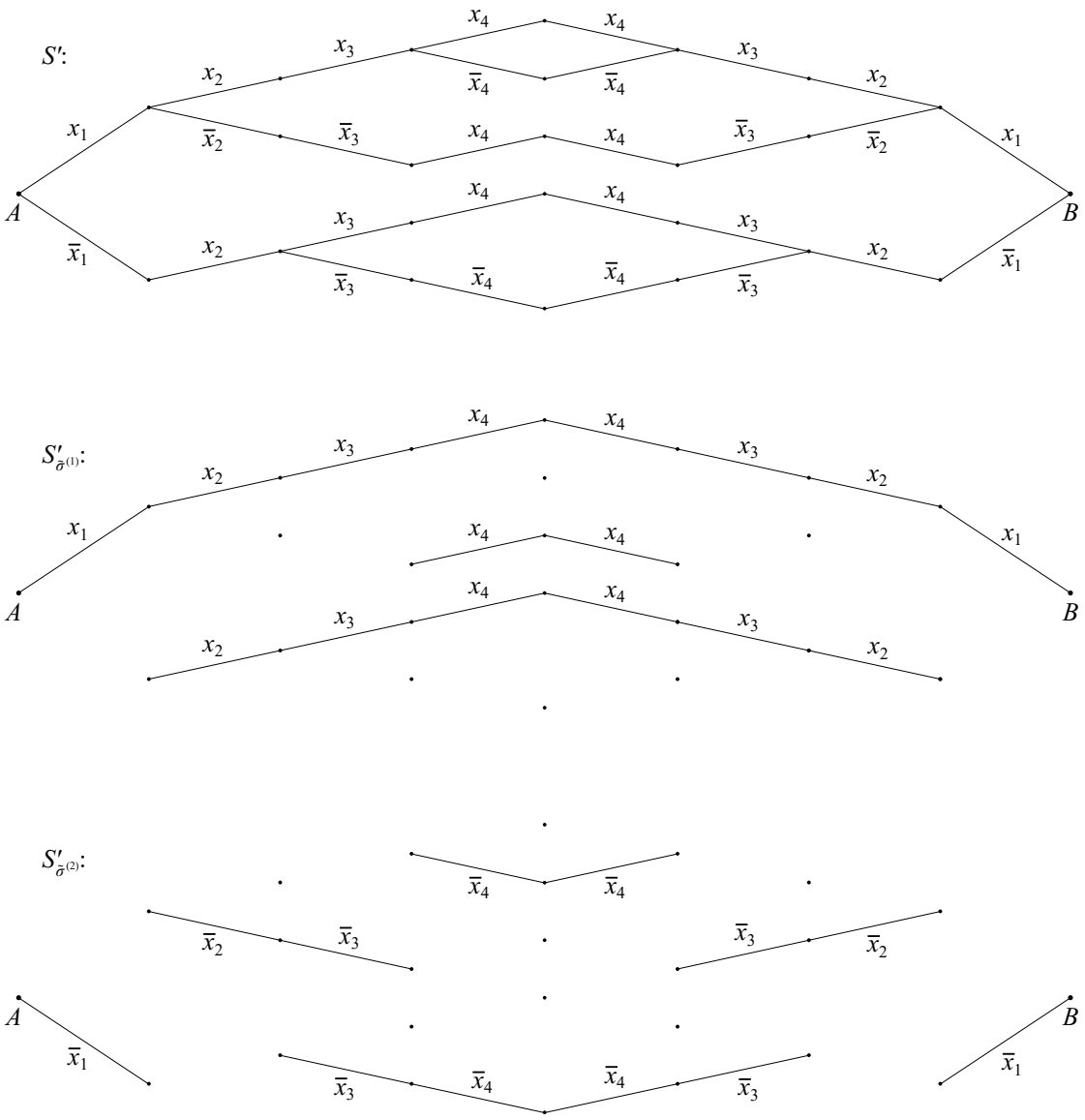

Рис. 2. Схема S' и подсхемы $S'_{\tilde\sigma(1)}, S'_{\tilde\sigma(2)}$

и не содержит вершины B. Обозначим эти k_i компонент связности через $C^i_1, \ldots, C^i_{k_i}$, где C^i_1 — та из них, в которую входит вершина A. Тогда C^i_1 — цепь, и A обязана быть одной из концевых вершин данной цепи. Вторую её концевую вершину обозначим через b^i_1 (вершина b^i_1 может совпадать с одним из полюсов A, B схемы S'). В случае $k_i \geqslant 2$ обозначим концевые вершины цепей $C^i_2, \ldots, C^i_{k_i}$ через a^i_2 и b^i_2, …, $a^i_{k_i}$ и $b^i_{k_i}$ соответственно. Компоненту связности подсхемы $S'_{\tilde\sigma^{(i)}}$, содержащую полюс B схемы S', обозначим через C^i_B. Тогда C^i_B — цепь, и B обязана быть одной из концевых вершин данной цепи; вторую её концевую вершину обозначим через w_i (вершина w_i может совпадать с одним из полюсов A, B схемы S').

Пусть $C_{\tilde\sigma^{(1)}}$ — цепь длины n, состоящая из контактов $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$, а $C_{\tilde\sigma^{(2)}}$ — цепь длины n, состоящая из контактов $x_1^{\overline{\sigma}_1}, \ldots, x_n^{\overline{\sigma}_n}$. Очевидно, что для любого $i \in \{1, 2\}$ цепь $C_{\tilde\sigma^{(i)}}$ проводит на наборе $\tilde\sigma^{(i)}$ и не проводит ни на каком другом двоичном n-разрядном наборе. Добавим в схему S' новую вершину A', для которой введём альтернативные обозначения $a^1_{k_1+1}$ и $a^2_{k_2+1}$, а также новую вершину B'. Для $i = 1, 2$ и $j = 1, \ldots, k_i$ возьмём экземпляр цепи $C_{\tilde\sigma^{(i)}}$ и подсоединим один его конец к вершине b^i_j,

а другой — к вершине a^i_{j+1} полученной схемы. Цепи

$$A - C^1_1 - b^1_1 - C_{\tilde\sigma^{(1)}} - a^1_2 - \ldots - a^1_{k_1} - C^1_{k_1} - b^1_{k_1} - C_{\tilde\sigma^{(1)}} - a^1_{k_1+1},$$
$$A - C^2_1 - b^2_1 - C_{\tilde\sigma^{(2)}} - a^2_2 - \ldots - a^2_{k_2} - C^2_{k_2} - b^2_{k_2} - C_{\tilde\sigma^{(2)}} - a^2_{k_2+1},$$

которые при этом получаются, обозначим для краткости через $C^1_{A,A'}$ и $C^2_{A,A'}$ соответственно; концами каждой из них являются вершины A и A'. Для каждого такого $i \in \{1,2\}$, что вершина w_i отлична от вершин A, B (назовём выполнение этого условия *случаем* $<i>$), возьмём ещё один экземпляр цепи $C_{\tilde\sigma^{(i)}}$ и подсоединим один его конец к вершине w_i, а другой — к вершине B' полученной схемы. Считаем, что все внутренние вершины всех экземпляров цепей $C_{\tilde\sigma^{(1)}}$ и $C_{\tilde\sigma^{(2)}}$, добавленных в схему S', отличны друг от друга, от вершин схемы S' и от каждой из вершин A', B'.

Напомним, что $A', B' \in \{V_1, \ldots, V_d\}$. Будем считать, что каждая вершина из множества $\{V_1, \ldots, V_d\} \setminus \{A', B'\}$ совпадает с вершиной A. Итоговую контактную схему с полюсами $A, B, V_1, \ldots, V_d$ обозначим через S.

Докажем, что между полюсами A и B схемы S при отсутствии в ней неисправностей реализуется функция $f(\tilde x^n)$. На любом двоичном наборе $\tilde\pi$ длины n, отличном от наборов $\tilde\sigma^{(1)}$ и $\tilde\sigma^{(2)}$, ни одна из цепей $C_{\tilde\sigma^{(1)}}$, $C_{\tilde\sigma^{(2)}}$, подсоединённых к схеме S' для получения схемы S, не проводит, поэтому схема S функционирует на наборе $\tilde\pi$ между полюсами A и B в точности, как схема S' на этом же наборе и выдаёт на нём значение $f(\tilde\pi)$. Рассмотрим произвольное $i \in \{1,2\}$. Если $f(\tilde\sigma^{(i)}) = 1$, то на наборе $\tilde\sigma^{(i)}$ схема S' выдаёт значение 1, а значит, схема S, полученная из S' добавлением некоторых вершин и контактов, выдаёт на нём между полюсами A и B значение 1, равное $f(\tilde\sigma^{(i)})$.

Пусть $f(\tilde\sigma^{(i)}) = 0$. На наборе $\tilde\sigma^{(i)}$ ни один из экземпляров цепи $C_{\tilde\sigma^{(3-i)}}$ не проводит, поэтому на данном наборе схема S функционирует между полюсами A и B в точности, как схема S_i, полученная из схемы S удалением всех экземпляров цепи $C_{\tilde\sigma^{(3-i)}}$, добавленных в ходе преобразования схемы S' в схему S. Будем считать, что S_i — двухполюсная контактная схема с полюсами A и B. Легко видеть, что схема S_i получается из схемы S' добавлением вершин A' и B', а затем добавлением для каждого $j = 1, \ldots, k_i$ экземпляра цепи $C_{\tilde\sigma^{(i)}}$, концы которого отождествляются с вершинами b^2_j и a^2_{j+1}, а также — в случае $<i>$ — добавлением ещё одного экземпляра цепи $C_{\tilde\sigma^{(i)}}$, концы которого отождествляются с вершинами w_i и B'. В силу введённых обозначений каждый контакт подсхемы $S'_{\tilde\sigma^{(i)}}$, состоящей из всех вершин и всех проводящих на наборе $\tilde\sigma^{(i)}$ контактов схемы S', содержится в одной из её компонент связности $C^i_1, \ldots, C^i_{k_i}, C^i_B$, каждая из которых представляет собой цепь, причём компоненты связности $C^i_1, \ldots, C^i_{k_i}$ попарно различны, а в случае $k_i \geqslant 2$ компонента связности C^i_B отлична от $C^i_2, \ldots, C^i_{k_i}$. Из равенства $f(\tilde\sigma^{(i)}) = 0$ вытекает, что на наборе $\tilde\sigma^{(i)}$ в схеме S' нет проводимости между полюсами, поэтому вершины A и B содержатся в разных компонентах связности подсхемы $S'_{\tilde\sigma^{(i)}}$ и компонента связности C^i_B отлична также от C^i_1. Таким образом, компоненты связности $C^i_1, \ldots, C^i_{k_i}, C^i_B$ подсхемы $S'_{\tilde\sigma^{(i)}}$ попарно различны. При переходе от схемы S' к схеме S_i сначала к ним добавляются ещё две компоненты связности $C_{A'}$ и $C_{B'}$, каждая из которых состоит из единственной вершины — соответственно A' и B', а затем компоненты связности $C^i_1, \ldots, C^i_{k_i}, C_{A'}$ объединяются в одну цепь $C^i_{A,A'}$, а компоненты связности C^i_B и $C_{B'}$ в случае $<i>$ объединяются в одну цепь $B - C^i_B - w_i - C_{\tilde\sigma^{(i)}} - B'$, которую мы обозначим через $C^i_{B,B'}$, но полюсы A и B всё равно остаются в разных компонентах связности подсхемы, состоящей из всех вершин и всех проводящих на наборе $\tilde\sigma^{(i)}$ контактов схемы S_i. Отсюда следует, что на наборе $\tilde\sigma^{(i)}$ нет проводимости между вершинами A и B в схеме S_i, а значит, и в схеме S. Тем самым установлено, что схема S на указанном наборе выдаёт между полюсами A и B значение 0, равное $f(\tilde\sigma^{(i)})$.

В итоге получаем, что схема S реализует между полюсами A и B функцию $f(\tilde{x}^n)$. Докажем, что при выборе в качестве тестового полюсного множества $\mathcal{P}$ данная схема неизбыточна и допускает ЕПТ $\{\tilde{\sigma}^{(1)}, \tilde{\sigma}^{(2)}\}$; отсюда будет следовать неравенство $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \leqslant 2$. Рассмотрим произвольное $i \in \{1, 2\}$. Из построения схемы S нетрудно получить, что объединение всех её контактов, проводящих на наборе $\tilde{\sigma}^{(i)}$, т. е. всех контактов $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$ при $i = 1$ и всех контактов $x_1^{\overline{\sigma}_1}, \ldots, x_n^{\overline{\sigma}_n}$ при $i = 2$, представляет собой цепь $C^i_{A,A'}$ между полюсами A и A' схемы, объединённую — в случае $<i>$ — с цепью $C^i_{B,B'}$ между полюсами B и B' схемы, причём указанные две цепи не имеют общих вершин. Отсюда вытекает, что при обрыве произвольного контакта, содержащегося в цепи $C^i_{A,A'}$, проводимость между полюсами A и A' схемы S на наборе $\tilde{\sigma}^{(i)}$ меняется с 1 на 0, а в случае $<i>$ верен аналогичный факт с заменой всюду A на B и A' на B'. Из приведённых рассуждений следует, что неисправность любого контакта схемы S обнаруживается на одном из наборов $\tilde{\sigma}^{(1)}, \tilde{\sigma}^{(2)}$ хотя бы между одной из двух пар полюсов $\{A, A'\}$, $\{B, B'\}$, каждая из которых по построению принадлежит множеству $\mathcal{P}$. Тем самым установлено, что схема S неизбыточна и допускает ЕПТ $\{\tilde{\sigma}^{(1)}, \tilde{\sigma}^{(2)}\}$. Неравенство $D^{d,\mathcal{P}}_{\text{ЕП}}(f) \leqslant 2$, а вместе с ним теорема 1 доказаны. ■

Заключение

Для любого целого $d \geqslant 2$ описан достаточно обширный класс тестовых полюсных множеств $\mathcal{P}$, для каждого из которых найдено точное значение величины $D^{d,\mathcal{P}}_{\text{ЕП}}(f)$ для произвольной булевой функции $f(\tilde{x}^n)$. Ранее в работе [11] получен аналогичный результат для $d = 1$ и произвольного тестового полюсного множества $\mathcal{P}$. Не охваченным пока остаётся случай, когда $d \geqslant 2$ и любые две пары полюсов из множества $\mathcal{P}\backslash\{\{A, B\}\}$ пересекаются, т. е. содержат общий полюс; однако и для этого случая у нас есть наработки. Сравнение равенства $D^{d,\mathcal{P}}_{\text{ЕП}}(n) = \min(2, n)$, установленного в следствии 1, с равенствами $D_{\text{ЕП}}(n) = n$ из [6] и $D^{1,\mathcal{P}}_{\text{ЕП}}(n) = \min(3, n)$ из [11] демонстрирует, что путём добавления в двухполюсные контактные схемы ещё по крайней мере двух полюсов можно в ряде случаев уменьшить функцию Шеннона длины ЕПТ размыкания с n до 2 и даже уменьшить её с 3 до 2 по сравнению с трёхполюсными контактными схемами. Тем самым установлена практическая целесообразность реализации булевых функций контактными схемами с двумя и более дополнительными полюсами.

ЛИТЕРАТУРА

1. *Лупанов О. Б.* Асимптотические оценки сложности управляющих систем. М.: Изд-во Моск. ун-та, 1984. 138 с.
2. *Чегис И. А., Яблонский С. В.* Логические способы контроля работы электрических схем // Труды МИАН. 1958. Т. 51. С. 270–360.
3. *Яблонский С. В.* Некоторые вопросы надёжности и контроля управляющих систем // Математические вопросы кибернетики. Вып. 1. М.: Наука, 1988. С. 5–25.
4. *Редькин Н. П.* Надёжность и диагностика схем. М.: Изд-во Моск. ун-та, 1992. 192 с.
5. *Редькин Н. П.* О проверяющих тестах замыкания и размыкания // Методы дискретного анализа в оптимизации управляющих систем. Вып. 40. Новосибирск: ИМ СО АН СССР, 1983. С. 87–99.
6. *Попков К. А.* О проверяющих тестах размыкания для контактных схем // Дискретная математика. 2017. Т. 29. Вып. 4. С. 66–86.
7. *Попков К. А.* О диагностических тестах размыкания для контактных схем // Дискретная математика. 2019. Т. 31. Вып. 2. С. 124–143.

8. *Мадатян Х. А.* Полный тест для бесповторных контактных схем // Проблемы кибернетики. Вып. 23. М.: Наука, 1970. С. 103–118.
9. *Редькин Н. П.* О диагностических тестах для контактных схем // Вестник Московского университета. Сер. 1. Математика. Механика. 2019. № 2. С. 35–37.
10. *Попков К. А.* О полных диагностических тестах для контактных схем при обрывах и/или замыканиях контактов // Изв. вузов. Поволжский регион. Физико-математические науки. 2019. № 3 (51). С. 3–24.
11. *Попков К. А.* Короткие проверяющие тесты размыкания для контактных схем с дополнительным полюсом // Дискретная математика. 2024. Т. 36. Вып. 4. С. 117–137.

REFERENCES

1. *Lupanov O. B.* Asimptoticheskie otsenki slozhnosti upravlyayushchikh sistem [Asymptotic Bounds of the Complexity of Control Systems]. Moscow, MSU Publ., 1984. 138 p. (In Russian)
2. *Chegis I. A. and Yablonskiy S. V.* Logicheskie sposoby kontrolya raboty elektricheskikh skhem [Logical ways of monitoring the operation of electrical circuits]. Trudy MIAN, 1958, vol. 51, pp. 270–360. (in Russian)
3. *Yablonskiy S. V.* Nekotorye voprosy nadezhnosti i kontrolya upravlyayushchikh sistem [Some questions of reliability and verification of control systems]. Matematicheskie Voprosy Kibernetiki, iss. 1, Moscow, Nauka Publ., 1988, pp. 5–25. (in Russian)
4. *Red'kin N. P.* Nadezhnost' i diagnostika skhem [Circuits Reliability and Diagnostics]. Moscow, MSU Publ., 1992. 192 p. (in Russian)
5. *Red'kin N. P.* O proveryayushchikh testakh zamykaniya i razmykaniya [On fault detection tests of closure and opening]. Metody Diskretnogo Analiza v Optimizatsii Upravlyayushchikh Sistem, iss. 40, Novosibirsk, Math. Inst. Sib. Br. USSR Acad. Sci., 1983, pp. 87–99. (in Russian)
6. *Popkov K. A.* On fault detection tests of contact break for contact circuits. Discrete Math. Appl., 2018, vol. 28, no. 6, pp. 369–383.
7. *Popkov K. A.* On diagnostic tests of contact break for contact circuits. Discrete Math. Appl., 2020, vol. 30, no. 2, pp. 103–116.
8. *Madatyan Kh. A.* Polnyy test dlya bespovtornykh kontaktnykh skhem [Complete test for non-repetitive contact circuits]. Problemy Kibernetiki, iss. 23, Moscow, Nauka Publ., 1970, pp. 103–118. (in Russian)
9. *Red'kin N. P.* Diagnostic tests for contact circuits. Moscow Univ. Math. Bull., 2019, vol. 74, no. 2, pp. 62–64.
10. *Popkov K. A.* O polnykh diagnosticheskikh testakh dlya kontaktnykh skhem pri obryvakh i/ili zamykaniyakh kontaktov [On complete diagnostic tests for contact circuits under breaks and/or closures of contacts]. Izvestiya Vysshikh Uchebnykh Zavedeniy. Povolzhskiy Region. Fiziko-matematicheskiye nauki, 2019, no. 3 (51), pp. 3–24. (in Russian)
11. *Popkov K. A.* Korotkie proveryayushchie testy razmykaniya dlya kontaktnykh skhem s dopolnitel'nym polyusom [Short fault detection tests of contact break for contact circuits with an additional pole]. Diskretnaya Matematika, 2024, vol. 36, no. 4, pp. 117–137. (in Russian)

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.111.6 DOI 10.17223/20710410/71/5

ПЕРЕЧИСЛЕНИЕ 2-ДЕРЕВЬЕВ С ОРИЕНТИРОВАННЫМИ ЯЧЕЙКАМИ

В. Р. Верденко*, В. А. Воронов*,**

*Адыгейский государственный университет, г. Майкоп, Россия
**Московский физико-технический институт, г. Долгопрудный, Россия

E-mail: v-vor@yandex.ru

Рассматривается задача о перечислении 2-деревьев с ориентированными ячейками с точностью до изоморфизма. Под 2-деревом мы понимаем простой граф, полученный из K_3 последовательным добавлением вершин, соединенных с концами некоторого ребра. Будем говорить, что ячейки 2-дерева ориентированы, если в каждой ячейке (треугольнике) вершины независимо перенумерованы числами $1, 2, 3$. Ячейки частично ориентированы, если в каждой ячейке независимо отмечена одна вершина. При помощи теоремы Редфилда — Пойа и характеристики неподобия для корневых структур вычислены производящие функции для числа 2-деревьев с ориентированными и частично ориентированными ячейками.

Ключевые слова: *2-деревья, теория перечисления Пойа, производящие функции.*

ENUMERATION OF 2-TREES WITH DIRECTED CELLS

V. R. Verdenko*, V. A. Voronov*,**

*Adyghe State University, Maikop, Russia
**Moscow Institute of Physics and Tecnology, Dolgoprudnyy, Russia

We consider the problem of enumerating 2-trees with oriented cells up to isomorphism. A 2-tree is a simple graph obtained from K_3 by the iterative addition of vertices connected to the ends of some edge. The cells of a 2-tree are oriented if in each cell (triangle) the vertices are independently labeled 1, 2, 3. The cells are partially oriented if one vertex is independently labeled in each cell. Using the Redfield — Polya theorem and dissymmetry lemma, we compute generating functions for the number of 2-trees with oriented and partially oriented cells.

Keywords: *2-trees, Polya counting, generating functions.*

Введение

Перечисление конечных множеств с заданной на них структурой с точностью до изоморфизма является одной из основных задач комбинаторики. Обычно в задачах данного типа требуется предъявить некий алгоритм вычисления последовательности

$\{q_n : n = 0, 1, 2, \dots\}$, где q_n — число неизоморфных структур, заданных на n-элементном множестве. Существуют весьма эффективные реализации переборных алгоритмов, позволяющих использовать параллельные вычисления для явного построения всех попарно неизоморфных структур. Наиболее известен комплекс утилит NAUTY AND TRACES, в течение нескольких десятилетий разрабатываемый Б. Маккеем [1].

Кроме непосредственного перебора всех структур для n элементов, существует ряд методов, позволяющих вычислять q_n намного быстрее. Некоторые задачи такого типа решаются при помощи леммы Бёрнсайда, основанной на ней теоремы Редфилда — Пойа и подходов, разработанных в 1950–70-е годы Ф. Харари, Э. Палмером, Р. Робинсоном и другими авторами. Обычно при этом выводятся соотношения для производящей функции, $\varphi(x) = \sum\limits_{k=0}^{\infty} q_k x^k$, при помощи которых удаётся последовательно вычислять её коэффициенты. При этом, в отличие от переборных алгоритмов, объём вычислений растет существенно медленнее, чем q_n.

Классическими являются задачи о перечислении непомеченных графов на n вершинах с точностью до изоморфизма, непомеченных деревьев, 2-деревьев. Соотношения, позволяющие вычислять производящие функции в этих случаях, приведены в монографии Ф. Харари и Э. Палмера [2]. В последние десятилетия данный подход был распространен на непомеченные 2-деревья с ориентированными рёбрами [3], 2-деревья с многоугольными ячейками [4], k-деревья [5], ориентированные гиперграфы [6], k-однородные гиперграфы [7].

Исторически первыми приложениями теории Пойа были задачи о перечислении химических соединений [8], которые продолжают изучаться и в настоящее время. Химические связи в молекуле не могут образовывать произвольное 2-дерево, поскольку валентности атомов ограничены. Тем не менее рассматриваемые в работе методы после адаптации под конкретную задачу могут быть применены для перечисления сложных молекул (например, соединений фуллеренов и нанотрубок, гетерофуллеренов, других полициклических соединений и их изотопных модификаций).

В данной работе решаются две задачи о перечислении 2-деревьев с n ориентированными ячейками, т. е. 2-деревьев, в которых на каждом треугольнике (ячейке) независимо введена ориентация одного из двух типов (рис. 1). В сравнении с перечислением непомеченных 2-деревьев [2] без ориентации некоторые формулы упрощаются, некоторые — наоборот, становятся сложнее и требуют дополнительных рассуждений.

а *б*

Рис. 1. 2-Дерево с ориентированными ячейками (*а*) и частично ориентированное 2-дерево (*б*)

Введём необходимые определения.

Определение 1. Треугольник K_3 и любой граф, который можно получить из K_3 при помощи операции добавления вершины, соединённой с концами некоторого ребра, будем называть 2-деревом.

На каждом шаге индуктивного построения добавляется одна вершина, два ребра и один треугольник. Будем называть треугольники ячейками.

Определение 2. Ячейка — это треугольник в 2-дереве.

Можно также рассматривать 2-дерево как 3-однородный гиперграф, в котором каждая ячейка является ребром гиперграфа. Фактически под ориентацией ячеек мы понимаем ориентацию рёбер соответствующего гиперграфа.

Определение 3. Будем говорить, что ячейки 2-дерева ориентированы, если в каждой ячейке вершины пронумерованы числами 1, 2, 3. В ячейках, имеющих общие вершины, нумерация выбирается независимо.

Определение 4. Частично ориентированное 2-дерево — это 2-дерево, в котором в каждой ячейке выделено по одной вершине.

Для краткости будем называть 2-деревья с ориентированными и частично ориентированными ячейками ориентированными и частично ориентированными 2-деревьями. В настоящей работе рассматривается только ориентация ячеек.

Определение 5. Торцевое ребро 2-дерева — это ребро, которое принадлежит только одной ячейке.

Определение 6. Ребро uv (частично) ориентированного 2-дерева G будем называть симметричным, если существует автоморфизм ψ графа G, для которого $\psi(u) = v$, $\psi(v) = u$.

Определение 7. Два подграфа H_1, H_2 графа G называются подобными, если существует такой автоморфизм ψ графа G, что $\psi(H_1) = H_2$. Говоря про неподобные подграфы (в частности, вершины, рёбра, ячейки), имеем в виду, что такого автоморфизма не существует.

Далее будем рассматривать неподобные вершины, рёбра, ячейки. Подобие задаёт отношение эквивалентности. Говоря, что в графе есть k неподобных объектов, будем подразумевать, что можно выделить максимум k попарно неподобных объектов такого вида, т. е. k — это число классов эквивалентности.

1. Теорема Редфилда — Пойа

Для полноты изложения приведём частный случай теоремы Редфилда — Пойа, которого достаточно для наших целей.

Определение 8. Пусть дана группа подстановок A. Цикловой индекс группы A — это многочлен от переменных $s_1, s_2, \ldots, s_n$:

$$Z(A) = Z(A; s_1, \ldots, s_n) = |A|^{-1} \sum_{\alpha \in A} \sum_{k=1}^{n} s_k^{j_k(\alpha)},$$

где $j_k(\alpha)$ означает количество циклов длины k в подстановке α.

Например, для цикловых индексов симметрической группы по определению имеем $Z(S_1) = s_1$, $Z(S_2) = \dfrac{1}{2}(s_1^2 + s_2)$. Далее $Z(S_n)$ можно вычислять с помощью рекуррентной формулы

$$Z(S_n) = n^{-1} \sum_{k=1}^{n} s_k Z(S_{n-k}).$$

Следуя [2], будем использовать такое обозначение: если $f(x)$ — производящая функция, то $Z(A; f(x))$ — результат подстановки $f(x^k)$ вместо s_k в $Z(A)$, $k = 1, 2, \ldots, n$. Для краткости будем писать $Z(A_1 - A_2; f(x))$ вместо $Z(A_1; f(x)) - Z(A_2; f(x))$.

Приведём также следующий классический результат, который применяется при перечислении деревьев, 2-деревьев и т. д., чтобы учитывать автоморфизмы, переставляющие несколько поддеревьев, присоединённых к корневой вершине или корневому ребру:

Лемма 1 [2]. Для произвольной производящей функции $f(\cdot)$ справедливо соотношение

$$\sum_{n=0}^{\infty} Z(S_n; f(x)) = \exp\left(\sum_{k=1}^{\infty} \frac{f(x^k)}{k}\right). \tag{1}$$

Как обычно, полагаем, что элементарная функция от производящей функции вычисляется путём подстановки в ряд Маклорена, а равенство между полученными формальными рядами означает равенство всех коэффициентов при одинаковых степенях [9].

Теперь перейдём к частному случаю теоремы Редфилда — Пойа и сформулируем его для вершинных раскрасок графа. Введём следующие обозначения:

— P — конечное или счётное множество цветов;
— каждый цвет $p \in P$ имеет неотрицательную целую стоимость $w(p)$;
— задана $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots$ — производящая функция для числа цветов стоимости k, то есть a_k означает количество цветов стоимости k;
— G — некоторый граф на n вершинах;
— $A = \mathrm{Aut}(G) \subseteq S_n$ — группа автоморфизмов G, действующая на множестве вершин;
— стоимость раскраски графа равна сумме стоимостей цветов вершин;
— b_k — число классов эквивалентности раскрасок графа G стоимости k.

Теорема 1 [2, 8]. Производящая функция $B(x)$ для числа раскрасок графа G стоимости k может быть вычислена по формуле

$$B(x) = b_0 + b_1 x + b_2 x^2 + \ldots = Z(A; f(x)).$$

Выделим особый случай раскраски n-элементного множества, когда цвета должны быть попарно различны.

Утверждение 1 [2]. Пусть $C(x)$ — производящая функция для числа раскрасок вершин графа K_n стоимости k попарно различными цветами при тех же предположениях, что и в теореме 1. Тогда

$$C(x) = Z(A_n - S_n; f(x)).$$

Эффективным приёмом, который потребуется в дальнейшем, является переход от структур с выделенным корнем к структурам без корня, основанный на некотором линейном соотношении между числом классов эквивалентности (неподобных корневых структур) в графе. Это соотношение называют характеристикой неподобия, а соответствующее утверждение — леммой о неподобии ("dissymmetry lemma", "dissymmetry theorem"). Предположим, что $H_1, H_2, \ldots, H_m$ — структуры (вершины, рёбра, ячейки), которые могут присутствовать в классе графов $\mathcal{G}$. Обозначим $\#(H_i, G)$ число неподобных структур H_i в графе G. Пусть $\varphi_i(x)$ — производящая функция для графов с выделенной корневой структурой H_i.

Лемма 2 [2]. Если при фиксированных коэффициентах $\alpha_1, \ldots, \alpha_m$ для любого графа $G \in \mathcal{G}$ выполнено

$$\sum_{i=1}^{m} \alpha_i \cdot \#(H_i, G) = 1,$$

то производящая функция $\varphi(x)$ для числа графов на n вершинах из $\mathcal{G}$ удовлетворяет соотношению

$$\sum_{i=1}^{m} \alpha_i \varphi_i(x) = \varphi(x).$$

Далее эта техника применяется к 2-деревьям с ориентированными ячейками.

2. Случай ориентированных ячеек

Докажем соотношения, позволяющие вычислять производящую функцию в случае ориентированных ячеек, который оказывается несколько проще, чем случай частично ориентированных ячеек. При этом формулы, связывающие производящие функции корневых структур, заметно упрощаются в сравнении с неориентированным случаем, приведённым в [2].

Теорема 2. Пусть q — количество неподобных рёбер в ориентированном 2-дереве G; s — количество неподобных ячеек в G. Тогда справедливо равенство (характеристика неподобия для ориентированных 2-деревьев)

$$q - 2s = 1. \tag{2}$$

Доказательство. Проведем индукцию по числу ячеек. В случае одной ориентированной ячейки имеем верное равенство $3 - 2 = 1$.

Пусть дано ориентированное 2-дерево G, имеющее $k > 1$ ячеек. Согласно определению 2-дерева, среди них найдётся «висячая» ячейка uvw, которая имеет с остальными одно общее ребро. По предположению индукции равенство (2) имеет место после удаления uvw. Обозначим через G' граф, полученный после удаления этой ячейки.

Отметим, что при введённой в ячейке ориентации все три ребра uv, vw, uw попарно не подобны. Рассмотрим два случая:

1) Если рассматриваемая ячейка не подобна ячейкам G', то при её добавлении увеличивается на 1 число классов эквивалентности ячеек и на 2 — число классов рёбер, так как одно из рёбер uv, vw, uw принадлежало графу G'.

2) Если uvw подобна какой-либо ячейке G', то её ребра подобны соответствующим рёбрам ячеек из её класса эквивалентности, и при добавлении uvw число классов рёбер не увеличивается.

Таким образом, в обоих случаях равенство (2) сохраняется после добавления ячейки uvw. ■

Обозначим:

- $t_{\triangle,1}(x)$ — производящая функция ориентированных 2-деревьев с заданным числом ячеек;
- $q(x)$ — производящая функция ориентированных 2-деревьев с выделенным корневым ребром;
- $s(x)$ — производящая функция ориентированных 2-деревьев с корневой ячейкой.

Используя лемму 2, из равенства (2) получаем

$$q(x) - 2s(x) = t_{\triangle,1}(x).$$

Определение 9. Ориентированное 2-дерево G с корневым ребром uv будем называть симметричным, если существует автоморфизм G (сохраняющий ориентацию ячеек), который переводит u в v, а v в u. Если такого автоморфизма не существует, будем называть G несимметричным.

Введём следующие обозначения для производящих функций ориентированных 2-деревьев с различными корневыми структурами:

— $N_1(x)$ — если корнем является торцевое ребро uv и граф несимметричен;
— $M(x)$ — если корнем является произвольное ребро (не обязательно торцевое) и граф симметричен;
— $N(x)$ — если корнем является произвольное ребро и граф несимметричен.

Отметим, что 2-дерево с торцевым ребром не может быть симметричным с учётом ориентации. Покажем, что эти функции связаны соотношениями, которые позволяют последовательно вычислять их коэффициенты.

Утверждение 2. Производящая функция несимметричных ориентированных 2-деревьев с корнем в торцевом ребре удовлетворяет соотношению

$$N_1(x) = 3x(1 + M(x) + 2N(x))^2. \tag{3}$$

Доказательство. Построим 2-дерево, начиная с одной ориентированной ячейки (множитель x). В ней можно выделить корень (торцевое ребро) тремя способами (множитель 3).

К двум другим рёбрам можно ничего не присоединять (слагаемое 1), можно присоединить симметричный граф (слагаемое $M(x)$) или несимметричный двумя способами (слагаемое $2N(x)$).

Поскольку исходная ячейка является ориентированной, учитывать перестановки присоединённых графов не требуется, т. е. полученный множитель $(1+M(x)+2N(x))^2$ перечисляет все способы присоединения подграфов к ячейке. ■

Утверждение 3. Для производящей функции симметричных ориентированных 2-деревьев с корневым (не обязательно торцевым) ребром справедливо равенство

$$M(x) = \sum_{n=1}^{\infty} Z(S_n; N_1(x^2)). \tag{4}$$

Доказательство. Заметим, что в ориентированном 2-дереве ребро может быть симметричным только в том случае, если к нему присоединены пары «зеркальных» подграфов, так как при автоморфизме, меняющем местами концы ребра, ориентированная ячейка не может перейти в себя.

Пусть uv — симметричное корневое ребро. Будем присоединять к нему пары одинаковых подграфов, перечисляемых функцией $N_1(x)$. При этом число ячеек удваивается, т. е. пары «зеркальных» копий перечисляются функцией $N_1(x^2)$. Таких пар возьмём n для каждого $n = 1, 2, \ldots$ Так как не имеет значения, в каком порядке их присоединять, то следует учесть действие симметрической группы S_n.

Отметим, что суммирование здесь ведётся с $n = 1$, поскольку K_2 без присоединённых подграфов по определению не является 2-деревом. ■

Утверждение 4. Производящая функция ориентированных 2-деревьев с несимметричным корневым (не обязательно торцевым) ребром может быть найдена из уравнения

$$M(x) + 2N(x) = \sum_{n=1}^{\infty} Z(S_n; 2N_1(x)). \tag{5}$$

Доказательство. Покажем, что правая и левая части равны, так как обе описывают ориентированные 2-деревья с корневым ребром uv и *помеченными независимо от ориентации ячеек вершинами* u и v. Здесь мы считаем различными несимметричные изоморфные графы G_1, G_2, для которых изоморфизм переводит u в v', а v в u', причём uv — корневое ребро в G_1, $u'v'$ — корневое ребро в G_2. В определении $N(x)$ и $N_1(x)$ выше мы считали такие графы одинаковыми.

В самом деле, чтобы получить правую часть уравнения, будем действовать аналогично доказательству формулы (4), но присоединять надо не две «зеркальные» копии подграфа, а один несимметричный подграф одним из двух способов.

Чтобы получить левую часть, выделим из присоединённых несимметричных подграфов «зеркальные» пары (если они есть) и составим из них максимальный подграф с тем же корневым ребром (но симметричным). Такие подграфы перечисляются функцией $M(x)$. Остальные присоединённые подграфы образуют подграф с несимметричным корневым ребром и перечисляются функцией $2N(x)$, поскольку мы полагаем, что вершины u, v помечены. Для каждого графа, полученного присоединением к uv нескольких подграфов, это разложение на симметричную и несимметричную составляющие определяется однозначно. ■

С помощью соотношения (1) преобразуем формулы (4), (5) следующим образом:

$$1 + M(x) = \exp\left(\sum_{n=1}^{\infty} \frac{N_1(x^{2n})}{n}\right); \tag{6}$$

$$1 + M(x) + 2N(x) = \exp\left(\sum_{n=1}^{\infty} \frac{2N_1(x^{n})}{n}\right). \tag{7}$$

Теперь при помощи формул (3), (6), (7) найдём коэффициенты рядов $M_1(x)$, $N_1(x)$, $M(x)$, $N(x)$. Младшие коэффициенты вычислим вручную. Далее, за счёт домножения на x в формуле $N_1(x)$, зная свободный коэффициент у $M(x)$ и $N(x)$, можно найти коэффициент при x в $N_1(x)$, затем коэффициент при x в $M(x)$ и $N(x)$, затем коэффициент при x^2 в $N_1(x)$ и т. д. После необходимых вычислений получим

$$\begin{aligned} N_1(x) &= 3x + 36x^2 + 666x^3 + 14268x^4 + 3336231x^5 + \dots, \\ M(x) &= 3x^2 + 42x^4 + 784x^6 + 17163x^8 + 409386x^{10} \dots, \\ N(x) &= 3x + 45x^2 + 910x^3 + 20376x^4 + 493803x^5 + \dots \end{aligned}$$

Найдём производящую функцию для ориентированных 2-деревьев с корневым ребром:

$$q(x) = M(x) + N(x) = 3x + 48x^2 + 910x^3 + 20418x^4 + 493803x^5 + \dots$$

Утверждение 5. Производящая функция для ориентированных 2-деревьев с корневой ячейкой удовлетворяет соотношению

$$s(x) = x(1 + M(x) + 2N(x))^3.$$

Доказательство. Возьмём ориентированную корневую ячейку (множитель x) и к трём её рёбрам присоединим подграфы, перечисляемые множителем $1 + M(x) + 2N(x)$. Поскольку корневая ячейка ориентирована, учитывать перестановки подграфов не требуется. ■

Подставляя найденные выше ряды, получаем

$$s(x) = x + 18x^2 + 387x^3 + 9024x^4 + 223893x^5 + \dots$$

Теперь с помощью соотношения (2) можно найти производящую функцию для ориентированных 2-деревьев:

$$t_{\triangle,1}(x) = q(x) - 2s(x) = x + 12x^2 + 136x^3 + 2370x^4 + 46017x^5 + \dots$$

Вычисление данных производящих функций реализовано в python. Код размещён на Github [10].

3. Случай частично ориентированных ячеек

Вывод соотношений для частично ориентированных 2-деревьев во многом повторяет предыдущий случай, поэтому часть пояснений опущена.

Начнем с характеристики неподобия.

Теорема 3. Если для одного частично ориентированного 2-дерева q — количество неподобных ребер, s_0 — количество неподобных ячеек, у которых все три ребра неподобны, s_1 — количество неподобных ячеек, у которых два ребра подобны друг другу, то

$$q - 2s_0 - s_1 = 1. \tag{8}$$

Доказательство. Проведём индукцию по числу ячеек k, начиная с одной частично ориентированной ячейки. При $k = 1$ имеем $s_0 = 0$, $s_1 = 1$, и равенство (8) верно.

В 2-дереве G с $k+1$ ячейкой найдётся «висячая» ячейка uvw, присоединённая по ребру vw. Пусть для графа $G' = G \setminus u$, в котором эта ячейка удалена, формула (8) выполнена. Докажем, что она выполнена и для G.

Если ячейка uvw подобна какой-либо ячейке G', то все её рёбра принадлежат классам эквивалентности из G', а значит, при добавлении этой ячейки числа q, s_0, s_1 не изменяются.

Если новая ячейка не подобна ячейкам G', то возникает ещё два случая:

1) Частичная ориентация uvw выбрана таким образом, что отмечена одна из вершин v, w. Тогда рёбра uv, vw, uw неподобны друг другу, а значит, q увеличивается на 2, s_0 на 1, s_1 не изменяется.

2) Отмечена вершина u, рёбра uv, uw подобны. Тогда q увеличивается на 1, s_0 не изменяется, s_1 увеличивается на 1.

В обоих случаях формула (8) справедлива для G. ■

Перечислим обозначения для производящих функций корневых структур на частично ориентированных 2-деревьях:

— $s_0(x)$ — корнем является ячейка с попарно неподобными рёбрами;
— $s_1(x)$ — корнем является ячейка с двумя подобными рёбрами;
— $q(x)$ — корнем является произвольное ребро;
— $N_1(x)$ — корнем является торцевое ребро, граф несимметричен;
— $M_1(x)$ — корнем является торцевое ребро, граф симметричен;
— $N(x)$ — корнем является произвольное ребро, граф несимметричен;
— $M(x)$ — корнем является произвольное ребро, граф симметричен.

Пусть $t_{\triangle,2}(x)$ — производящая функция частично ориентированных 2-деревьев. Согласно лемме 2, из (8) получаем соотношение для производящих функций:

$$q(x) - 2s_0(x) - s_1(x) = t_{\triangle,2}(x). \tag{9}$$

Найдём соотношения, которые связывают производящие функции для корневых структур.

Утверждение 6. Для случая торцевого ребра и симметричного графа имеем

$$M_1(x) = x(1 + M(x^2) + 2N(x^2)).$$

Доказательство. Построим граф, начиная с одной ячейки с отмеченной вершиной, содержащей корневое ребро (множитель x).

Корнем обязано быть ребро напротив отмеченной вершины, так как граф симметричен. К другим двум рёбрам исходной ячейки можно ничего не присоединять (слагаемое 1), присоединить два одинаковых симметричных графа (слагаемое $M(x^2)$) или присоединить два одинаковых несимметричных графа одним из двух способов (слагаемое $2N(x^2)$). ■

Утверждение 7. Для случая торцевого ребра и несимметричного графа имеем

$$N_1(x) = 3xZ\left(A_2 - S_2, 1 + M(x) + 2N(x)\right) + x(1 + M(x^2) + 2N(x^2)).$$

Доказательство. Возьмём ячейку с выделенным торцевым ребром uv без какой-либо ориентации (множитель x). Отдельно разберём два случая, соответствующие первому и второму слагаемому. В первом случае несимметричность обеспечивают присоединённые графы, во втором — ориентация исходной ячейки:

1) К другим двум рёбрам присоединим различные подграфы, перечисляемые функцией $1 + M(x) + 2N(x)$. Подстановка в цикловой индекс $Z(A_2 - S_2) = (s_1^2 - s_2)/2$ позволяет перечислить только те случаи, когда присоединённые графы различны. Затем мы тремя способами можем выделить вершину в исходной ячейке (множитель 3).

2) Множитель $1 + M(x^2) + 2N(x^2)$ означает, что к ячейке присоединены два одинаковых подграфа. В этом случае ориентация ячейки, нарушающая симметрию, единственна с точностью до автоморфизма. ■

Утверждение 8. В случаях произвольного (не обязательного торцевого) симметричного или несимметричного корневого ребра справедливы соотношения

$$M(x) = \sum_{n=1}^{\infty} Z(S_n; M_1(x) + N_1(x^2)); \tag{10}$$

$$M(x) + 2N(x) = \sum_{n=1}^{\infty} Z(S_n; M_1(x) + 2N_1(x)). \tag{11}$$

Доказательство. Ряды для $M(x)$ и $M(x) + 2N(x)$ получены аналогично формулам (4) и (5). Отличие только в том, что в случае частично ориентированных 2-деревьев существуют симметричные подграфы с корнем в торцевом ребре, поэтому в функции, которая подставляется в цикловой индекс, есть слагаемое $M_1(x)$. ■

Преобразуем формулы (10) и (11) с помощью соотношения (1):

$$1 + M(x) = \exp\left(\sum_{n=1}^{\infty} \frac{M_1(x^n) + N_1(x^{2n})}{n}\right);$$

$$1 + M(x) + 2N(x) = \exp\left(\sum_{n=1}^{\infty} \frac{M_1(x^n) + 2N_1(x^n)}{n}\right).$$

Так как ряды для $M_1(x)$ и $N_1(x)$ содержат домножение на x, а ряды $M(x)$ и $M(x)+2N(x)$ — нет, то мы можем последовательно найти все их коэффициенты. Затем вычислим ряд $q(x) = M(x) + N(x)$:

$$N_1(x) = x + 9x^2 + 84x^3 + 921x^4 + 10914x^5 + \ldots,$$

$$M_1(x) = x + 3x^3 + 24x^5 + 235x^7 + 2649x^9 \ldots,$$
$$N(x) = x + 11x^2 + 115x^3 + 1317x^4 + 16077x^5 + \ldots,$$
$$M(x) = x + 2x^2 + 5x^3 + 15x^4 + 42x^5 + \ldots,$$
$$q(x) = 2x + 13x^2 + 120x^3 + 1332x^4 + 16119x^5 + \ldots$$

Перейдём к производящим функциям для графов с корневыми ячейками.

Утверждение 9. В случае корневой ячейки с двумя подобными рёбрами производящая функция имеет вид $s_1(x) = M_1(x)(1 + M(x))$.

Доказательство. Возьмём графы, перечисляемые $M_1(x)$, и объявим, что ячейка, которая раньше содержала корневое ребро uv, теперь является корневой. К ребру uv можно ничего не присоединять (слагаемое 1) или присоединить симметричный граф (слагаемое $M(x)$). ■

Утверждение 10. В случае корневой ячейки с тремя неподобными ребрами имеет место равенство

$$s_0(x) = xZ(A_2 - S_2; 1 + M(x) + 2N(x))(1 + M(x) + 2N(x)) + xN(x)(1 + M(x^2) + 2N(x^2)).$$

Доказательство. Пусть дана корневая ячейка uvw с выделенной вершиной u (множитель x). Неподобие рёбер uv, uw может быть обусловлено либо тем, что к ним присоединены различные подграфы, либо тем, что к ребру vw присоединён несимметричный подграф. Убедимся, что в этих двух случаях возникнут именно такие слагаемые, как в приведённой формуле:

1) К рёбрам uv, uw присоединим различные графы, перечисляемые $1 + M(x) + 2N(x)$. Подстановка в цикловой индекс $Z(A_2 - S_2)$ обеспечивает, что присоединяемые графы различны. К ребру vw присоединим произвольный граф с корневым ребром (возможно, пустой). Соответствующий множитель равен $1 + M(x) + 2N(x)$.

2) К ребру vw присоединён несимметричный подграф (множитель $N(x)$), к рёбрам uv, uw присоединены одинаковые подграфы (множитель $1 + M(x^2) + 2N(x^2)$). ■

Вычислим s_0 и s_1:

$$s_0(x) = 4x^2 + 47x^3 + 578x^4 + 7254x^5 + \ldots,$$
$$s_1(x) = x + x^2 + 5x^3 + 8x^4 + 45x^5 + \ldots$$

Подставив их в формулу (9), найдём производящую функцию для частично ориентированных 2-деревьев:

$$t_{\triangle,2}(x) = q(x) - 2s_0(x) - s_1(x) = x + 4x^2 + 21x^3 + 168x^4 + 1566x^5 + \ldots$$

Код, вычисляющий данные коэффициенты, также размещён на Github [10].

Заключение

Приведённые формулы и программная реализация могут быть адаптированы к другим задачам о k-деревьях и гиперграфах, но при этом могут потребоваться более изощрённые комбинаторные рассуждения.

Укажем классы графов, для которых на данный момент не найдены методы перечисления с точностью до изоморфизма, основанные на теории Пойа: графы, свободные от заданного подграфа, в частности графы без треугольников; (p, q)-разреженные графы на n вершинах (т. е. графы, в которых индуцированный подграф на m вершинах

содержит не более $pm - q$ рёбер); частично ориентированные k-однородные гиперграфы.

Если некоторая задача о перечислении известного класса графов (например, графов без треугольников) в действительности не может быть решена при помощи метода итеративного вычисления производящих функций, то доказательство неразрешимости, вероятно, также представляет значительный интерес, но этот вопрос находится далеко за рамками настоящей работы.

ЛИТЕРАТУРА

1. *McKay B. D. and Piperno A.* Practical graph isomorphism. II // J. Symbolic Comput. 2014. V. 60. P. 94–112.
2. *Харари Ф., Палмер Э.* Перечисление графов. М.: Мир. 1977. 328 с.
3. *Fowler T., Gessel I. M., Labelle G., and Leroux P.* The specification of 2-trees // Adv. Appl. Math. 2002. V. 28. No. 2. P. 145–168.
4. *Labelle G., Lamathe C., and Leroux P.* Labelled and unlabelled enumeration of k-gonal 2-trees // J. Combinat. Theory. Ser. A. 2004. V. 106. No. 2. P. 193–219.
5. *Gainer-Dewar A. and Gessel I. M.* Counting unlabeled k-trees // J. Combinat. Theory. Ser. A. 2014. V. 126. P. 177–193.
6. *Qian J.* Enumeration of unlabeled directed hypergraphs // Electronic J. Combinatorics. 2013. V. 20. Iss. 1. Article no. P46.
7. *Qian J.* Enumeration of unlabeled uniform hypergraphs // Discrete Math. 2014. V. 326. P. 66–74.
8. *Pólya G. and Read R. C. H.* Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds. Berlin; Heidelberg: Springer, 1987.
9. *Ландо С. К.* Лекции о производящих функциях. М.: МЦМНО, 2007. 144 с.
10. `https://github.com/VVerdenko/oriented-2-trees` — Вычисление производящих функций для 3-ориентированных и частично 3-ориентированных 2-деревьев. 2025.

REFERENCES

1. *McKay B. D. and Piperno A.* Practical graph isomorphism, II. J. Symbolic Comput., 2014, vol. 60, pp. 94–112.
2. *Harary F. and Palmer E. M.* Graphical Enumeration. N.Y.: Academic Press, 1973. 272 p.
3. *Fowler T., Gessel I. M., Labelle G., and Leroux P.* The specification of 2-trees. Adv. Appl. Math., 2002, vol. 28, no. 2, pp. 145–168.
4. *Labelle G., Lamathe C., and Leroux P.* Labelled and unlabelled enumeration of k-gonal 2-trees. J. Combinat. Theory, Ser. A, 2004, vol. 106, no. 2, pp. 193–219.
5. *Gainer-Dewar A. and Gessel I. M.* Counting unlabeled k-trees. J. Combinat. Theory, Ser. A, 2014, vol. 126, pp. 177–193.
6. *Qian J.* Enumeration of unlabeled directed hypergraphs. Electronic J. Combinatorics, 2013, vol. 20, iss. 1, article no. P46.
7. *Qian J.* Enumeration of unlabeled uniform hypergraphs. Discrete Math., 2014, vol. 326, pp. 66–74.
8. *Pólya G. and Read R. C. H.* Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds. Berlin; Heidelberg, Springer, 1987.
9. *Lando S. K.* Lectures on Generating Functions. Providence, AMS, 2003.
10. `https://github.com/VVerdenko/oriented-2-trees` — Computation of generating functions for 3-oriented and partially 3-oriented 2-trees. 2025.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.8 DOI 10.17223/20710410/71/6

СЛОЖНОСТЬ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С КРИТЕРИЕМ ОПТИМИЗАЦИИ ЭКОНОМИЧЕСКОГО ЭФФЕКТА ОТ ИСПОЛЬЗОВАНИЯ КВОТ НА ВЫБРОСЫ[1]

А. М. Булавчук

Сибирский федеральный университет, г. Красноярск, Россия

E-mail: abulavchuk@sfu-kras.ru

Рассматривается модель задачи календарного планирования с оптимизацией экономического эффекта от использования квот на выбросы. При построении модели учитывается актуальная российская практика обращения с углеродными единицами. Модель предполагает поиск оптимального расписания инвестиционного проекта, при реализации которого используются углеродные квоты. Критерием оптимальности выступает разница между доходами и расходами, обусловленными операциями с углеродными единицами. Ограничениями задачи выступают организационные и технологические взаимосвязи между работами, а также предельный срок завершения проекта. Все параметры модели являются детерминированными. Сформулирована и доказана теорема о том, что задача календарного планирования с критерием оптимизации экономического эффекта от использования квот на выбросы является NP-трудной в сильном смысле даже в случае работ единичной длительности. Взаимосвязи между проектными работами могут быть представлены в виде неориентированного графа. Это позволяет в процессе доказательства использовать сведение к данной задаче задачи о клике, NP-трудность которой известна. Рассмотрены также особые случаи соотношений между ценами квот и штрафами. Доказано утверждение о том, что при условии равенства и постоянной величине цен и штрафов в каждый момент времени задача календарного планирования с оптимизацией экономического эффекта от использования квот на выбросы может быть сведена к известному варианту задачи календарного планирования с неограниченными ресурсами и критерием максимизации NPV. Для решения численных примеров использована модификация разработанной ранее программы для IBM ILOG CPLEX.

Ключевые слова: *задача календарного планирования инвестиционных проектов, углеродные квоты, задача о клике, NP-трудность.*

[1]Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ (соглашение № 075-02-2025-1790).

COMPLEXITY OF THE SCHEDULING PROBLEM WITH THE CRITERION OF OPTIMISATION OF ECONOMIC EFFECT FROM THE USE OF EMISSION QUOTAS

A. M. Bulavchuk

Siberian Federal University, Krasnoyarsk, Russia

We consider a scheduling model that optimizes the economic impact of using emission quotas. When constructing the model, the current Russian practice of handling carbon units is taken into account. The model involves searching for the optimal schedule of an investment project, the implementation of which uses carbon quotas. The optimality criterion is the difference between income and expenses resulting from transactions with carbon units. The problem constraints include the organizational and technological dependencies between activities, as well as the project deadline. All model parameters are deterministic. We prove that the scheduling problem with the objective of optimizing the economic impact of emission quotas is strongly NP-hard, even when all tasks have the same duration. We show that the relationships between project activities can be represented in the form of an undirected graph. This allowed us to use a reduction of the clique problem, whose NP-hardness is known, to this problem during the proof process. Special cases of relationships between quota prices and fines are also considered. It was proved that, assuming constant prices and fines over time, the problem of scheduling to optimize the economic effect of emission quotas can be reduced to the well-known scheduling problem with unlimited resources and the NPV maximization criterion. To solve numerical examples, a modification of the previously developed algorithm for IBM ILOG CPLEX was used.

Keywords: *project scheduling problem, carbon quotas, clique problem, NP-hardness.*

Введение

Сокращение выбросов парниковых газов — важная практическая задача, один из путей решения которой — введение практики оборота углеродных единиц. Правительством Российской Федерации в последние годы предприняты важные шаги в этом направлении, первым из которых стало принятие Федерального закона № 296-ФЗ «Об ограничении выбросов парниковых газов» [1]. Данный документ закрепил ключевые показатели обращения с углеродными единицами и создал основу для моделирования операций с квотами. Начатый в 2022 г. на территории Сахалинской области эксперимент должен стать основой для расширения практики обращения с углеродными единицами [2]. Таким образом, моделирование обращения с углеродными единицами является актуальным направлением в сфере календарного планирования инвестиционных проектов.

В работе [3] сформулирована модель задачи календарного планирования инвестиционных проектов с критерием оптимизации экономического эффекта от использования квот на выбросы. Данная задача является новой в ряду задач календарного планирования с экономическими критериями оптимальности. Предшествующие исследования акцентировали внимание на оптимизации сроков реализации проектов и чистой приведённой стоимости. Например, в [4] рассматривается задача календарного планирования со складируемыми ресурсами и директивными сроками. Авторы приходят к выводу о существовании варианта этой задачи, который разрешим за полиномиальное время.

В работе [5] также рассматривается задача со складируемыми ресурсами, но с критерием чистой приведённой стоимости. Авторы доказывают, что такая задача является NP-трудной в сильном смысле даже при работах единичной длительности. Другой вариант схожей задачи, предполагающий использование кредитов, рассмотрен в [6]. Показана NP-трудность задачи максимизации прибыли для случая, когда размер используемого кредита не ограничен. Эффективно разрешимые случаи задачи календарного планирования с переменной интенсивностью потребления и поступления ресурсов нескладируемого типа рассматриваются в работе [7], где доказано, что если ширина заданного на множестве работ частичного порядка ограничена константой, то задача псевдополиномиально разрешима, являясь при этом NP-трудной.

Таким образом, изучение вычислительной сложности подобных задач является важным направлением исследований. В данной работе доказано, что задача календарного планирования инвестиционного проекта с квотами на объёмы выбросов углекислого газа также является NP-трудной в сильном смысле. Кроме того, показано, что при некоторых условиях на значения параметров задача может быть сведена к задаче календарного планирования инвестиционных проектов с критерием максимизации чистой приведённой стоимости (NPV).

1. Постановка задачи

Рассмотрим инвестиционный проект, в ходе которого требуется выполнить N работ. Каждая работа характеризуется длительностью в целых периодах d_j, $j \in \{1, 2, \ldots, N\}$. Технологические и организационные взаимосвязи между работами можно задать в виде условий частичного порядка E. Будем называть расписанием проекта вектор $S = (s_1, s_2, \ldots, s_N)$, компоненты которого s_j определяют моменты начала работы $j \in \{1, 2, \ldots, N\}$. Для каждой пары работ $(i, j) \in E$ должно выполняться условие $s_i + d_i \leqslant s_j$. Если срок реализации проекта составляет T периодов, то $0 \leqslant s_j \leqslant T - d_j$.

Пусть при выполнении работы j в период её реализации $\tau \in \{1, \ldots, d_j\}$ выбрасываются парниковые газы в объёме $g_j(\tau)$. Будем считать эти параметры детерминированными, хотя модель может быть модифицирована для стохастического и нечёткого случаев [3].

Ресурсами проекта выступают квоты на выбросы $q(t)$, выделяемые в период $t \in \in \{1, 2, \ldots, T\}$. Единицей выполнения квоты $e(t)$ будем называть разницу между квотой и суммарными выбросами от всех работ, реализуемых в период t. Формула расчёта единицы выполнения квоты имеет вид

$$e(t) = q(t) - \sum_{j \in V(t)} g_j(t - s_j),$$

где $V(t)$ — множество работ, выполняемых в интервале $[t-1, t)$. Если в какой-либо момент времени $e(t) < 0$, то в конце периода t вносится плата за превышение квоты $h(t)$. Положительное значение $e(t)$ соответствует неизрасходованной части квоты. Неизрасходованные единицы выполнения квоты могут быть реализованы по цене $p(t)$.

Таким образом, задача календарного планирования инвестиционного проекта с использованием квот на выбросы заключается в нахождении расписания S, максимизирующего экономический эффект $R(S)$ от операций с единицами выполнения квоты. Ограничениями задачи выступают условия частичного порядка на множестве работ и предельный срок реализации проекта. Математическая модель задачи имеет вид

$$R(S) = \sum_{t \in t^+} \frac{e(t)p(t)}{(1 + r_0)^t} + \sum_{t \in t^-} \frac{e(t)h(t)}{(1 + r_0)^t} \to \max_S; \tag{1}$$

$$s_i + d_i \leqslant s_j, \quad (i,j) \in E; \tag{2}$$

$$0 \leqslant s_j \leqslant T - d_j, \quad j \in \{1, 2, \ldots, N\}, \tag{3}$$

где r_0 — ставка дисконтирования для одного периода; t^+ — множество моментов времени, в которые $e(t) \geqslant 0$; t^- — множество моментов времени, в которые $e(t) < 0$.

2. Сложность задачи

Рассмотрим проект, состоящий из работ единичной длительности, связанных условиями частичного порядка E. Пусть цена продажи единицы квоты в каждый момент времени равна p, а штраф за превышение квоты в каждый момент времени равен

$$h > p(1 + r_0)^2. \tag{4}$$

Величину выбросов для каждой работы примем равной 1. Для известных квот $q(t)$ требуется определить, существует ли расписание со значением экономического эффекта не менее некоторой величины R^*.

Воспользуемся приведенной в [8] идеей сведения к данной задаче задачи о клике.

Задача о клике

Условие: имеется неориентированный граф (W, U) и дано натуральное число k.

Вопрос: существует ли в нём полный подграф с числом вершин не меньше k?

Задача о клике является NP-трудной в сильном смысле [9].

На основе графа (W, U) построим проект, состоящий из $|U| + |W|$ работ. Работы u_{ij}, соответствующие рёбрам $(i,j) \in U$, будем называть рёберными, а работы w_j, соответствующие вершинам $j \in W$, — вершинными. Рёберную работу u_{ij} будем считать предшествующей работам w_i и w_j. Все работы имеют единичную длительность и характеризуются единичными выбросами. Пусть общий срок реализации проекта составляет 3 года, а квоты на выбросы распределены по годам следующим образом:

$$q(1) = |U| - k(k-1)/2; \tag{5}$$

$$q(2) = |W| - k + k(k-1)/2; \tag{6}$$

$$q(3) = k. \tag{7}$$

Суммарная квота равна $|U| + |W|$ и совпадает с совокупной потребностью всех работ проекта. Отсюда следует, что максимальный размер экономического эффекта в данной постановке не может превышать нуля. Из условия (4) следует, что продажа неизрасходованных квот невыгодна, поскольку, с учётом реинвестирования, каждая проданная квота может через два года принести $p(1 + r_0)^2$, а убыток h из-за нехватки квот эту величину превышает. Таким образом, для достижения максимума экономического эффекта необходимо полностью расходовать квоты в каждом году.

Лемма 1. Пусть проект из $|U| + |W|$ работ построен на основе графа (W, U) и удовлетворяет условиям (5)–(7). В графе (W, U) существует клика размера k тогда и только тогда, когда для задачи (1)–(3) существует расписание работ проекта, для которого размер экономического эффекта равен $R^* = 0$.

Доказательство. Пусть известно, что какие-то k вершинных работ образуют клику. Этой клике соответствуют $k(k-1)/2$ рёберных работ. В первый год квота позволяет выполнить $|U| - k(k-1)/2$ рёберных работ. Во второй год можно выполнить оставшиеся рёберные работы и все вершинные, которые не входят в клику. Связывающие их ребёрные работы были выполнены в первый год. Поскольку после второго года остаются невыполненными только вершинные работы, входящие в клику, ресурсов

квоты в размере k в третьем году будет достаточно для завершения проекта. Таким образом, наличие клики позволяет в каждом году израсходовать квоты и получить $R^* = 0$.

Пусть известно, что $R^* = 0$. Отсюда следует, что квоты каждого года израсходованы полностью. После первого года остались невыполненными $k(k-1)/2$ рёберных работ. Предположим, что максимальная клика в графе имеет размерность $m < k$. Для того чтобы минимизировать число вершинных работ, для которых не выполнены соответствующие рёберные, нужно оставить невыполненными рёберные работы, связывающие вершины клики. Оставшиеся $k(k-1)/2 - m(m-1)/2$ работ, выполнение которых откладывается до второго года, будем выбирать таким образом, чтобы они связывали какие-то $k - m$ вершин друг с другом и с вершинами клики. Из предположения о размере максимальной клики $m < k$ следует, что хотя бы одна рёберная работа будет выбрана не по этому правилу. В противном случае это означало бы наличие клики размера k. Тогда после первого года не будут выполнены рёберные работы, соответствующие, по меньшей мере, $k + 1$ вершинной. Дополнительная работа, для которой не выполнена рёберная, не может быть выполнена во втором году, что приведёт к неполному использованию квоты. Однако при неполном использовании квоты $R^* < 0$. Возникшее противоречие разрешается при $m = k$. ■

Теорема 1. Задача календарного планирования (1)–(3) с критерием оптимизации экономического эффекта от использования квот на выбросы является NP-трудной в сильном смысле даже в случае работ единичной длительности.

Доказательство. Сведение задачи о клике к задаче (1)–(3) является полиномиальным, так как для его выполнения требуется не более $\mathcal{O}\left((|U| + |W|)^2\right)$ операций. Из леммы 1 следует, что нахождение оптимального расписания означает решение задачи о клике. Однако задача о клике является NP-трудной в сильном смысле. Отсюда следует, что задача (1)–(3) тоже является NP-трудной в сильном смысле. ■

Пример 1. Проиллюстрируем приведённые в доказательстве рассуждения на примере условного проекта. На рис. 1 приведены граф и сетевой график проекта из восьми работ, $|U| = 4$, $|W| = 4$. Рассмотрим случай $k = 2$. В табл. 1 приведено распределение квот, а также оптимальное расписание проекта. Экономический эффект для этого расписания $R^* = 0$.

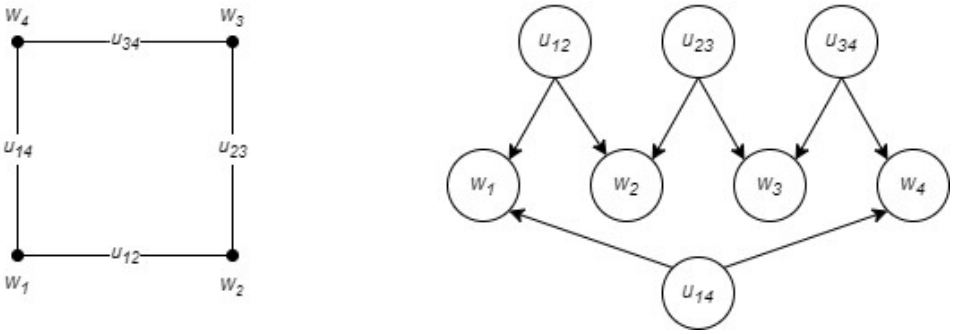

Рис. 1. Граф и сетевой график проекта, $N = 8$

Оптимальное расписание для случая $k = 3$ представлено в табл. 2. Поскольку в графе отсутствует клика размера 3, оптимальный экономический эффект отрицателен. Это означает, что при заданных условиях обращение с квотами будет убыточным, что, однако, не препятствует реализации проекта.

Таблица 1

Оптимальное расписание проекта
$(N = 8,\ k = 2)$

t	1	2	3
Работы	u_{12}, u_{23}, u_{34}	u_{14}, w_2, w_3	w_1, w_4
$q(t)$	3	3	2
$e(t)$	0	0	0

Если $p = 1$, $h = 2$, $r_0 = 0{,}1$, то $R^* = \dfrac{1}{(1+0{,}1)^2} - \dfrac{2}{(1+0{,}1)^3} = -0{,}676.$

Таблица 2

Оптимальное расписание проекта
$(N = 8,\ k = 3)$

t	1	2	3
Работы	u_{12}	u_{23}, u_{34}, u_{14}	w_1, w_2, w_3, w_4
$q(t)$	1	4	3
$e(t)$	0	1	-1

Добавим в граф ребро u_{13} (рис. 2). Для данного проекта $|U| = 5$, $|W| = 4$. Поскольку в получившемся графе имеется клика размера 3, то есть и расписание, для которого $R^* = 0$ (табл. 3).

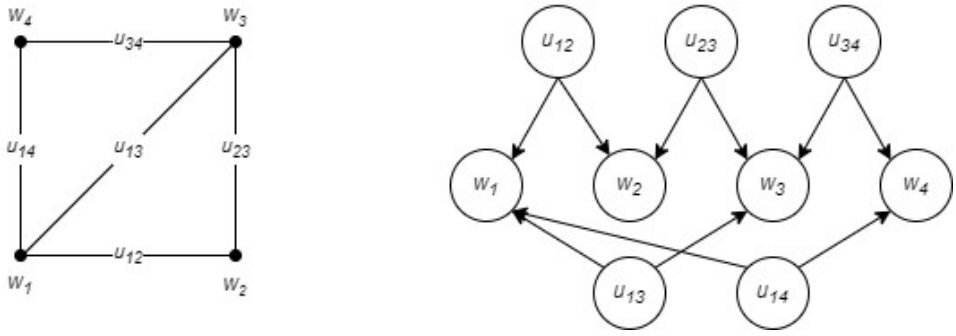

Рис. 2. Граф и сетевой график проекта, $N = 9$

Таблица 3

Оптимальное расписание проекта
$(N = 9,\ k = 3)$

t	1	2	3
Работы	u_{12}, u_{23}	u_{13}, u_{34}, u_{14}, w_2	w_1, w_3, w_4
$q(t)$	2	4	3
$e(t)$	0	0	0

3. Особые случаи

При доказательстве NP-трудности рассматриваемой задачи мы исходили из предположения, что штраф за превышение квоты существенно превышает цену её продажи. Рассмотрим случаи, когда это предположение нарушается.

Пусть в каждый момент времени выполняется равенство $p(t) = h(t)$. Тогда целевая функция задачи примет вид

$$R(S) = \sum_{t=1}^{T} \frac{e(t)p(t)}{(1+r_0)^t}.$$

С практической точки зрения данный вариант описывает ситуацию, когда при недостатке квот они могут быть приобретены на рынке, а цены покупки и продажи совпадают. Сравним этот случай с задачей календарного планирования инвестиционных проектов с критерием максимизации NPV [10]. Введём фиктивную работу, длительность которой совпадает с предельным сроком реализации проекта T, а доход от её реализации в каждый момент времени равен $p(t)q(t)$. Данная работа позволит учитывать доходы от реализации квот, если в какой-либо момент времени работы по проекту не осуществляются. Выбросы парниковых газов можно рассматривать как упущенную выгоду от реализации квот.

3.1. Случай $p(t) = h(t) = \text{const}$

Если $p(t) = p = \text{const}$, то каждая из работ характеризуется потоком платежей с компонентами $p \cdot g_j(\tau)$. Величина $p\big(q(t) - \sum\limits_{j \in V(t)} g_j(t - s_j)\big)$ в каждый момент времени может рассматриваться как сумма компонентов потоков платежей $\sum\limits_{j \in V(t)} c_j(t - s_j)$. Тогда для случая $p(t) = h(t) = p$ целевые функции сравниваемых задач идентичны. Совпадают также ограничения на порядок работ и предельный срок завершения проекта. Однако в задаче (1)–(3) отсутствуют бюджетные ограничения. Такой вариант носит название задачи календарного планирования с неограниченными ресурсами и критерием максимизации NPV. Для этой задачи предложены рекурсивные алгоритмы нахождения оптимального решения [11, 12], при этом вопрос о её полиномиальной разрешимости остаётся открытым. Сложность этого варианта требует дополнительных исследований. Рекурсивные алгоритмы могут быть как полиномиальными, так и экспоненциальными [13]. Таким образом, верно следующее

Утверждение 1. При условии равенства и постоянной величине цен и штрафов в каждый момент времени задача календарного планирования с оптимизацией экономического эффекта от использования квот на выбросы может быть сведена к известному варианту задачи календарного планирования с неограниченными ресурсами и критерием максимизации NPV.

Поскольку данная постановка задачи сводится к задаче календарного планирования, для решения численных примеров воспользуемся разработанной ранее программой для решателя IBM ILOG CPLEX [10]. Модель задачи, для решения которой применяется программа, имеет вид

$$R(X) = \sum_{t=1}^{T} \sum_{j=1}^{N} z_{tj} \cdot x_{tj} \to \max_X,$$

$$\sum_{t=1}^{T} x_{tj} \cdot t \leqslant T - d_j, \qquad j = 1, \ldots, N,$$

$$\sum_{t=1}^{T} (x_{ti} - x_{tj})\, t \leqslant -d_i, \quad (i,j) \in E,$$

где X — матрица, элементы которой $x_{tj} = 1$, если $s_j = t$, и $x_{tj} = 0$ в противном случае, $j \in \{1, \ldots, N\}$, $t \in \{1, \ldots, T\}$. Коэффициенты целевой функции рассчитываются по формуле

$$z_{tj} = \sum_{\tau=0}^{d_j-1} \frac{c_j(\tau)}{(1+r_0)^{\tau+t}},$$

где $t \in \{1, \ldots, T\}$ — момент начала работы j; $c_j(\tau) = p(\tau+1)\, q(\tau+1)$ — для фиктивной работы и $c_j(\tau) = -p(t+\tau)\, g_j(\tau+1)$ — для остальных.

Пример 2. Рассмотрим пример проекта, сетевой график которого приведён на рис. 2. Пусть $p(t) = h(t) = 1$. Введём фиктивную работу f, доход от которой в каждый момент времени равен стоимости продажи имеющихся квот. Эта работа выполняется в течение всего срока реализации проекта и не связана условиями частичного порядка с другими работами.

Результаты расчётов приведены в табл. 4. Оптимальный экономический эффект от использования квот составляет $R^* = 0{,}24$. Решение, приведённое в табл. 3, также является допустимым, но в новых условиях нулевой баланс квот не обеспечивает оптимального экономического эффекта. Доходы от продажи квот в первом периоде компенсируют штрафы в последующих.

Таблица 4

Оптимальное расписание проекта
($N = 9$, $p(t) = h(t) = 1$)

t	1	2	3
Работы	f	u_{12}, u_{13} u_{14}, u_{23}, u_{24}	w_1, w_2, w_3, w_4
$q(t)$	2	4	3
$e(t)$	2	-1	-1

3.2. Случай $p(t) = h(t) \neq \text{const}$

Если в любой момент времени $p(t) = h(t) \neq \text{const}$, то задача не сводится к задаче календарного планирования с критерием максимизации NPV. При переходе к денежным оценкам от натуральных мы будем получать разные потоки платежей для различных моментов начала работы. Однако имеющийся алгоритм легко адаптируется для этого случая. Модификации подвергаются только коэффициенты целевой функции.

Пример 3. Рассмотрим случай растущих цен. Пусть $p(1) = 1$, $p(2) = 1{,}2$ и $p(3) = 1{,}5$. В табл. 5 приведено оптимальное расписание проекта. Экономический эффект для этого расписания составляет $R^* = 0{,}654$. Растущие цены, а значит, и штрафы вынуждают производителя начинать проект как можно раньше и продавать подорожавшие квоты в последний год. Если цены каждый год снижаются, то оптимальное расписание будет совпадать с приведённым в табл. 4.

Таблица 5

Оптимальное расписание проекта
($N = 9$, $p(t) = h(t) \neq \text{const}$)

t	1	2	3
Работы	f, u_{12}, u_{13} u_{14}, u_{23}, u_{24}	w_1, w_2, w_3, w_4	–
$q(t)$	2	4	3
$e(t)$	-3	0	3

3.3. Случай $p(t) > h(t)$

Пусть в каждый момент времени выполняется неравенство $p(t) > h(t)$. Очевидно, что в этом случае целосообразной будет продажа всех имеющихся квот. Представим экономический эффект в виде разницы $R(S) = R_q - R_g(S)$, где R_q — доход от продажи квот; $R_g(S)$ — дисконтированная сумма штрафов за выбросы:

$$R(S) = \sum_{t=1}^{T} \frac{q(t)p(t)}{(1+r_0)^t} - \sum_{t=1}^{T} \frac{h(t) \sum\limits_{j \in V(t)} g_j(t - s_j)}{(1+r_0)^t}.$$

Поскольку первое слагаемое не зависит от выбранного расписания, решение задачи сводится к нахождению расписания, минимизирующего штрафы. При $h(t) = h = \text{const}$ снова получаем задачу календарного планирования с неограниченными ресурсами и критерием максимизации NPV.

Заключение

Таким образом, показано, что задача календарного планирования инвестиционных проектов с критерием оптимизации экономического эффекта от использования квот на выбросы является NP-трудной в сильном смысле. Это, в частности, означает, что для её решения необходимо использовать эвристические алгоритмы. Например, под условия задачи могут быть адаптированы генетический алгоритм и алгоритм имитации отжига, разработанные для схожей задачи [10]. Для нескольких примеров показано, что точные решения рассматриваемой задачи могут быть найдены с помощью IBM ILOG CPLEX.

Автор выражает благодарность рецензенту за полезные замечания и рекомендации, позволившие улучшить содержание работы, а также благодарит Дарью Владиславовну Семенову за помощь и критику.

ЛИТЕРАТУРА

1. Федеральный закон № 296-ФЗ «Об ограничении выбросов парниковых газов», 02.07.2021.
2. Федеральный закон № 34-ФЗ «О проведении эксперимента по ограничению выбросов парниковых газов в отдельных субъектах Российской Федерации», 06.03.2022.
3. *Булавчук А. М., Семенова Д. В.* О задаче календарного планирования с критерием оптимизации экономического эффекта от использования квот на выбросы // УБС. 2025. Вып. 113. С. 215–231.
4. *Гимади Э. Х., Залюбовский В. В., Севастьянов С. В.* Полиномиальная разрешимость задач календарного планирования со складируемыми ресурсами и директивными сроками // Дискретн. анализ и исслед. опер. Сер. 2. 2000. Т. 7. № 1. С. 9–34.
5. *Сервах В. В., Щербинина Т. А.* О сложности одной задачи календарного планирования со складируемыми ресурсами // Вестн. НГУ. Сер. матем., мех., информ. 2008. Т. 8. Вып. 3. С. 105–112.
6. *Казаковцева Е. А., Сервах В. В.* Сложность задачи календарного планирования с кредитованием // Дискретн. анализ и исслед. опер. 2015. Т. 22. Вып. 4. С. 35–49.
7. *Еремеев А. В., Коваленко Ю. В.* Эффективно разрешимые случаи задачи календарного планирования с переменной интенсивностью потребления и поступления ресурсов нескладируемого типа // Известия Иркутского государственного университета. Серия Математика. 2014. Вып. 9. С. 26–38.
8. *Lenstra J. K., Rinnooy Kan A. H. G., and Brucker P.* Complexity of machine scheduling problems // Ann. Discrete Math. 1977. V. 1. P. 343–362.

9. *Karp R. M.* Reducibility among combinatorial problems // R. E. Miller and J. W. Thatcher (eds.). Complexity of Computer Computations. N.Y.: Plenum Press, 1972. P. 85–103.
10. *Bulavchuk A. M. and Semenova D. V.* Two heuristic algorithms for RCPSP with NPV criterion // Журн. СФУ. Сер. Матем. и физ. 2023. Т. 16. № 5. С. 639–650.
11. *Demeulemeester E., Herroelen W., and Van Dommelen P.* An Optimal Recursive Search Procedure for the Deterministic Unconstrained Max-NPV Project Scheduling Problem. Res. Report 9603. Department of Applied Economics, Katholieke Universiteit Leuven, 1996.
12. *De Reyck B. and Herroelen W.* An Optimal Procedure for the Unconstrained Max-NPV Project Scheduling Problem with Generalized Precedence Relations. Res. Report 9642. Department of Applied Economics, Katholieke Universiteit Leuven, 1996.
13. *Быкова В. В.* Математические методы анализа рекурсивных алгоритмов // Журн. СФУ. Сер. Матем. и физ. 2008. Т. 1. № 3. С. 236–246.

REFERENCES

1. Federal'nyy zakon No. 296-FZ "Ob ogranichenii vybrosov parnikovykh gazov" [Federal Law No. 296-FZ "On Limiting Greenhouse Gas Emissions".] 02.07.2021. (in Russian)
2. Federal'nyy zakon No. 34-FZ "O provedenii eksperimenta po ogranicheniyu vybrosov parnikovykh gazov v otdel'nykh sub"ektakh Rossiyskoy Federatsii" [Federal Law No. 34-FZ "On conducting an experiment to limit greenhouse gas emissions in certain constituent entities of the Russian Federation".] 06.03.2022. (in Russian)
3. *Bulavchuk A. M. and Semenova D. V.* O zadache kalendarnogo planirovaniya s kriteriem optimizatsii ekonomicheskogo effekta ot ispol'zovaniya kvot na vybrosy [On the project scheduling problem with the criterion for optimizing the economic effect from the use of emission quotas]. UBS, 2025, vol. 113, pp. 215–231. (in Russian)
4. *Gimadi E. Kh., Zalyubovskiy V. V., and Sevast'yanov S. V.* Polinomial'naya razreshimost' zadach kalendarnogo planirovaniya so skladiruemymi resursami i direktivnymi srokami [Polynomial solvability of scheduling problems with storable resources and directive deadlines]. Diskretn. Anal. Issled. Oper., Ser. 2, 2000, vol. 7, no. 1, pp. 9–34. (in Russian)
5. *Servakh V. V. and Shcherbinina T. A.* O slozhnosti odnoy zadachi kalendarnogo planirovaniya so skladiruemymi resursami [Complexity of some project scheduling problem with nonrenewable resources]. Vestn. Novosib. Gos. Univ., Ser. Mat. Mekh. Inform., 2008, vol. 8, no. 3, pp. 105–112. (in Russian)
6. *Kazakovtseva E. A. and Servakh V. V.* The complexity of the project scheduling problem with credits. J. Appl. Industr. Math., 2015, vol. 9, no. 4, pp. 489–496.
7. *Eremeev A. V. and Kovalenko Yu. V.* Effektivno razreshimye sluchai zadachi kalendarnogo planirovaniya s peremennoy intensivnost'yu potrebleniya i postupleniya resursov neskladiruemogo tipa [Polynomially solvable cases of the project scheduling problem with changing consumption and supply rates of nonaccumulative resources]. Bulletin of Irkutsk State University. Ser. Math., 2014, vol. 9, pp. 26–38. (in Russian)
8. *Lenstra J. K., Rinnooy Kan A. H. G., and Brucker P.* Complexity of machine scheduling problems. Ann. Discrete Math., 1977, vol. 1, pp. 343–362.
9. *Karp R. M.* Reducibility among combinatorial problems. R. E. Miller and J. W. Thatcher (eds.), Complexity of Computer Computations, N.Y., Plenum Press, 1972, pp. 85–103.
10. *Bulavchuk A. M. and Semenova D. V.* Two heuristic algorithms for RCPSP with NPV criterion // J. Sib. Fed. Univ. Math. Phys., 2023, vol. 16, no. 5, pp. 639–650.
11. *Demeulemeester E., Herroelen W., and Van Dommelen P.* An Optimal Recursive Search Procedure for the Deterministic Unconstrained Max-NPV Project Scheduling Problem. Res. Report 9603, Department of Applied Economics, Katholieke Universiteit Leuven, 1996.

12. *De Reyck B. and Herroelen W.* An Optimal Procedure for the Unconstrained Max-NPV Project Scheduling Problem with Generalized Precedence Relations. Res. Report 9642, Department of Applied Economics, Katholieke Universiteit Leuven, 1996.
13. *Bykova V. V.* Matematicheskie metody analiza rekursivnyh algoritmov [Mathematical methods for the analysis of recursive algorithms]. J. Sib. Fed. Univ. Math. Phys., 2008, vol. 1, no. 3, pp. 236–246. (in Russian)

УДК 004.8 DOI 10.17223/20710410/71/7

НАХОЖДЕНИЕ ПРООБРАЗОВ НЕПОЛНОРАУНДОВОЙ ФУНКЦИИ СЖАТИЯ КРИПТОГРАФИЧЕСКОЙ ХЕШ-ФУНКЦИИ SKEIN-512-256 ПРИ ПОМОЩИ SAT-РЕШАТЕЛЯ[1]

О. С. Заикин

Математический центр НГУ, г. Новосибирск, Россия
ИДСТУ СО РАН, г. Иркутск, Россия

E-mail: oleg.zaikin@icc.ru

Рассматривается криптографическая хеш-функция Skein-512-256, вышедшая в 2010 г. в финал конкурса NIST на новую криптографическую хеш-функцию. Функция сжатия Skein-512-256 обрабатывает 512-битовый блок сообщения в течение 72 раундов, каждый раунд состоит из 12 операций. Предлагается практическая алгебраическая атака нахождения прообраза на неполнораундовые версии функции сжатия Skein-512-256. Соответствующие вычислительные задачи сводятся к экземплярам проблемы булевой выполнимости (SAT). С помощью последовательного SAT-решателя были найдены прообразы функции сжатия Skein-512-256, состоящей из первого раунда и семи первых операций второго раунда. Задействование параллельного SAT-решателя позволило увеличить число операций второго раунда до восьми. Ранее в литературе была предложена практическая атака нахождения прообраза максимум на один раунд функции сжатия Skein-512-256.

Ключевые слова: *криптографическая хеш-функция, Skein, атака нахождения прообраза, алгебраический криптоанализ, SAT.*

PREIMAGE ATTACK ON ROUND-REDUCED SKEIN512-256 COMPRESSION FUNCTION USING SAT SOLVER

O. S. Zaikin

NSU Mathematical Center, Novosibirsk, Russia
ISDCT SB RAS, Irkutsk, Russia

Consider the cryptographic hash function Skein-512-256, which became a finalist of the NIST hash function competition in 2010. Skein-512-256 compression function processes a 512-bit block message block in 72 rounds, 12 operations each. A practical algebraic preimage attack on round-reduced versions of the Skein-512-256 compression function is proposed. The corresponding computational problems are reduced to instances of the Boolean satisfiability problem (SAT). Using sequential SAT solvers, preimages of the first round and the first 7 operations of the second round of the compression function have been found. By applying a parallel SAT solver, the number of the second round's operations has been increased to 8. Earlier, a practical preimage attack on at most 1-round Skein-512-256 compression function was published.

Keywords: *cryptographic hash function, Skein, preimage attack, algebraic cryptanalysis, SAT.*

[1]Работа выполнена при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2025-349.

Введение

Криптографические хеш-фунции широко используются в современном цифровом мире для хеширование паролей, проверки целостности данных, формирования электронных цифровых подписей и эмиссии криптовалюты. Безопасные криптографические хеш-функции должны обладать рядом свойств, среди которых стойкость к нахождению коллизий, прообразов и вторых прообразов [1].

В 2007 г. Национальный институт стандартов и технологий США объявил конкурс на новую криптографическую хеш-функцию. Основной целью конкурса была замена действующего на тот момент стандарта — семейства криптографических хеш-функций SHA-2. В 2012 г. из пяти финалистов был выбран Keccak, и именно он стал новым стандартом под названием SHA-3. С тех пор криптографическая стойкость Keccak была тщательно проанализирована с помощью различных подходов. При этом стойкость остальных финалистов остаётся относительно слабо изученной.

Одним из финалистов конкурса стало семейство Skein [2]. Основной криптографической хеш-функцией семейства является Skein-512-256, в которой функция сжатия базируется на 72-раундовом блочном шифре Threefish, работающем на 512-битовых блоках сообщения. В каждом раунде Threefish к 64-битовым частям внутреннего состояния применяются следующие операции: исключающее или, сложение по модулю 2^{64} и циклический сдвиг. Результатом Skein-512-256 является хеш длиной 256 бит.

Нахождение прообраза первого раунда функции сжатия Skein-512-256 является простой вычислительной задачей — соответствующая практическая алгебраическая атака была представлена в [3]. Под алгебраической атакой понимается атака путём решения систем алгебраических уравнений [4]. Конкретнее, в [3] использован SAT-подход, который является одним из наиболее эффективных на практике способов реализации алгебраических атак.

SAT формулируется следующим образом: по данной пропозициональной булевой формуле ответить на вопрос, существует ли набор значений переменных этой формулы, при которых она истинна [5]. Такой набор называется выполняющим. Программы для решения SAT называются SAT-решателями. При этом булева формула обычно рассматривается в конъюнктивной нормальной форме (КНФ), т. е. в виде конъюнкции дизъюнктов. Дизъюнкт — это дизъюнкция литералов, а литерал — это булева переменная либо её отрицание. Если SAT-решатель может доказать выполнимость КНФ, то, как правило, он выдаёт также и соответствующий выполняющий набор.

В [3] для решения экземпляров SAT использованы SAT-решатели, основанные на полном алгоритме Conflict-Driven Clause Learning (CDCL) [6]. По-видимому, впервые SAT-решатели были применены для анализа стойкости криптографических хеш-функций в работе [7], с тех пор такой подход используется довольно широко.

Отметим, что найти прообраз первых двух раундов функции сжатия Skein-512-256 на текущий момент очень сложно. Каждый раунд этой функции состоит из двенадцати основных операций. В настоящей работе предлагается новая алгебраическая атака нахождения прообраза на неполнораундовые версии функции сжатия Skein-512-256. Конкретнее, проанализированы версии функции сжатия, в которых первый раунд присутствует полностью, во втором раунде оставлены только несколько первых операций, а остальные раунды отброшены. Как и в [3], для этих целей используется SAT-подход. С помощью последовательного CDCL-решателя Kissat найдены прообразы ослабленной функции сжатия Skein-512-256, состоящей из первого раунда и семи операций второго раунда. Задействован метод Cube-and-Conquer, согласно которому SAT-задача разбивается на более простые подзадачи, каждая из которых решается с помощью

CDCL-решателя [8]. Это позволяет увеличить число операций во втором раунде до восьми.

При применении Cube-and-Conquer к описанным задачам обнаружено, что многие подзадачи оказываются очень простыми. Предлагается подход к предварительному препроцессингу КНФ, в рамках которого итеративно запускается Cube-and-Conquer, с помощью ограниченного CDCL решаются только простые подзадачи, а информация о них добавляется в КНФ. Данный подход позволяет найти некоторые прообразы быстрее, а также отыскать прообразы для ряда случаев, которые ранее решению не поддавались.

Работа имеет следующую структуру: в п. 1 описана криптографическая хеш-функция Skein-512-256; п. 2 посвящён SAT-кодировке функции сжатия Skein-512-256; в п. 3 стойкость неполнораундовых версий функции сжатия Skein-512-256 к нахождению прообразов исследуется при помощи CDCL-решателя Kissat и метода Cube-and-Conquer. Кроме того, описан новый подход к препроцессингу КНФ с помощью итеративного применения метода Cube-and-Conquer и приведены соответствующие результаты вычислительных экспериментов.

1. Криптографическая хеш-функция Skein-512-256

Семейство криптографических хеш-функций Skein разработано группой авторов во главе с Брюсом Шнайером [2]. Здесь кратко опишем основную криптографическую хеш-функцию данного семейства, Skein-512-256, которая оперирует с 512-битовыми блоками сообщения и формирует 256-битовый хеш. Отметим, что Skein-512-256 была предложена в качестве замены SHA-256 из семейства SHA-2.

В Skein-512-256 на 512-битовых блоках сообщения итеративно вызывается функция сжатия, которая формирует 512-битовый выход. При этом гораздо более распространённой является ситуация, в которой длина выхода функции сжатия в несколько раз меньше, чем длина входа. Например, в MD5, SHA-1 и SHA-256 длина выхода в 4, 3,2 и 2 раза меньше длины входа соответственно. Данная особенность функции сжатия Skein-512-256 связана с тем, что она базируется на блочном шифре Threefish, который оперирует с 512-битовыми блоками.

Введём следующие термины:

— секретный ключ K длиной 512 бит;
— параметр T длиной 128 бит;
— открытый текст P длиной 512 бит;
— внутреннее состояние S длиной 512 бит;
— шифртекст C длиной 512 бит.

В Threefish выполняется 72-раундовая функция шифрования $C = E(K, T, P)$. Перед первым раундом формируются раундовые ключи, зависящие от секретного ключа K. После этого формируется внутреннее состояние S — восемь 64-битовых слов. Инициализация S осуществляется путём смешивания значений T, P и раундовых ключей. Затем в каждом из 72 раундов модифицируются все восемь слов S. Раз в четыре раунда внутреннее состояние дополнительно смешивается с раундовыми ключами и T.

Опишем раунд более подробно. Сначала формируются четыре пары 64-битовых слов внутреннего состояния, а затем каждая пара обновляется с помощью вызова нелинейной функции MIX. Эта функция получает на вход два 64-битовых слова (x_0, x_1) и вычисляет два 64-битовых слова (y_0, y_1) следующим образом:

$$y_0 = (x_0 + x_1) \bmod 2^{64}, \quad y_1 = (x_1 \lll R) \oplus y_0.$$

Здесь $\lll$ — циклический сдвиг влево; R — константа, которая задана для каждого раунда. В итоге соответствующей паре слов внутреннего состояния присваиваются значения (y_0, y_1).

Вернёмся к функции сжатия Skein-512-256. Её 512-битовый выход вычисляется следующим образом: $E(K, T, P) \oplus P$, где P — это текущий блок сообщения. Таким образом, стойкость функции сжатия полностью зависит от стойкости блочного шифра Threefish.

Для вычисления хеша функция сжатия вызывается в режиме Unique Block Iteration (UBI), который является вариантом режима Matyas — Meyer — Oseas [9]. На первом блоке сообщения на вход функции сжатия в качестве K подаётся набор инициализирующих констант, указанных в стандарте [2], а на всех остальных блоках сообщения — 512-битовый выход функции сжатия на предыдущем блоке сообщения. Также подаётся на вход и 128-битовое значение T, которое содержит различную служебную информацию: сколько байт сообщения уже обработано, является ли текущий блок сообщения первым или последним и т. д. Благодаря режиму UBI функция сжатия является параметризованной, а каждый блок сообщения обрабатывается уникальной версией этой функции. Это сделано для повышения криптографической стойкости хеш-функций семейства Skein. На последнем блоке сообщения функция сжатия вызывается дважды, затем итоговый хеш Skein-512-256 вычисляется взятием 256 старших битов внутреннего состояния.

Лучшая на данный момент теоретическая атака нахождения прообраза на полнораундовую хеш-функцию Skein-512-256 требует перебора $2^{511,76}$ вариантов [10], что лишь немного лучше, чем полный перебор. В [3] представлена практическая атака нахождения прообраза на первый раунд хеш-функции Skein-512-256, а также на первый раунд функции сжатия Skein-512-256.

В настоящей работе исследуется стойкость неполнораундовых версий функции сжатия Skein-512-256 к практическим алгебраическим атакам нахождения прообраза. Как и в [3], для этого применяется SAT-подход. В следующем пункте с этой целью строится SAT-кодировка функции сжатия Skein-512-256.

2. SAT-кодировка нахождения прообраза функции сжатия Skein-512-256

Опишем постановку задачи нахождения прообраза неполнораундовой функции сжатия Skein-512-256, SAT-кодировку этой задачи и семейство КНФ, сгенерированных на её основе.

2.1. Постановка задачи

Рассмотрим функцию сжатия Skein-512-256 при её вызове на первом блоке сообщения. В этом случае 512-битовый секретный ключ K блочного шифра Threefish известен (см. п. 1). Значения двух 64-битовых слов параметра T в случае первого вызова равны 64 и 8070450532247928832 соответственно [2]. Из входных данных остаётся неизвестным только 512-битовый блок сообщения P. Рассматривается следующая задача: по известному 512-битовому выходу неполнораундовой функции сжатия найти неизвестный 512-битовый вход (т. е. блок сообщения). Несмотря на то, что в такой постановке секретный ключ известен, это не делает задачу простой. Действительно, перед первым раундом шифра Threefish внутреннее состояние S инициализируется путём смешивания раундовых ключей (которые известны, так как они зависят от секретного ключа), T и P. Раз P неизвестно, то и S тоже неизвестно.

2.2. Кодирование в SAT с помощью CBMC

На предварительном этапе были проанализированы существующие трансляторы, способные сводить задачи анализа стойкости криптографических хеш-функций к SAT. Зачастую для этих целей используется Transalg [11]. В частности, доступны онлайн построенные с его помощью КНФ, кодирующие задачи нахождения коллизий и прообразов функций сжатия MD4, MD5, SHA-1 и SHA256. При этом Transalg поддерживает только сложение по модулю 2^{32}, а в функции сжатия Skein-512-256 используется сложение по модулю 2^{64}. По этой причине Transalg использован не был.

В результате предварительного анализа был выбран транслятор Bounded Model Checker for C programs (CBMC), предназначенный для проверки свойств программ на языке программирования C [12]. Одним из способов проверки является сведение проверяемого свойства и описанного в программе алгоритма к SAT путём формирования КНФ. Отметим, что CBMC не может закодировать в SAT операции над вещественными числами — все переменные, с которыми оперирует алгоритм, должны быть целочисленными. В построенной с помощью CBMC КНФ часть булевых переменных соответствует входу алгоритма, другая часть — выходу алгоритма, а остальные переменные являются дополнительными и нужны для преобразований, которые вычисляют выход по входу. Если в КНФ присвоить значения только выходным переменным, то сформируется задача поиска входа по известному выходу. При этом достаточно закодировать сам алгоритм, т. е. никакие свойства программы проверять не требуется. Именно в таком режиме CBMC используется далее.

CBMC принимает на вход программу на языке C после предварительной подготовки, которая подробно описана в [13]. Эта подготовка состоит из следующих ключевых этапов:

1) из заголовочных файлов и файлов с исходным кодом сформировать один файл с исходным кодом;
2) если целочисленной переменной `var` присваивается значение `val`, но эта переменная, а также операции над ней должны быть закодированы в КНФ, то добавить в исходный код следующую строку: `__CPROVER_assume(var==val)`.

Отметим, что `__CPROVER_assume(var==val)` — это специальная CBMC-функция, которой нет в составе языка C. При её использовании в КНФ добавляются дизъюнкты, кодирующие равенство `var=val`. Булевы переменные КНФ, кодирующие целочисленную переменную `var`, должны быть уже введены ранее путём объявления переменной `var` (без инициализации) и присваивания ей результата операций над другими переменными программы. Если в исходном коде переменной `var` присвоить значение `val` напрямую, то переменные и дизъюнкты в КНФ для неё добавлены не будут. В случае функции сжатия Skein-512-256 для построения корректной SAT-кодировки специальная функция `__CPROVER_assume()` должна быть применена для присваивания секретному ключу инициализирующих констант, а также для означивания выхода функции сжатия. Используемые значения выхода рассмотрены далее.

Для генерации КНФ взята реализация криптографической хеш-функции Skein на языке C, поданная ранее на конкурс Национального института стандартов и технологий США [2]. Конкретнее — речь о Skein версии 1.3. Затем была выделена реализация функции сжатия Skein-512-256, которая в основном содержится в функции `Skein_512_Process_Block()`. После этого была проведена предварительная подготовка для запуска CBMC, о которой речь шла выше.

2.3. Генерация задач нахождения прообраза

Нахождение прообраза первого раунда функции сжатия Skein-512-256 в описанной постановке является простой задачей для современных SAT-решателей. При этом для первых двух раундов соответствующая задача уже очень сложная. По этой причине были исследованы промежуточные варианты. Напомним, что на каждом раунде внутреннее состояние модифицируется путём четырёх вызовов функции MIX, которая состоит из трёх операций: сложения по модулю 2^{64}, циклического сдвига, сложения по модулю 2. Таким образом, каждый раунд состоит из двенадцати базовых операций. Рассмотрим одиннадцать версий неполнораундовой функции сжатия Skein-512-256, таких, что в i-й функции используется первый раунд и первые i операций второго раунда, где $i \in \{1, \ldots, 11\}$.

Для каждой из одиннадцати функций сжатия анализировалось нахождение прообразов следующих десяти выходов:

— 512 нулевых битов;
— 512 единичных битов;
— восемь 512-битовых последовательностей с регулярной структурой и одинаковым числом нулей и единиц.

Выходы с регулярной структурой были сформированы следующим образом: j-й выход равен последовательности из 2^j единичных битов и 2^j нулевых битов, которая повторялась $512 \cdot 2^{-j-1}$ раз, где $j \in \{1, \ldots, 8\}$. Таким образом формируется периодическая последовательность с периодом $512 \cdot 2^{-j-1}$: первый выход с регулярной структурой равен последовательности 1100, повторённой 128 раз, а последний — 256 единичным и 256 нулевым битам.

Для генерации 110 КНФ использован транслятор CBMC версии 5.95.1, который вызывался с ключом `--dimacs`. Характеристики построенных КНФ приведены в табл. 1. Ключевые характеристики не зависят от значения выхода, поэтому в таблице приведены КНФ только для первого выхода, состоящего из 512 нулевых битов.

Таблица 1

Характеристики КНФ, кодирующих нахождение прообразов нулевого выхода

i	Переменные	Дизъюнкты	Литералы
1	4 925	15 404	41 942
2	4 989	15 532	42 198
3	5 053	15 788	42 966
4	5 181	16 677	45 757
5	5 245	16 805	46 013
6	5 309	17 061	46 781
7	5 437	17 950	49 572
8	5 501	18 078	49 828
9	5 565	18 334	50 596
10	5 693	19 223	53 387
11	5 757	19 351	53 643

В дополнение к данным табл. 1 отметим, что CBMC кодирует каждый вызов функции MIX путём добавления 256 булевых переменных и 1 273 дизъюнктов, эти дизъюнкты содержат 3 815 литералов.

3. Вычислительные эксперименты

Представим результаты применения SAT-решателей к описанным КНФ. Все эксперименты проведены на персональном компьютере, оснащённом 64 гигабайтами оперативной памяти и 16-ядерным процессором AMD Ryzen 3950X.

3.1. Нахождение прообразов при помощи CDCL-решателя

На первом этапе был использован CDCL-решатель Kissat [14], который последние годы несколько раз становился победителем конкурса SAT Competition. Конкретнее — использована версия 4.0.1 этого решателя. Kissat является однопоточной программой, поэтому в рамках экспериментов каждому экземпляру Kissat было выделено одно ядро процессора. На каждой из 110 КНФ Kissat был запущен с лимитом времени 24 ч. При $i = 1$ и 2 (напомним, что i означает количество операций во втором раунде) все SAT-задачи были решены, при этом в среднем решатель работал менее одной секунды. Примерно для половины выходов был найден выполняющий набор, т. е. были найдены прообразы ослабленных версий функции сжатия Skein512-256. Для оставшейся половины выходов решателем была доказана невыполнимость КНФ, т. е. доказано, что у этих выходов нет прообразов для данных версий функции сжатия.

Для $i = 3$ ситуация поменялась — SAT-задачу для выхода № 10 решатель не смог решить за 24 ч; напомним, что этот выход состоит из 256 единичных и 256 нулевых битов. При этом большинство SAT-задач были решены менее чем за секунду или за несколько секунд. Дла $i = 4$ картина оказалась примерно такой же — не была решена только SAT-задача для выхода № 10. Для $i = 5$ уже две SAT-задачи не были решены. Далее сложность SAT-задач существенно растёт с увеличением i. Для $i = 8$ и всех последующих i ни одна SAT-задача решена не была. В табл. 2 приведены результаты для $i = 4, 5, 6, 7, 8$. Запись «SAT»/«UNSAT» значит, что для соответствующей КНФ была доказана выполнимость/невыполнимость. В первом случае это означает нахождение прообраза, а во втором — доказательство, что для конкретного выхода прообраза не существует. Запись «–» означает, что Kissat не смог решить SAT-задачу за 24 ч. Если SAT-задача решена менее чем за секунду, то указана дробная часть; в остальных случаях время округлено до целого значения.

Таблица 2

Результаты решения SAT-задач с помощью Kissat и время решения, с

№ выхода	$i = 4$		$i = 5$		$i = 6$		$i = 7$		$i = 8$	
1	UNSAT	0,40	UNSAT	0,44	UNSAT	2	UNSAT	4	–	–
2	UNSAT	0,15	UNSAT	0,29	–	–	–	–	–	–
3	SAT	29	UNSAT	0,06	SAT	133	–	–	–	–
4	UNSAT	18	UNSAT	20	UNSAT	67	UNSAT	109	–	–
5	SAT	4	SAT	2	UNSAT	419	SAT	26 066	–	–
6	SAT	1	SAT	0,39	SAT	180	–	–	–	–
7	UNSAT	2	UNSAT	2	UNSAT	21	UNSAT	33	–	–
8	UNSAT	0,15	–	–	–	–	–	–	–	–
9	UNSAT	0,15	UNSAT	840	UNSAT	396	UNSAT	1731	–	–
10	–	–	–	–	–	–	–	–	–	–

Судя по результатам, приведённым в табл. 2, наличие прообраза, а также сложность решения SAT-задачи существенно зависит от выхода. Для выхода № 1 (т. е. для 512 нулевых битов) доказано отсутствие прообраза в четырёх случаях. Кроме того, SAT-задачи для этого выхода оказались очень простыми. SAT-задачи для выхода № 10 оказались самыми сложными — решатель не смог решить ни одну из них. Наибольшее

количество прообразов найдено для выходов № 5 и 6, а время решения этих SAT-задач почти во всех случаях существенно больше, чем для выхода № 1. Из представленных результатов следует, что рассмотренные версии функции сжатия при $1 \leqslant i \leqslant 7$ (из $\{0,1\}^{512}$ в $\{0,1\}^{512}$) не биективны, так как они не сюръективны. Можно выдвинуть гипотезу, что и полнораундовая функция сжатия Skein-512-256 не биективна.

3.2. Нахождение прообразов при помощи Cube-and-Conquer

Одним из самых эффективных подходов к решению трудных SAT-задач является Cube-and-Conuqer [8]. Согласно этому подходу, сначала на КНФ запускается lookahead-эвристика, которая осуществляет декомпозицию задачи на более простые подзадачи. При этом lookahead строит двоичное дерево, в котором узлы соответствуют переменным КНФ, а рёбра — значениям переменных. Есть два типа листьев такого дерева — отсечённые листья, для которых lookahead самостоятельно доказал отсутствие решений, и прерванные листья. Каждая из подзадач формируется путём подстановки в КНФ значений всех переменных от корня дерева до соответствующего прерванного листа. Набор этих значений называется кубом, а процедура декомпозиции — кубированием.

На этапе кубирования ключевым является значение порога прерывания n. Прерванный лист формируется в том случае, когда при подстановке текущего куба в КНФ и простейшего препроцессинга (итеративного применения правила единичного дизъюнкта) в модифицированной КНФ оказывается меньше n переменных. На втором этапе метода Cube-and-Conuqer на каждой подзадаче, соответствующей прерванному листу, запускается CDCL-решатель. Если исходная КНФ выполнима, то в результате решения хотя бы одной подзадачи будет найден выполняющий набор. Если же исходная КНФ невыполнима, то на всех подзадачах будет доказана невыполнимость. Все подзадачи можно решать независимо друг от друга, поэтому Cube-and-Conuqer хорошо подходит для решения трудных SAT-задач в параллельных вычислительных системах.

С помощью Cube-and-Conuqer был впервые решён ряд трудных комбинаторных задач. Самой известной из них является булева проблема пифагоровых троек [15]. В [16] предложен параллельный SAT-решатель EnCnC, реализующий метод Cube-and-Conuqer для анализа стойкости криптографических примитивов с учётом ряда особенностей этой предметной области. С помощью этого решателя были впервые найдены прообразы 43-шаговой функции сжатия MD4, 29-раундовой функции сжатия MD5 и 24-раундовой функции сжатия SHA-1 [16, 17].

Кратко рассмотрим принцип работы EnCnC в режиме полного SAT-решателя. В качестве начального значения n выбирается число переменных в КНФ после препроцессинга. Затем n уменьшается с некоторым шагом и для каждого из новых значений с помощью lookahead-решателя генерируются все подзадачи, при этом отслеживается число отсечённых листьев. Процесс заканчивается, когда на очередном n число подзадач превышает заданную верхнюю границу. Затем формируется набор значений n, таких, что число соответствующих подзадач находится в заданном диапазоне (т. е. учитывается не только верхняя, но и нижняя граница), а число отсечённых листьев не меньше заданного лимита. На следующем этапе для каждого выбранного таким образом n формируется случайная выборка подзадач. Подзадачи из выборки решаются с помощью CDCL-решателя с некоторым лимитом времени. Для выборок, в которых все подзадачи решены, вычисляется прогнозное время решения всех подзадач. Прогнозное время вычисляется путём умножения среднего времени решения подзадач из выборки на общее число подзадач. В итоге выбирается лучшее значение n с точки

зрения прогнозного времени и запускается решение всех оставшихся нерешёнными подзадач, сформированных на этом n.

SAT-решатель EnCnC был запущен на четырёх КНФ, которые соответствуют $i = 8, 9$ и выходам № 5 и 6. Эти выходы выбраны по той причине, что на них чаще всего находились прообразы при меньших значениях i. Напомним, что эти выходы равны последовательности 1111111100000000, повторённой 32 раза, и последовательности 11111111111111110000000000000000, повторённой 16 раз, соответственно; при $i = 8$ рассматривается задача нахождения прообраза функции сжатия Skein512-256, которая состоит из первого раунда и 8 первых операций (из 12) второго раунда. Для $i = 9$ задача определяется аналогично. Вычислительные эксперименты проводились на том же компьютере с 16-ядерным процессором. Использованы следующие значения входных параметров EnCnC:

— -nstep = 5 — значение n уменьшается с шагом 5;
— -minc = 1000 — выбираются только такие n, для которых формируется минимум 1 000 подзадач;
— -maxc = 20000 — выбираются только такие n, для которых формируется максимум 20 000 подзадач;
— -minref = 1 — выбираются только такие n, для которых отсекается минимум 1 лист;
— -sample = 100 — 100 подзадач в случайной выборке для каждого выбранного n;
— -lasolver = march_cu — lookahead-решатель march_cu;
— -cdclsolvers = kissat4.0.1 — CDCL-решатель Kissat версии 4.0.1;
— -maxcdcltime = 5000 — CDCL-решатель работает с лимитом времени 5 000 с.

Результаты этапа прогнозирования для $i = 8$ представлены в табл. 3. Прогнозное время указано в сутках при условии использования 16 ядер; приведены только выбранные на предварительном этапе значения n. Для обеих КНФ начальное значение n равно 2 465.

Таблица 3

Результаты этапа прогнозирования для КНФ при $i = 8$

n	Подзадач	Отсечённых листьев	Прогнозное время решения
Выход № 5			
2 355	1 431	1	8 ч 1 мин
2 350	4 619	11	2 дн. 2 ч 54 мин
2 345	7 876	23	2 дн. 8 ч 44 мин
2 340	13 262	42	3 дн. 6 мин
Выход № 6			
2 355	2 567	1	2 дн. 7 ч 18 мин
2 345	7 916	3	4 дн. 5 ч 5 мин
2 340	6 141	4	2 дн. 17 ч 53 мин
2 330	16 096	19	5 дн. 4 ч 2 мин

Согласно табл. 3, лучшие прогнозные значения для обеих КНФ найдены на $n = 2355$, поэтому именно этот порог прерывания был использован на этапе решения. Все подзадачи для выхода № 5 были решены за 8 ч 46 мин на 16 ядрах. Оказалось, что все соответствующие КНФ невыполнимы, т. е. доказано, что прообразов для выхода № 5 при $i = 8$ не существует. В случае выхода № 6 выполняющий набор (соответственно и прообраз для данного выхода) на одной из КНФ был найден за 6 ч 7 мин, при этом было решено 15,7 % подзадач.

Результаты прогнозирования для случая $i = 9$ представлены в табл. 4.

Таблица 4

Результаты этапа прогнозирования для КНФ при $i = 9$

n	Подзадач	Отсечённых листьев	Прогнозное время решения
Выход № 5			
2 445	1 879	102	40 с
2 440	2 622	146	44 с
2 435	3 649	216	52 с
2 430	5 054	301	1 м 26 с
2 425	6 938	432	1 м 24 с
2 420	9 343	572	1 м 42 с
2 415	12 622	802	2 м 13 с
2 410	17 020	1078	3 м 19 с
Выход № 6			
2 430	2 321	98	11 с
2 425	3 136	151	11 с
2 420	4 168	200	15 с
2 415	5 448	298	19 с
2 410	7 260	402	25 с
2 405	9 426	559	32 с
2 400	12 034	741	34 с
2 395	15 070	1044	51 с
2 390	18 634	1392	53 с

Прогнозное время для $i = 9$ оказалось необычно маленьким. Это контринтуитивно, так как для $i = 9$ SAT-задачи должны быть сложнее (или, по крайней мере, не проще), чем для $i = 8$. Причина в том, что во всех случаях в выборках подзадачи были очень простыми — Kissat решал их быстрее чем за 1 с. Эти прогнозы проверены в режиме решения. В случае выхода № 5 использовалось значение $n = 2445$; 1 876 подзадач были решены очень быстро (в среднем за 0,5 с), но оставшиеся 3 подзадачи не были решены даже за 24 ч. Для выхода № 6 ситуация повторилась — 2320 подзадач были решены в среднем за 0,1 с при $n = 2430$, но оставшаяся подзадача также не решилась за 24 ч. Таким образом, прогнозное время оказалось неточным из-за того, что в выборки были включены только очень простые подзадачи. Отметим, что ранее при анализе стойкости MD4, MD5 и SHA-1 с помощью EnCnC такой ситуации не возникало.

3.3. Применение Cube-and-Conquer для препроцессинга КНФ

Для решения проблемы с излишне оптимистичными прогнозами в случае $i = 9$ в рамках настоящего исследования был разработан алгоритм итеративного применения Cube-and-Conquer. Цель этого алгоритма состоит в преобразовании КНФ таким образом, чтобы Cube-and-Conquer не формировал по ней слишком простые подзадачи.

Алгоритм принимает на вход следующие данные:

1) КНФ C;
2) число кубов k;
3) lookahead-решатель lasolver;
4) CDCL-решатель cdclsolver;
5) максимальное число конфликтов confl для CDCL-решателя.

Алгоритм состоит из следующих шагов:

Шаг 1. Начиная с порога прерывания n, равного числу переменных в C после препроцессинга по правилу единичного дизъюнкта, уменьшать n и запускать lasolver на C; найти минимальное n такое, что lasolver формирует не более k подзадач на C.

Шаг 2. Запустить на каждой из k подзадач cdclsolver с лимитом в confl конфликтов.

Шаг 3. Пусть cdclsolver решил $s \leqslant k$ подзадач, а на остальных $k - s$ подзадачах он был прерван при достижении confl конфликтов. Возникает пять случаев:

1) $s = 0$: выдать UNKNOWN и модфицированную КНФ C; остановить алгоритм;
2) $s = k$ и на всех s подзадачах cdclsolver доказал невыполнимость: выдать UNSAT и остановить алгоритм;
3) $s > 0$ и хотя бы на одной из s подзадач cdclsolver нашёл выполняющий набор: выдать SAT и остановить алгоритм;
4) $0 < s < k - 1$ и для всех s подзадач cdclsolver доказал невыполнимость: добавить в C s дизъюнктов, являющихся отрицаниями s соответствующих кубов (на которых C оказалась невыполнимой); перейти к шагу 1;
5) $s = k - 1$ и для всех s подзадач cdclsolver доказал невыполнимость: добавить в C s дизъюнктов, являющихся отрицаниями s соответствующих кубов (на которых C оказалась невыполнимой); добавить в C однолитеральные дизъюнкты, состоящие из литералов, входящих в оставшийся куб; перейти к шагу 1.

Рассмотрим пример. Допустим, что $k = 4$ и на первом шаге найдено минимальное n, такое, что lasolver сгенерировал на КНФ C четыре куба. Конкретнее, речь о следующих кубах: $\neg x_3 \wedge x_7$; $\neg x_4 \wedge x_9$; $x_1 \wedge \neg x_3 \wedge x_9$; $x_5 \wedge x_{15} \wedge \neg x_{21}$. Допустим также, что на шаге 2 на подзадачах № 1 и 3 cdclsolver доказал невыполнимость (т. е. доказана невыполнимость КНФ $C \wedge \neg x_3 \wedge x_7$ и $C \wedge x_1 \wedge \neg x_3 \wedge x_9$), а на оставшихся двух подзадачах cdclsolver был прерван. Тогда на шаге 3 возникает четвёртый случай: в C добавляются дизъюнкты $\neg(\neg x_3 \wedge x_7) = (x_3 \vee \neg x_7)$ и $\neg(x_1 \wedge \neg x_3 \wedge x_9) = (\neg x_1 \vee x_3 \vee \neg x_9)$. Иными словами, C заменяется на $C \wedge (x_3 \vee \neg x_7) \wedge (\neg x_1 \vee x_3 \vee \neg x_9)$ и на этой модифицированной КНФ выполняется шаг 1 алгоритма.

Если бы в рассмотренном примере cdclsolver смог на шаге 2 успешно решить ещё и подзадачу № 4, то на шаге 3 возник бы пятый случай и к КНФ C были бы добавлены следующие дизъюнкты: $(x_3 \vee \neg x_7)$; $(\neg x_1 \vee x_3 \vee \neg x_9)$; $(\neg x_5 \vee \neg x_{15} \vee x_{21})$; $(\neg x_4)$; (x_9). Последние два однолитеральных дизъюнкта кодируют тот факт, что куб № 2 является истинным.

Отметим, что предложенный алгоритм идейно схож с добавлением к КНФ дизъюнктов, сгенерированных с помощью лазеек (англ. backdoors) [18]. И в том и в другом случае добавляемые дизъюнкты являются логическим следствием исходной КНФ, а значит, никакие выполняющие наборы (если они есть) таким образом не запрещаются. При этом пятый случай шага 3 особенный: фактически с помощью добавления однолитеральных дизъюнктов осуществляется присваивание значений соответствующих переменных. Это следствие того, что на всех остальных кубах было доказано, что КНФ невыполнима.

Представленный выше алгоритм, выполняющий препроцессинг с помощью Cube-and-Conquer, реализован на языке программирования C++ в виде многопоточной программы. Исходный код программы, а также все используемые в этом исследовании КНФ и программы для CBMC доступны онлайн [19]. В определённых условиях эта многопоточная программа может решить SAT-задачу, но основная её цель состоит в препроцессинге КНФ таким образом, чтобы по ней Cube-and-Conquer не генерировал слишком простые задачи (такие, как в п. 3.2 при $i = 9$). Кроме того, на модифицированной таким образом КНФ SAT-задача может быть решена быстрее, чем до модификации.

Сначала разработанная программа была запущена на компьютере на каждой из одиннадцати SAT-задач, на которых Kissat не смог найти решение за 24 ч при $i \in \{4, 5, 6, 7\}$ (см. п. 3.1). Использовались следующие входные параметры: $k = 16$ (по числу ядер процессора); lookahead-решатель march_cu; CDCL-решатель Kissat; confl = 30 000. При таком ограничении на число конфликтов Kissat работает примерно 0,5 с. Время работы программы и число итераций алгоритма зависело от КНФ: всего потребовалось от 4 до 67 с и от 2 до 12 итераций. На модифицированных КНФ был снова запущен Kissat на 24 ч. В результате была решена одна SAT-задача, а именно — найден прообраз выхода № 10 для $i = 5$ за 5 ч. Приведём подробную статистику препроцессинга на этой КНФ: выполнены 4 итерации в течение 54 с; при этом было три пятых случая на шаге 3 и затем один первый случай. Всего по результатам препроцессинга в КНФ было добавлено 64 дизъюнкта, 16 из них — однолитеральных. Отметим, что этот выход является самым сложным — до этого Kissat смог решить SAT-задачу для него максимум для $i = 2$.

На следующей стадии программа была применена с теми же параметрами к двум КНФ для $i = 9$, рассмотренным выше. На КНФ для выхода № 5 препроцессинг ничего не дал, так как в первой же итерации на шаге 3 возник первый случай, т. е. на всех подзадачах Kissat был прерван. По этой причине программа была запущена ещё раз, но с параметром confl = 100 000. С таким ограничением Kissat работает примерно 3 с. На этот раз на КНФ для выхода № 5 программа отработала за 207 с, в течение которых было выполнено 30 итераций алгоритма. В модифицированной КНФ в итоге оказалось на 239 дизъюнктов больше, чем в исходной. На КНФ для выхода № 6 программа выполнила 9 итераций за 99 с (при этом также было использовано confl = 100 000). В результате в КНФ были добавлены 164 дизъюнкта. На обеих модифицированных КНФ был запущен Cube-and-Conquer-решатель EnCnC в режиме прогнозирования с теми же входными параметрами, что и в п. 3.2. Результаты представлены в табл. 5.

Таблица 5

Результаты для КНФ при $i = 9$ после препроцессинга

n	Подзадач	Отсечённых листьев
Выход № 5		
1355	2 631	3
1350	4 761	3
1345	8 485	6
1340	15 028	9
Выход № 6		
1345	3 018	1
1340	5 376	2
1335	9 700	2
1330	17 103	4

Во-первых, отметим, что значения n значительно меньше, чем до препроцессинга. Для выхода № 5 они меняются в диапазоне 1340–1355 переменных, а ранее было 2410–2445. Для выхода № 6 ситуация аналогичная. Это один из ключевых результатов предложенного препроцессинга — по сути, в модифицированных КНФ с точки зрения Cube-and-Conquer примерно на 1 000 переменных меньше, чем в исходной. Во-вторых, во всех выборках подзадачи оказались сложными и не были решены за 5 000 с, поэтому в табл. 5 отсутствует столбец с прогнозным временем решения. Таким образом, EnCnC теперь даёт гораздо более реалистичную картину, чем рань-

ше. Можно построить такую нижнюю оценку прогнозного времени для выхода № 5: если каждую из 15 028 подзадач можно решить за 5 000 с, то на 16-ядерном компьютере на это потребуется 55 дней. Для выхода № 6 аналогичная нижняя оценка составляет 62 дня работы компьютера.

Заключение

Проанализирована стойкость неполнораундовых вариантов функции сжатия Skein512-256 к нахождению прообразов. Ранее в данном контексте рассматривался только первый раунд данной функции сжатия; в настоящей работе рассмотрены варианты с первым раундом и несколькими первыми операциями второго раунда. Задачи нахождения прообраза сведены к SAT с помощью транслятора CBMC. Показано, что время решения соответствующих SAT-задач при помощи алгоритма CDCL и метода Cube-and-Conquer существенно зависит от значения выхода функции сжатия, а также от используемого числа операций второго раунда. В результате найдены прообразы функции сжатия, состоящей из первого раунда и 8 (из 12) операций второго раунда. Предложен подход к препроцессингу КНФ с помощью итеративного использования метода Cube-and-Conquer. Предварительное применение этого препроцессинга позволило найти прообраз для одной конфигурации, которая ранее оказывалась слишком трудной для CDCL-решателя. Также этот препроцессинг позволил построить более точные оценки времени нахождения прообраза версии функции сжатия, в которой число операций второго раунда увеличено до 9. Можно констатировать, что на данный момент даже 2 (из 72) раунда функции сжатия Skein512-256 являются стойкими к нахождению прообраза с помощью SAT-подхода.

ЛИТЕРАТУРА

1. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. 2-е изд. М.: Гелиос АРВ, 2002. 480 с.
2. `https://www.schneier.com/academic/skein/` — The Skein Hash Function Family. 2010.
3. *Homsirikamol E., Morawiecki P., Rogawski M., and Srebrny M.* Security margin evaluation of SHA-3 contest finalists through SAT-based attacks // LNCS. 2012. V. 7564. P. 56–67.
4. *Bard G. V.* Algebraic Cryptanalysis. N.Y.: Springer, 2009. 356 p.
5. *Семёнов А. А., Беспалов Д. В.* Технологии решения многомерных задач логического поиска // Вестн. Том. гос. ун-та. 2005. № 14. С. 61–73.
6. *Marques-Silva J. and Sakallah K.* GRASP: a search algorithm for propositional satisfiability // IEEE Trans. Computers. 1999. V. 48. No. 5. P. 506–521.
7. *Mironov I. and Zhang Z.* Applications of SAT solvers to cryptanalysis of hash functions // LNCS. 2006. V. 4121. P. 102–115.
8. *Heule M. J. H., Kullmann O., Wieringa S., and Biere A.* Cube and Conquer: guiding CDCL SAT solvers by lookaheads // LNCS. 2012. V. 7261. P. 50–65.
9. *Matyas S. M., Meyer C. H., and Oseas J.* Generating strong one-way functions with cryptographic algorithm // IBM Techn. Disclosure Bull. 1985. V. 27. No. 10A. P. 5658–5659.
10. *Khovratovich D., Rechberger C., and Savelieva A.* Bicliques for preimages: attacks on Skein-512 and the SHA-2 family // LNCS. 2012. V. 7549. P. 244–263.
11. *Отпущенников И. В., Семёнов А. А.* Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1(11). С. 96–115.
12. *Clarke E. M., Kroening D., and Lerda F.* A tool for checking ANSI-C programs // LNCS. 2004. V. 2978. P. 168–176.

13. *Заикин О. С., Давыдов В. В., Кирьянова А. П.* Применение алгоритмов решения проблемы булевой выполнимости для анализа финалистов конкурса SHA-3 // Выч. мет. программирование. 2024. Т. 25. Вып. 3. С. 259–273.
14. *Biere A. and Fleury M.* Gimsatul, IsaSAT and Kissat entering the SAT Competition 2022 // Proc. SAT Competition 2022 — Solver and Benchmark Descriptions. University of Helsinki, 2022. P. 10–11.
15. *Heule M. J. H., Kullmann O., and Marek V. W.* Solving and verifying the Boolean Pythagorean triples problem via Cube-and-Conquer // LNCS. 2016. V. 9710. P. 228–245.
16. *Zaikin O.* Inverting cryptographic hash functions via Cube-and-Conquer // J. Artif. Int. Res. 2024. V. 81. P. 359–399.
17. *Zaikin O.* Inverting step-reduced SHA-1 and MD5 by parameterized SAT solvers // Proc. CP 2024. Leibniz Intern. Proc. Informatics (LIPIcs). 2024. V. 307. P. 31:1–31:19.
18. *Andreev A., Chukharev K., Kochemazov S., and Semenov A.* Using backdoors to generate learnt information in SAT solving // Proc. ECAI 2024. P. 4173–4180.
19. `https://github.com/olegzaikin/IterCnC` — Реализация итеративного препроцессинга КНФ при помощьи Cube-and-Conquer. 2025.

REFERENCES

1. *Alferov A. P., Zubov A. Yu., Kuzmin A. S., and Cheremushkin A. V.* Osnovy kriptografii [Basics of Cryptography]. 2nd ed. Moscow, Gelios ARV, 2002. 480 p. (in Russian)
2. `https://www.schneier.com/academic/skein/` — The Skein Hash Function Family. 2010.
3. *Homsirikamol E., Morawiecki P., Rogawski M., and Srebrny M.* Security margin evaluation of SHA-3 contest finalists through SAT-based attacks. LNCS, 2012, vol. 7564, pp. 56–67.
4. *Bard G. V.* Algebraic Cryptanalysis. N.Y., Springer, 2009. 356 p.
5. *Semenov A. A. and Bespalov D. V.* Tekhnologii resheniya mnogomernykh zadach logicheskogo poiska [Technologies for solving multidimensional logical search problems]. Vestnik TSU, 2005, no. 14, pp. 61–73. (in Russian)
6. *Marques-Silva J. and Sakallah K.* GRASP: a search algorithm for propositional satisfiability. IEEE Trans. Computers, 1999, vol. 48, no. 5, pp. 506–521.
7. *Mironov I. and Zhang Z.* Applications of SAT solvers to cryptanalysis of hash functions. LNCS, 2006, vol. 4121, pp. 102–115.
8. *Heule M. J. H., Kullmann O., Wieringa S., and Biere A.* Cube and Conquer: guiding CDCL SAT solvers by lookaheads. LNCS, 2012, vol. 7261, pp. 50–65.
9. *Matyas S. M., Meyer C. H., and Oseas J.* Generating strong one-way functions with cryptographic algorithm. IBM Techn. Disclosure Bull., 1985, vol. 27, no. 10A, pp. 5658–5659.
10. *Khovratovich D., Rechberger C., and Savelieva A.* Bicliques for preimages: attacks on Skein-512 and the SHA-2 family. LNCS, 2012, vol. 7549, pp. 244–263.
11. *Otpuschennikov I. V. and Semenov A. A.* Tekhnologiya translyatsii kombinatornykh problem v bulevy uravneniya [Technology for translating combinatorial problems into Boolean equations]. Prikladnaya Diskretnaya Matematika, 2011, no. 1(11), pp. 96–115. (in Russian)
12. *Clarke E. M., Kroening D., and Lerda F.* A tool for checking ANSI-C programs. LNCS, 2004, vol. 2978, pp. 168–176.
13. *Zaikin O. S., Davydov V. V., and Kiryanova A. P.* Primeneniye algoritmov resheniya problemy bulevoy vypolnimosti dlya analiza finalistov konkursa SHA-3 [SAT-based analysis of SHA-3 competition finalists]. Vychislitel'nye Metody i Programmirovanie, 2024, vol. 25, no. 3, pp. 259–273. (in Russian)

14. *Biere A. and Fleury M.* Gimsatul, IsaSAT and Kissat entering the SAT Competition 2022. Proc. SAT Competition 2022 — Solver and Benchmark Descriptions, University of Helsinki, 2022, pp. 10–11.
15. *Heule M. J. H., Kullmann O., and Marek V. W.* Solving and verifying the Boolean Pythagorean triples problem via Cube-and-Conquer. LNCS, 2016, vol. 9710, pp. 228–245.
16. *Zaikin O.* Inverting cryptographic hash functions via Cube-and-Conquer. J. Artif. Int. Res., 2024, vol. 81, pp. 359–399.
17. *Zaikin O.* Inverting step-reduced SHA-1 and MD5 by parameterized SAT solvers. Proc. CP 2024, Leibniz Intern. Proc. Informatics (LIPIcs), 2024, vol. 307, pp. 31:1–31:19.
18. *Andreev A., Chukharev K., Kochemazov S., and Semenov A.* Using backdoors to generate learnt information in SAT solving. Proc. ECAI 2024, pp. 4173–4180.
19. `https://github.com/olegzaikin/IterCnC` — Implementation of iterative Cube-and-Conquer-based preprocessing for SAT, 2025.

УДК 004.72:004.021:519.17 DOI 10.17223/20710410/71/8

МОДЕЛИРОВАНИЕ И ОЦЕНКА РЕСУРСНЫХ ЗАТРАТ АЛГОРИТМОВ МАРШРУТИЗАЦИИ В СЕТЯХ НА КРИСТАЛЛЕ С ДВУМЕРНОЙ ЦИРКУЛЯНТНОЙ ТОПОЛОГИЕЙ[1]

Э. А. Монахова*, О. Г. Монахов*, Э. Р. Рзаев**, Е. В. Лежнев**, А. Ю. Романов**

*Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

**Национальный исследовательский университет «Высшая школа экономики», г. Москва, Россия

E-mail: emilia@rav.sscc.ru, monakhov@rav.sscc.ru, erzaev@hse.ru, elezhnev@hse.ru, a.romanov@hse.ru

Исследуется совместное конструирование топологий семейств оптимальных по диаметру циркулянтных сетей $C(N; \pm 1, \pm s_2)$ и реализуемых для них алгоритмов маршрутизации сложности $O(1)$. Предлагаемый алгоритм маршрутизации основан на использовании масштабируемых параметров L-образных шаблонов плотной укладки графов на плоскости для семейств оптимальных сетей. Определены аналитические формулы зависимости этих параметров от диаметра графов для семейств оптимальных сетей $C(N; \pm 1, \pm s_2)$, сокращающие сложность их расчёта до $O(1)$. Проведено сравнение предлагаемого алгоритма с известным алгоритмом маршрутизации, модификацией которого он является, по затратам времени на маршрутизацию в семействах оптимальных графов и показано уменьшение времени его исполнения в среднем в 2 раза. Выполнено моделирование исследуемого алгоритма в качестве основы маршрутизатора сети на кристалле на языке описания аппаратуры Verilog. Получены данные сравнения его с другими алгоритмами маршрутизации по занимаемым логическим ресурсам и ресурсам памяти.

Ключевые слова: *неориентированная циркулянтная сеть, алгоритм маршрутизации, семейства оптимальных циркулянтов, сети на кристалле.*

MODELING AND RESOURCE COST ESTIMATION OF ROUTING ALGORITHMS IN NETWORKS ON A CHIP WITH A TWO-DIMENSIONAL CIRCULANT TOPOLOGY

E. A. Monakhova*, O. G. Monakhov*, E. R. Rzaev**, E. V. Lezhnev**, A. Y. Romanov**

*Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

**HSE University, Moscow, Russia

In this paper, we investigate the joint construction of topologies of families of optimal diameter circulant networks $C(N; \pm 1, \pm s_2)$ and the routing algorithms of complexity $O(1)$ implemented for them. The proposed routing algorithm is based on the use of scalable parameters of L-shaped dense graph packing patterns on the plane for families of optimal networks. Analytical formulas for the dependence of these parameters on the graph diameter of families of optimal networks $C(N; \pm 1, \pm s_2)$ are

[1]Исследование выполнено за счёт гранта Российского научного фонда № 25-11-00248.

determined, reducing the complexity of their calculation to $O(1)$. A comparison of the proposed algorithm with the routing algorithm on which it is based was carried out. In families of optimal graphs, the proposed algorithm showed an average two-fold decrease in execution time. The implementation of the investigated routing algorithm as the basis for a network-on-a-chip router is carried out in the hardware description language Verilog. The data of its comparison with other routing algorithms based on the occupied logical and memory resources have been obtained.

Keywords: *undirected circulant network, routing algorithm, families of optimal circulant networks, network-on-a-chip.*

Введение

Структура циркулянтных сетей степени четыре изучается в различных прикладных областях и в качестве топологии сетей связи кластеров вычислительных систем и сетей на кристалле [1–8]. Работы исследователей посвящены в основном изучению их топологических показателей, открытию семейств оптимальных графов, методов их синтеза, разработке алгоритмов маршрутизации и методов их практического применения.

Циркулянтная сеть степени четыре (двумерная циркулянтная сеть) представляет собой неориентированный граф $C(N; \pm s_1, \pm s_2)$ с $1 \leqslant s_1 < s_2 < N/2$ и множеством вершин $V = \mathbb{Z}_N = \{0, 1, \ldots, N-1\}$, где каждая вершина i связана с вершинами $i \pm s_1 \bmod N$ и $i \pm s_2 \bmod N$. Числа s_1, s_2 — образующие графа, N — порядок. Граф $C(N; \pm s_1, \pm s_2)$ связен, если НОД$(N, s_1, s_2) = 1$. На рис. 1 приведено изображение циркулянтной сети $C(21; \pm 1, \pm 6)$. Диаметр графа — длина максимального кратчайшего пути на множестве всевозможных пар вершин, среднее расстояние — математическое ожидание всех длин кратчайших путей между парами вершин. Оптимальным графом называется циркулянтный граф $C(N; \pm s_1, \pm s_2)$ с минимально возможным диаметром для заданного N. Известна точная нижняя граница диаметра оптимального графа с N вершинами: $\mathrm{ulb}(N) = \lceil (-1 + \sqrt{2N-1})/2 \rceil$ [2]. Решение проблемы минимизации диаметра (среднего расстояния) графов связано с оптимизацией задержек при передаче данных, скорости коммуникаций и, в конечном итоге, производительности системы [1–3, 9,10]. Другая актуальная проблема при использовании циркулянтов в качестве сетей связи состоит в разработке эффективных алгоритмов маршрутизации для передачи сообщений между парами узлов [11]. Алгоритм маршрутизации называется оптимальным, если передача сообщений происходит вдоль кратчайших путей из источника в приёмник. Известно несколько алгоритмов маршрутизации для циркулянтов степени четыре с разными оценками сложности (см. обзор в [4]).

В данной работе рассматриваются оптимальные алгоритмы, имеющие константную оценку по времени вычисления маршрута из источника в приёмник и не требующие таблиц маршрутизации, что является преимуществом при проектировании масштабных сетей на кристалле. К ним принадлежат алгоритмы аналитического типа [4, 5, 12, 13], разработанные для специальных семейств оптимальных циркулянтов, и алгоритмы с предварительной подготовкой топологических параметров сети для выполнения маршрутизации [13–15]. К числу последних, как наиболее эффективный из них для сетей на кристалле, относится алгоритм из [14], применимый для двумерных циркулянтов общего вида $C(N; \pm s_1, \pm s_2)$. Он основан на использовании параметров L-образных шаблонов [16–18], задающих плотную укладку графов $C(N; \pm s_1, \pm s_2)$ на плоскости $\mathbb{Z}^2$ (параметры a, b, p, q на рис. 2,*а*). На рис. 2,*б* и *в* показаны L-образ-

ные шаблоны представлений циркулянтных графов $C(10;\pm1,\pm4)$ и $C(12;\pm1,\pm4)$, на рис. 3 — фрагмент укладки графа $C(10;\pm1,\pm4)$ на плоскости. Алгоритм маршрутизации из [14] требует решения $\overline{x},\overline{y}$ сравнения $s_1x+s_2y\equiv 1 \pmod N$ и предварительного определения параметров a,b,p,q для рассматриваемого циркулянта. Для определения значений искомых параметров применяются алгоритмы со сложностью $O(N)$ [18] или $O(\log N)$ [16], что является затратным при расчётах в масштабных сетях с большим количеством узлов.

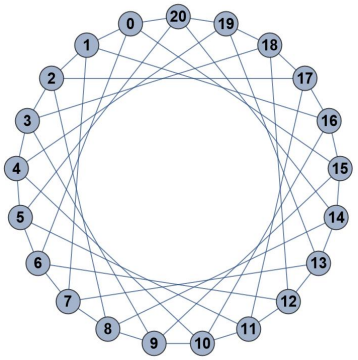

Рис. 1. Циркулянтная сеть $C(21;\pm1,\pm6)$

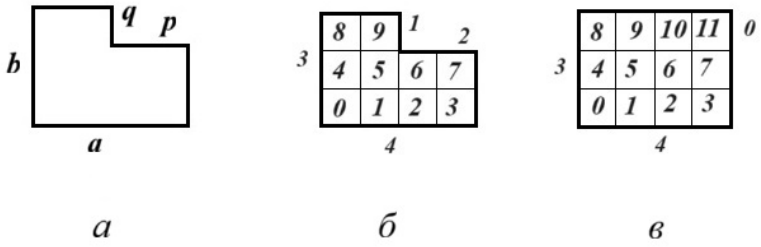

Рис. 2. Параметры L-образных шаблонов для циркулянтов $C(N;\pm1,\pm s_2)$

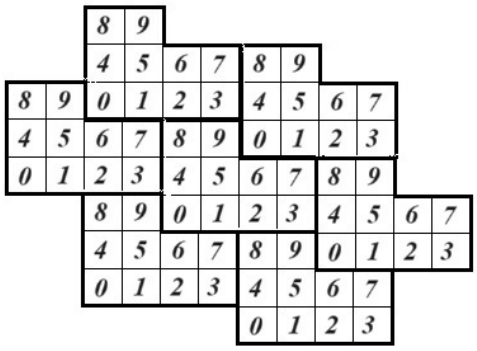

Рис. 3. Плотная укладка на плоскости L-образного шаблона графа $C(10;\pm1,\pm4)$

В настоящей работе исследовано совместное конструирование топологий семейств оптимальных по диаметру циркулянтных сетей $C(N;\pm1,\pm s_2)$ и реализуемых для них оптимальных алгоритмов маршрутизации сложности $O(1)$. С точки зрения выбора эффективного алгоритма маршрутизации для сетей на кристалле рассмотрено множество

семейств оптимальных сетей вида $C(N;\pm1,\pm s_2)$, задаваемых аналитически и масштабируемых по параметрам L-образных шаблонов. Описан алгоритм сложности $O(1)$ получения аналитических формул для параметров a,b,p,q масштабируемых семейств оптимальных циркулянтов $C(N;\pm1,\pm s_2)$. На примере ряда таких семейств проведено сравнение предложенного алгоритма сложности $O(1)$ с алгоритмом маршрутизации из [14] по среднему времени поиска кратчайших путей. Приведены результаты моделирования предложенного алгоритма маршрутизации в сети на кристалле и сравнения его по ресурсным затратам с другими известными алгоритмами. Сделаны выводы об эффективности алгоритма и его применимости в сетях на кристалле.

1. Плотная укладка циркулянтов на плоскости в виде L-образных шаблонов

В [16–18] описаны алгоритмы укладки ориентированных циркулянтных графов $C(N;s_1,s_2)$ на плоскости $\mathbb{Z}^2$ в виде L-образных шаблонов (далее для краткости L-шаблонов). В алгоритме 1 приведён псевдокод построения L-шаблонов и нахождения параметров a,b,p,q для неориентированных циркулянтов вида $C(N;\pm s_1,\pm s_2)$. Алгоритм 1 использовался далее при всех построениях L-шаблонов рассматриваемых графов. Процесс построения L-шаблона графа совпадает с представленным в работе [18], но в отличие от него шаги 1–3 алгоритма 1 описывают явным образом вычисление параметров a,b,p,q. В [16] значения этих параметров определяются по значениям N,s_1,s_2 графа другим алгоритмом — с рекурсивным использованием алгоритма деления Евклида, но в результате не строится сам L-шаблон.

Алгоритм 1. Вычисление параметров a,b,p,q для циркулянтных графов $C(N;\pm s_1,\pm s_2)$

Вход: циркулянтный граф $C(N;\pm s_1,\pm s_2)$.
Выход: Значения a, b, p, q L-шаблона графа.

1: На плоскости $\mathbb{Z}^2$ определить нулевую точку решётки $(0,0)$. Обходить точки первого квадранта с целочисленными координатами (x,y) по диагоналям в следующем порядке: $(0,0)$, $(0,1)$, $(1,0)$, $(0,2)$, $(1,1)$, $(2,0)$, $(0,3)$ и т. д. В каждую посещённую точку (x,y) записать номер вершины $k=(xs_1+ys_2)\bmod N$, если это значение не встречалось ранее. Закончить построение L-шаблона графа, когда все значения $0\leqslant k\leqslant N-1$ записаны.
2: В полученном массиве точек $\{(x,y,k)\}$ определить две точки: (x_1,y_1,k_1) с $\max\limits_x\max\limits_y\{(x,y,k)\}$ и (x_2,y_2,k_2) с $\max\limits_y\max\limits_x\{(x,y,k)\}$; $a:=x_2+1$, $b:=y_1+1$.
3: **Если** $ab\neq N$, **то**
4: $\quad p:=x_2-x_1$, $q:=y_1-y_2$, перейти в п. 11,
5: **иначе**
6: $\quad$ в массиве $\{(x,y,k)\}$ определить точку $(x_3,y_3,N-s_2)$.
7: **Если** $x_3\neq0$, **то**
8: $\quad p:=a-x_3$, $q:=0$, перейти в п. 11,
9: **иначе**
10: $\quad$ в массиве $\{(x,y,k)\}$ выбрать точку $(x_4,y_4,N-s_1)$, $p:=0$, $q:=b-y_4$.
11: Конец алгоритма.

На рис. 2,*б* показан L-шаблон графа $C(10;\pm1,\pm4)$, где $a=4$, $b=3$, $p=2$, $q=1$; на рис. 2,*в* — L-шаблон графа $C(12;\pm1,\pm4)$, где $a=4$, $b=3$, $p=0$ и $q=1$. В [17]

доказано, что L-шаблон двумерного циркулянтного графа всегда образует плотную укладку на плоскости, и найдена система сравнений для расположения нулевых вершин (вершин с номером 0) на плоскости, справедливая как для ориентированных, так и неориентированных циркулянтов:

$$\begin{aligned} as_1 - qs_2 &\equiv 0 \pmod N, \\ -ps_1 + bs_2 &\equiv 0 \pmod N. \end{aligned} \tag{1}$$

При этом число вершин графа равно $N = ab - pq$. Согласно [17], если есть несколько L-шаблонов размера N для заданной позиции нулевых вершин на плоскости, то они все соответствуют одним и тем же образующим s_1 и s_2.

2. Масштабируемость параметров L-шаблонов семейств оптимальных циркулянтов $C(N; \pm 1, \pm s_2)$

Рассмотрим в качестве объекта исследования элементы множества аналитически задаваемых семейств оптимальных циркулянтов вида $C(N; \pm 1, \pm s_2)$. По построению каждое такое семейство состоит из оптимальных графов, т. е. графов с минимально возможным диаметром d для заданного порядка графа. Графы каждого семейства описаны полиномами от диаметра d: N — квадратичная функция, s_2 — линейная или квадратичная функция. Члены семейства существуют при диаметрах

$$d = d_m + kP, \quad k \geqslant 0, \tag{2}$$

где d_m — минимальный диаметр, при котором член семейства является оптимальным графом; $P = \text{const} \in \{1, 2, \ldots\}$ — период появления графов семейства. Множество аналитически задаваемых семейств оптимальных циркулянтов вида $C(N; \pm 1, \pm s_2)$ представлено в Интернете в открытом доступе (далее DLN-датасет): `https://github.com/mila0411/Double-loop-networks/tree/main/Dataset`. В DLN-датасете содержится более 2000 семейств оптимальных графов. По этой ссылке в открытом доступе можно найти список параметров $\{N, s_2, d\}$ описаний всех оптимальных по диаметру циркулянтов вида $C(N; \pm 1, \pm s_2)$ для всех $12 \leqslant N \leqslant 50000$ и всех их оптимальных значений образующих $s_2 < N/2$. Эту часть базы данных будем называть просто «датасет».

В [19] введено понятие *L-масштабируемости* аналитически задаваемых семейств оптимальных сетей $C(N(d); \pm 1, \pm s_2(d))$, состоящее в том, что при укладке на плоскости $\mathbb{Z}^2$ члены семейств образуют последовательности L-шаблонов с параметрами a, b, p, q, аналитически задаваемыми в виде линейных полиномов от диаметра d. Рассмотрим кратко основные принципы концепции L-масштабируемости семейств.

Пусть задано семейство L-масштабируемых оптимальных аналитически задаваемых циркулянтов $C(N(d); \pm 1, \pm s_2(d))$, где диаметр $d \geqslant d_m$, $P \in \{1, 2, \ldots\}$ и выполняется (2). Для двух последовательных членов семейства вычислим значения полиномов $N(d)$, $s_2(d)$ при $d_1 = d_m$ и $d_2 = d_1 + P$. Получим графы $C(N(d_1); \pm 1, \pm s_2(d_1))$ и $C(N(d_2); \pm 1, \pm s_2(d_2))$. Применяя алгоритм 1 вычисления параметров a, b, p, q к найденным графам, определим значения $a(d_1)$, $b(d_1)$, $p(d_1)$, $q(d_1)$ и $a(d_2)$, $b(d_2)$, $p(d_2)$, $q(d_2)$. Для L-масштабируемого семейства диаметра d формулы для искомых параметров имеют следующий вид:

$$\begin{aligned} a(d) &= (\Delta a/P)d + a(d_1) - (\Delta a/P)d_1, \\ b(d) &= (\Delta b/P)d + b(d_1) - (\Delta b/P)d_1, \\ p(d) &= (\Delta p/P)d + p(d_1) - (\Delta p/P)d_1, \\ q(d) &= (\Delta q/P)d + q(d_1) - (\Delta q/P)d_1, \end{aligned} \tag{3}$$

при условии $\Delta a = a(d_2) - a(d_1) \geqslant 0$, $\Delta b = b(d_2) - b(d_1) \geqslant 0$, $\Delta p = p(d_2) - p(d_1) \geqslant 0$, $\Delta q = q(d_2) - q(d_1) \geqslant 0$. Геометрически это означает, что длины соответствующих сторон L-шаблонов укладки графов семейства на плоскости увеличиваются линейно при росте диаметра графа. Одна из них при этом может сохранять свою длину как, например, параметр $q = 1$ на рис. 4.

Таким образом, вычисление значений параметров a, b, p, q с помощью алгоритма 1 при двух малых значениях диаметра $d_1 = d_m$ и $d_2 = d_1 + P$ даёт формулы (3) для них, справедливые при любых возможных диаметрах (2) графов семейства. По формулам (3) можно определить параметры L-шаблона при больших N, таким образом, сложность решения проблемы определения параметров L-шаблонов $O(\log N)$ [16] сокращается до $O(1)$.

Для всех более чем 2000 аналитически задаваемых семейств оптимальных циркулянтов из DLN-датасета проведена проверка выполнения условия (1) и экспериментально подтверждена L-масштабируемость около 90 % семейств. В табл. 1 приведён фрагмент списка L-масштабируемых семейств графов, существующих при каждом диаметре $d \geqslant d_m$, т. е. при $P = 1$. Фрагмент представленных семейств включает значения минимального диаметра d_m, полиномы для N и s_2, коэффициенты при степенях d для полиномов параметров a, b, p, q.

Таблица 1

Фрагмент DLN-датасета L-масштабируемых семейств оптимальных графов

d_m	$\{N, s_2\}$	$\{a_1, a_0\}$	$\{b_1, b_0\}$	$\{p_1, p_0\}$	$\{q_1, q_0\}$
3	$\{2 - 2d + 2d^2, -1 + 2d\}$	{2, −1}	{1, 0}	{1, −2}	{0, 1}
10	$\{-8 - d + 2d^2, 4 + 2d\}$	{1, 0}	{3, −7}	{1, −4}	{1, −2}
5	$\{-3 - d + 2d^2, 2 + 2d\}$	{1, 1}	{2, −3}	{0, 0}	{1, −1}
8	$\{-13 + 2d^2, -5 + 2d\}$	{2, −5}	{1, 3}	{1, −2}	{0, 1}
7	$\{-11 + 2d^2, -5 + 2d\}$	{2,−5}	{1, 3}	{1, −4}	{0, 1}
4	$\{-4 + 2d^2, -3 + 2d\}$	{2, −3}	{1, 2}	{1, −2}	{0, 1}
2	$\{2d^2, -1 + 2d\}$	{2, −1}	{1, 1}	{1, −1}	{0, 1}
10	$\{-28 + d + 2d^2, 8 + 2d\}$	{1, 4}	{2, −7}	{0, 0}	{1, −3}
3	$\{-1 + 2d + 2d^2, 3 + 2d\}$	{1, 1}	{3, −1}	{1,0}	{1, 0}
1	$\{1 + 2d + 2d^2, 1 + 2d\}$	{2, 1}	{1, 1}	{1, 0}	{0, 1}

Проведённая экспериментальная проверка ограничена размерами датасета ($N \leqslant$ $\leqslant 50000$, $d \leqslant 158$). Докажем на некоторых примерах, что свойство L-масштабируемости сохраняется для построенных семейств оптимальных циркулянтов из DLN-датасета при любых диаметрах из области их определения (2). Рассмотрим известное в литературе (для ссылок см. [2]) семейство экстремальных циркулянтов — оптимальных двумерных графов с максимально возможным числом вершин при любом диаметре, у которых диаметр равен точной нижней границе $\mathrm{ulb}(N)$.

Утверждение 1 [19]. Параметры L-шаблонов для семейства оптимальных циркулянтов $C(2d^2 + 2d + 1; \pm 1, \pm(2d + 1))$ равны $a(d) = 2d + 1$, $b(d) = d + 1$, $p(d) = d$, $q(d) = 1$ при любом диаметре $d \geqslant 1$.

На рис. 4 изображены L-шаблоны укладки на плоскости трёх графов семейства из утверждения 1: $C(13; \pm 1, \pm 5)$, $C(25; \pm 1, \pm 7)$, $C(41; \pm 1, \pm 9)$ с диаметрами $d = 2, 3, 4$ соответственно.

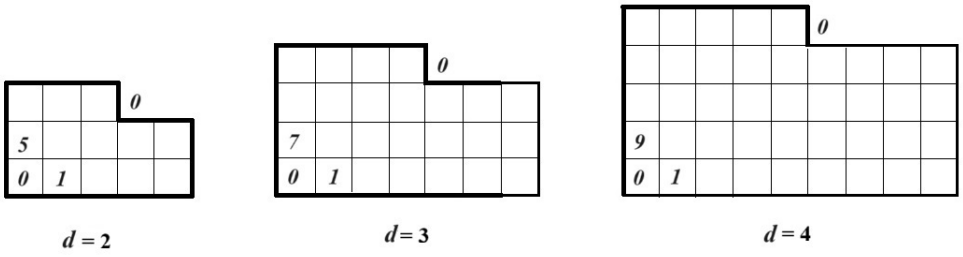

Рис. 4. Масштабируемость параметров L-шаблонов для графов семейства из утверждения 1

Утверждение 2. Параметры L-шаблонов для семейства оптимальных циркулянтов $C(2d^2; \pm 1, \pm(2d-1))$ равны $a(d) = 2d-1$, $b(d) = d+1$, $p(d) = d-1$, $q(d) = 1$ при любом диаметре $d > 1$.

Доказательство. Пусть $d_1 = 2$, $d_2 = 3$. Применив алгоритм 1, получим $\Delta a = 9 - 7 = 2$, $\Delta b = 17 - 11 = 6$, $\Delta p = 8 - 6 = 2$, $\Delta q = 6 - 4 = 2$. В силу (3) $a(d) = 2d-1$, $b(d) = d+1$, $p(d) = d-1$, $q(d) = 1$. Проверка выполнения (1) даёт при любом $d > 1$

$$2d - 1 - (2d-1) = 0,$$
$$-(d-1) + (d+1)(2d-1) = 2d^2 = N.$$

Утверждение 2 доказано. ■

Утверждение 3. Параметры L-шаблонов для семейства оптимальных циркулянтов $C(2d^2 + d - 1; \pm 1, \pm(2d+2))$ равны $a(d) = d+1$, $b(d) = 2d-1$, $p(d) = 0$, $q(d) = d$ при любом диаметре $d > 2$.

Доказательство. Пусть $d_1 = 3$, $d_2 = 4$. Применив алгоритм 1, получим $\Delta a = 8 - 6 = 2$, $\Delta b = 12 - 8 = 4$, $\Delta p = 0$, $\Delta q = 5 - 3 = 2$. В силу (3) $a(d) = d+1$, $b(d) = 2d-1$, $p(d) = 0$, $q(d) = d$. Проверка выполнения (1) даёт при любом $d > 2$

$$(d+1) - d(2d+2) = -2d^2 - d + 1 = -N,$$
$$-0 + (2d-1)(2d+2) = 4d^2 + 2d - 2 = 2N.$$

Утверждение 3 доказано. ■

Утверждение 4. Параметры L-шаблонов для семейства оптимальных циркулянтов $C(2d^2 + d - 28; \pm 1, \pm(2d+8))$ равны $a(d) = d+4$, $b(d) = 2d-7$, $p(d) = 0$, $q(d) = d-3$ при любом диаметре $d \geqslant 10$.

Доказательство. Пусть $d_1 = 10$, $d_2 = 11$. Применив алгоритм 1, получим $\Delta a = 15 - 14 = 1$, $\Delta b = 15 - 13 = 2$, $\Delta p = 0$, $\Delta q = 8 - 7 = 1$. В силу (3) $a(d) = d+4$, $b(d) = 2d-7$, $p(d) = 0$, $q(d) = d-3$. Проверка выполнения (1) даёт при любом $d \geqslant 10$

$$(d+4) - (d-3)(2d+8) = -2d^2 - d + 28 = -N,$$
$$-0 + (2d-7)(2d+8) = 4d^2 + 2d - 56 = 2N.$$

Утверждение 4 доказано. ■

Равенство $p(d) = 0$ в утверждениях 3 и 4 соответствует семействам с прямоугольным типом L-шаблона. Как показано в [16], прямоугольный тип L-шаблонов имеет место, если $bs_2 \equiv 0 \pmod N$. В наших случаях $N = (d+1)(2d-1)$ и $N = (d+4)(2d-7)$ соответственно. Оптимальные графы, которым соответствуют L-шаблоны прямоугольного вида (решётчатые структуры), хорошо подходят для практической реализации

в качестве топологий сетей на кристалле благодаря минимальному числу пересекающихся линий связи и независимости длины максимальной линии связи от числа узлов [20, 21], что является преимуществом при проектировании масштабных сетей на кристалле. Поиск семейств оптимальных кольцевых циркулянтов с прямоугольным типом L-шаблонов, которые к тому же существенно меньше по диаметру двумерных тороидальных структур того же размера, является актуальной задачей при проектировании топологий сетей на кристалле.

3. Алгоритм маршрутизации для семейств оптимальных циркулянтов

В силу симметрии циркулянтов при поиске кратчайших путей между двумя вершинами достаточно решить задачу поиска кратчайших путей из 0 во все вершины циркулянта. Известные алгоритмы маршрутизации [9, 12–15] решают проблему поиска кратчайших путей для двумерного циркулянтного графа, используя координаты девяти, семи или пяти соседних нулей в плотной укладке графа на плоскости и определяя кратчайший путь к вершине как минимум расстояний от неё до этих нулей.

В работе [19] для оптимальных сетей вида $C(N;\pm1,\pm s_2)$ предложена модификация алгоритма маршрутизации из [14], использующего координаты пяти соседних нулей. Обозначим эти нули как $(0,0)$, (u,v), $(-a_0,b_0)$, $(-u,-v)$ и $(a_0,-b_0)$. Далее приведён текст предложенного алгоритма маршрутизации (алгоритм 2). Запись типа $a_1+b_1[s_2]$ означает, что путь из 0 в вершину i содержит a_1 шагов по образующей $s_1=1$ плюс b_1 шагов по образующей s_2. Знаки a_1 и b_1 определяют направление движения по образующей $(+)$ или против $(-)$. Запись $\mathrm{round}(x)$ означает операцию округления: $\mathrm{round}(x)=\lfloor x+0{,}5\rfloor$.

Предварительный этап настройки алгоритма 2 для графов оптимального семейства $C(N(d);\pm1,\pm s_2(d))$, где d удовлетворяет (2), состоит в следующем: по формулам (3) определяются параметры $a(d)$, $b(d)$, $p(d)$, $q(d)$ для L-шаблона графов семейства. Решением $(\overline{x},\overline{y})$ сравнения $x+s_2(d)y\equiv1 \pmod N$ в данном случае является $\overline{x}=1$, $\overline{y}=0$. Этот предварительный этап делается один раз при формировании топологии системы в виде графа $C(N;\pm1,\pm s_2)$.

Алгоритм 2. Вычисление кратчайшего пути из 0 в любую вершину графа $C(N;\pm1,\pm s_2)$

Вход: параметры N, s_2, a, b, p, q, $u=a-p$, $v=b-q$, номер вершины приёмника $i\in\{1,\dots,N-1\}$.

Выход: кратчайший путь из 0 в вершину i графа.

1: **Если** $u\geqslant v$, **то**
   $a_0:=p$; $b_0:=b$;
2: **иначе**
3: $a_0:=a$; $b_0:=q$.
4: $(a_1,b_1):=(i,0)-(\mathrm{round}(ib_0/N),\mathrm{round}(-iv/N))\begin{pmatrix}u & v\\ -a_0 & b_0\end{pmatrix}$.
5: $P_1:=a_1+b_1[s_2]$, $P_2:=a_1-u+(b_1-v)[s_2]$, $P_3:=a_1+a_0+(b_1-b_0)[s_2]$, $P_4:=a_1+u+(b_1+v)[s_2]$, $P_5:=a_1-a_0+(b_1+b_0)[s_2]$.
6: Из путей P_1–P_5 выбрать минимальный по сумме шагов по двум образующим кратчайший путь P' из 0 в вершину i.

В силу симметрии циркулянтов кратчайший путь из вершины i в вершину j равен кратчайшему пути из 0 в вершину $(j-i) \bmod N$, поэтому при вычислении оптимально-

го маршрута из любого источника в приёмник делается указанная поправка с помощью циклического сдвига номеров вершин.

Алгоритм 2 реализован на PC с AMD Ryzen 5 5500U в системе высокоуровневого моделирования Wolfram Mathematica (WM) и экспериментально проверен на множестве оптимальных циркулянтов вида $C(N;\pm1,\pm s_2)$ из датасета путём сравнения с работой алгоритма Дейкстры, взятом из библиотеки подпрограмм WM. В силу симметрии циркулянтов в качестве источников сообщений рассматривались нулевые вершины графов, в качестве приёмников — каждая вершина $i = 1, 2, \ldots, N-1$. Для всех порядков графов $12 \leqslant N \leqslant 2048$ и всех их оптимальных образующих s_2 из датасета оптимальных циркулянтов алгоритм 2 показал 100 %-е совпадение по вычисляемой длине кратчайшего пути. Также выборочно проверялись отдельные структуры с числом вершин до $N = 25000$.

Отметим, что алгоритм 2 является модификацией алгоритма маршрутизации [14], предназначенной для работы с оптимальными графами датасета, а не его частным случаем при $t = \lfloor p/u \rfloor = 0$ (или $t = \lfloor q/v \rfloor = 0$), поскольку шаг 1 алгоритма 2 правильно задаёт координаты соседнего нуля $(-a_0, b_0)$ и при $t = 1$.

4. Результаты реализации алгоритма поиска кратчайшего пути

Проведено сравнение по времени работы алгоритма 2 с алгоритмом из [14]. В качестве тестовых заданий взяты графы семейств из утверждений 1–3. На рис. 5–7 приведены средние оценки времени T (в секундах) работы алгоритма 2 и алгоритма из [14], полученные для трёх семейств оптимальных графов из утверждений 1–3 соответственно. Здесь d — диаметр графов; источник сообщений — нулевая вершина; приёмники — все вершины графа. Точками на графиках обозначены результаты работы алгоритма 2, квадратами — алгоритма из [14]. Тестирование проводилось на PC с параметрами AMD Ryzen 5 5500U, 16 Гбайт, Windows 11.

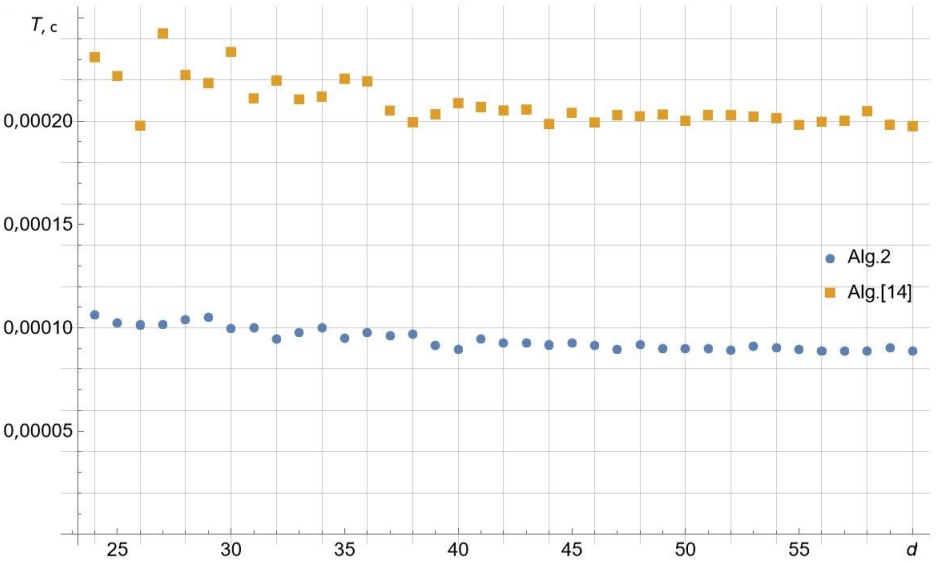

Рис. 5. Среднее время работы алгоритмов маршрутизации на графах семейства из утверждения 1

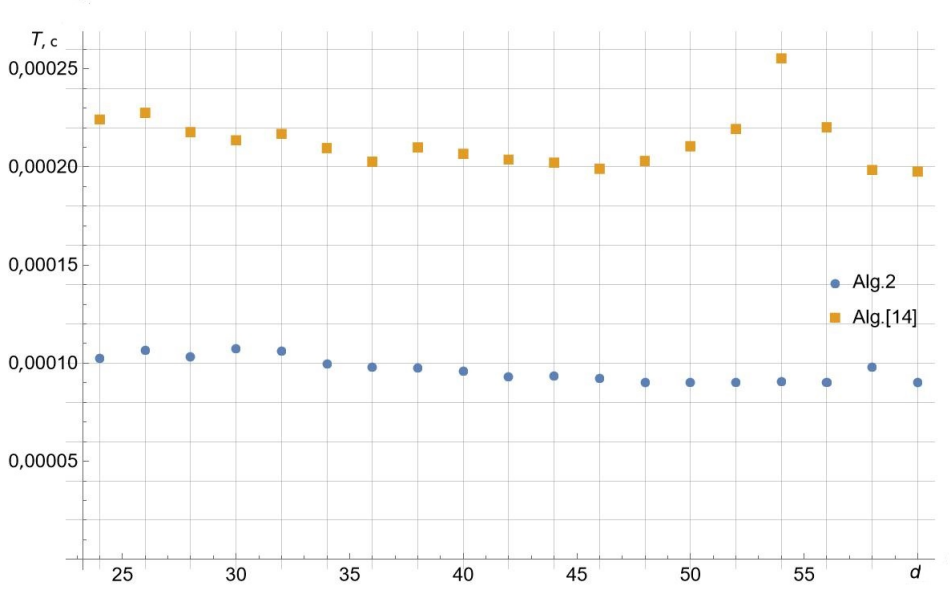

Рис. 6. Среднее время работы алгоритмов маршрутизации на графах семейства из утверждения 2

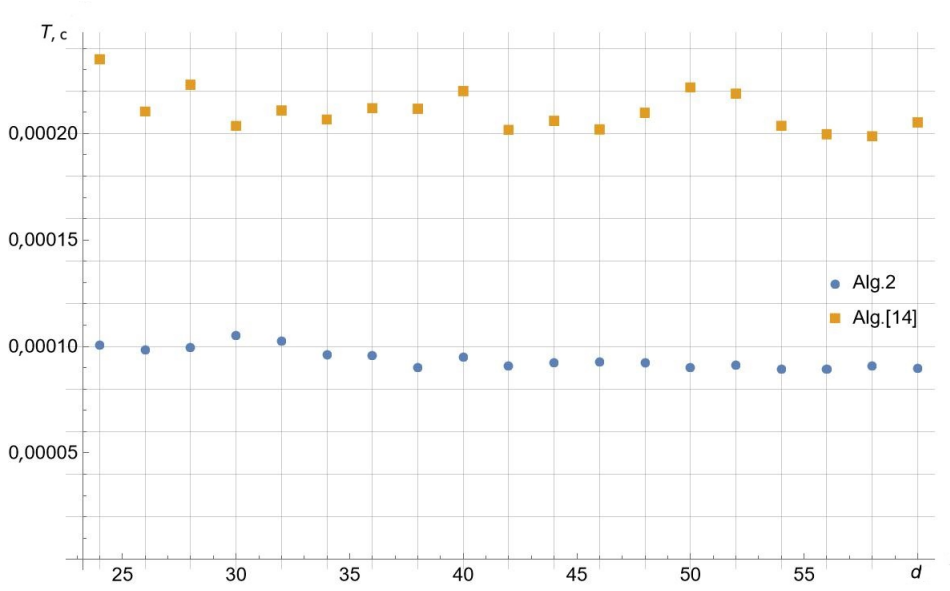

Рис. 7. Среднее время работы алгоритмов маршрутизации на графах семейства из утверждения 3

Новый алгоритм показал затраты времени для расчёта кратчайших путей в среднем в 2 раза меньше, чем алгоритм из [14]. Кроме того, он может работать на семействах оптимальных двумерных циркулянтов вида $C(N; \pm 1, \pm s_2)$, в отличие от алгоритмов из [9, 12, 13], применимых для специальных семейств оптимальных графов. Полученные результаты демонстрируют, что среднее время вычисления оптимального маршрута между двумя узлами не зависит от числа узлов в сети.

Отметим, что для графов с прямоугольным L-контуром укладки на плоскости (таких, как графы из утверждений 3 и 4) время исполнения алгоритмов маршрутизации может быть существенно уменьшено за счёт сокращения до трёх или четырёх количества необходимых для маршрутизации соседних нулей.

5. Результаты моделирования алгоритма маршрутизации в сети на кристалле: оценки ресурсных затрат

Для сетей на кристалле основными критическими показателями, ограничивающими возможности кристалла, являются затраты памяти и логических блоков, в том числе при сравнении различных видов алгоритмов маршрутизации.

Выполнено моделирование алгоритма 2 (обозначим его LR) в качестве основы маршрутизатора сети на кристалле и проверена корректность его работы в программе низкоуровневого моделирования HDLNoCGen [22]. Для реализации алгоритма LR в сети на кристалле на языке описания аппаратуры Verilog шаг 2 алгоритма был частично изменён. Это связано с тем, что для его работы необходимо производить деление и реализовать математику работы с вещественными числами, что является ресурсозатратной задачей для ПЛИС. Поэтому вычисление (a_1, b_1) разбивается на два этапа.

На первом этапе происходит вычисление округлённых значений x_1, x_2:

— $x_1 := ib_0 \operatorname{div} N$; если $ib_0 x_1 \geqslant N/2$, то $x_1 := x_1 + 1$;
— $x_2 := -iv \operatorname{div} N$; если $-ivx_2 \geqslant N/2$, то $x_2 := x_2 - 1$.

На втором этапе происходит дальнейшее преобразование шага 2:

— $(a_1, b_1) := (i - x_1 u + x_2 a_0, -x_1 v - x_2 b_0)$.

Шаги 3 и 4 алгоритма LR логически остаются без изменений, отличие — в последовательной организации вычислений $P_1, \ldots, P_5$ с целью уменьшения количества регистров, необходимого для хранения результатов.

Формула для теоретического расчёта ресурсов памяти в битах для общего вида алгоритма LR: $2\lceil \log_2 N \rceil + 10\lceil \log_2 2d \rceil$, где N — количество узлов в сети; d — диаметр графа; $\lceil \log_2 N \rceil$ — необходимое количество битов для хранения порядкового номера маршрутизатора в сети и общего количества маршрутизаторов; $\lceil \log_2 2d \rceil$ — необходимое количество битов для хранения одной переменной для промежуточных вычислений. Поскольку вычисление коэффициентов a, b, p, q происходит через диаметр циркулянта d, хранить их не требуется.

Тестирование алгоритма проводилось для семейств оптимальных циркулянтов из утверждений 1–4. Так как аппаратные затраты практически не зависят от вида семейств, приведены результаты тестирования на примере семейства оптимальных циркулянтов из утверждения 1 (табл. 2 и 3), из которых следует, что рост потребления ресурсов чипа при росте числа узлов может быть описан квадратичной зависимостью. При этом не наблюдается резких выбросов на данных, то есть зависимость можно описать гладкой функцией с достаточной точностью. Используя стандартный аппарат математической библиотеки Python, на основе полученных практических данных найдены следующие аппроксимационные формулы для расчёта ресурсов памяти и логических ресурсов:

$$U_{\text{reg}} = 0{,}0484N^2 + 34{,}71N + 114{,}27,$$
$$U_{\text{alm}} = 0{,}4566N^2 + 221{,}28N - 1141{,}2.$$

Проведено сравнение алгоритма LR и алгоритма из [14], применимого для циркулянтов $C(N; \pm s_1, \pm s_2)$. Они показали соизмеримые характеристики по занимаемым логическим ресурсам и ресурсам памяти. Но алгоритм LR лучше алгоритма из [14], как было показано в п. 4, почти в 2 раза по времени расчёта кратчайших путей в графе.

Проведено также сравнение алгоритма LR с четырьмя ранее моделированными в сети на кристалле алгоритмами маршрутизации: AA [23] — адаптивным алгоритмом

Таблица 2

Ресурсы памяти в регистрах REG

Циркулянт	N	Диаметр	Один маршрутизатор, REG	Вся сеть, REG
$C(25;1,7)$	25	3	23	745
$C(41;1,9)$	41	4	27	1410
$C(61;1,11)$	61	5	27	2089
$C(85;1,13)$	85	6	31	3367
$C(113;1,15)$	113	7	31	4468
$C(145;1,17)$	145	8	35	6431
$C(181;1,19)$	181	9	35	8069
$C(221;1,21)$	221	10	35	9816

Таблица 3

Логические ресурсы в ALM-блоках

Циркулянт	N	Диаметр	Один маршрутизатор, ALM	Вся сеть, ALM
$C(25;1,7)$	25	3	139	3415
$C(41;1,9)$	41	4	184	7682
$C(61;1,11)$	61	5	184	11537
$C(85;1,13)$	85	6	237	20679
$C(113;1,15)$	113	7	237	27496
$C(145;1,17)$	145	8	301	44561
$C(181;1,19)$	181	9	301	55683
$C(221;1,21)$	221	10	301	67956

для циркулянтов $C(N;\pm1,\pm s_2)$; AC [23], PEA [4], GRBT [9] — алгоритмами, применимыми для семейства циркулянтов вида $C(N;\pm d,\pm(d+1))$ при $d>1$ и использующими разные принципы вычисления кратчайшего пути в графах. На основе данных о занимаемых ресурсах памяти (REG) и логических ресурсах (ALM), приведённых в [4, 9, 23], созданы аппроксимационные формулы для вычисления значений ресурсов для циркулянтов с количеством узлов, определяемым согласно утверждению 1. Результаты сравнения приведены в табл. 4 и 5, где алгоритмы расположены в порядке возрастания требуемых ресурсов.

Таблица 4

Ресурсы памяти REG для разных алгоритмов маршрутизации

N	AA	LR	PEA	AC	GRBT
25	305	745	1286	1319	2550
41	554	1410	2293	2536	5007
61	913	2089	3569	4273	8232
85	1415	3367	5125	6674	12326
113	2100	4468	6975	9913	17413
145	3011	6431	9135	14190	23635
181	4202	8069	11624	19736	31155
221	5729	9816	14461	26811	40158

Получены следующие результаты: по занимаемым ресурсам памяти алгоритм LR лучше всех рассмотренных в табл. 4, кроме алгоритма AA. Алгоритм AA — лучше по ресурсам памяти, но проигрывает LR в быстродействии из-за большого количества

Таблица 5

Логические ресурсы ALM для разных алгоритмов маршрутизации

N	PEA	LR	GRBT	AC	AA
25	3560	3415	4755	6350	12524
41	7004	7682	9198	13666	25829
61	11352	11537	14435	26066	44121
85	16632	20679	20255	45721	68508
113	22879	27496	26405	75235	100319
145	30132	44561	32589	117648	141106
181	38438	55683	38470	176433	192639
221	47848	67956	43667	255495	256914

операций деления и присваивания [23]. По логическим ресурсам LR лучше алгоритмов AA и AC. По сравнению с алгоритмом GRBT он показал себя лучше по ALM для сети с размером до 61 узла, но алгоритм GRBT в 4 раза хуже по требуемым затратам памяти.

По сравнению с алгоритмом аналитического типа PEA он оказывается лучше почти в 1,5 раза по ресурсам памяти, но проигрывает до 30 % по логическим ресурсам. Так как ALM обычно менее ценный ресурс, чем память, которая определяет предельно возможное количество маршрутизаторов в сети на кристалле, то предложенный алгоритм для реализации предпочтительнее, чем алгоритмы PEA и GRBT. В отличие от алгоритмов PEA и GRBT, применимых для оптимальных циркулянтов вида $C(N; \pm d, \pm(d+1))$, алгоритм LR ориентирован на использование в сети на кристалле топологии кольцевых циркулянтов $C(N; \pm 1, \pm s_2)$. Отметим, что единичная образующая $s_1 = 1$ является удобной для реализации простых локальных алгоритмов маршрутизации и сложных адаптивных и отказоустойчивых вычислений, и для такой топологии сети на кристалле можно применять готовые 4-портовые маршрутизаторы, используемые для тороидальных топологий.

Заключение

В работе на уровнях высокоуровневого моделирования в системе Wolfram Mathematica и низкоуровнего моделирования сети на кристалле HDLNoCGen исследован комплексный подход к кодизайну топологий и алгоритмов маршрутизации для семейств оптимальных по диаметру кольцевых двумерных циркулянтных сетей. Проведено моделирование в сети на кристалле топологий семейств оптимальных кольцевых циркулянтов. На таких топологиях реализован и исследован новый алгоритм вычисления кратчайших путей по оценкам временных и ресурсных затрат. Реализация предложенного алгоритма маршрутизации в качестве основы маршрутизатора в сети на кристалле и проведённое сравнение с другими алгоритмами поиска кратчайших путей показали приемлемые значения его ресурсных затрат по памяти и логическим ресурсам и эффективность по затратам времени при расчёте кратчайших путей.

ЛИТЕРАТУРА

1. *Hwang F. K.* A survey on multi-loop networks // Theoret. Comput. Sci. 2003. No. 299. P. 107–121.
2. *Монахова Э. А.* Структурные и коммуникативные свойства циркулянтных сетей // Прикладная дискретная математика. 2011. № 3 (13). С. 92–115.

3. *Huang X., Ramos A. F., and Deng Y.* Optimal circulant graphs as low-latency network topologies // J. Supercomput. 2022. V. 78. No. 11. P. 13491–13510.
4. *Monakhova E. A., Romanov A. Y., and Lezhnev E. V.* Shortest path search algorithm in optimal two-dimensional circulant networks: Implementation for Networks-on-Chip // IEEE Access. 2020. V. 8. P. 215010–215019.
5. *Liu H., Li X., and Wang S.* Construction of dual optimal bidirectional double-loop networks for optimal routing // Mathematics. 2022. V. 10. No. 21. Paper 4016.
6. *Hoffmann R., Désérable D., and Seredyński F.* Cellular automata rules solving the wireless sensor network coverage problem // Nat. Comput. 2022. V. 21. P. 417–447.
7. *Erickson A., Stewart I. A., Navaridas J., and Kiasari A. E.* The stellar transformation // Comput. Netw. 2017. V. 113. P. 29–45.
8. *Fei J. and Lu C.* Adaptive sliding mode control of dynamic systems using double loop recurrent neural network structure // IEEE Trans. Neural Netw. Learn. Syst. 2018. V. 29. P. 1275–1286.
9. *Monakhova E. A., Monakhov O. G., and Romanov A. Yu.* Routing algorithms in optimal degree four circulant networks based on relative addressing: Comparative analysis for networks-on-chip // IEEE Trans. Network Sci. Eng. 2023. V. 10. No. 1. P. 413–425.
10. *Deng Y., Guo M., Ramos A. F., et al.* Optimal low-latency network topologies for cluster performance enhancement // J. Supercomput. 2020. V. 76. No. 12. P. 9558–9584.
11. *Muhsen Y. R., Husin N. A., Zolkepli M. B., et al.* 181Routing techniques in network-on-chip based multiprocessor-system-on-chip for IOT: A systematic review // Iraqi J. Comput. Sci. Math. 2024. V. 5. Iss. 1. Article 16.
12. *Beivide R., Herrada E., Balcazar J. L., and Arruabarrena A.* Optimal distance networks of low degree for parallel computers // IEEE Trans. Comput. 1991. V. 40. No. 10. P. 1109–1124.
13. *Jha P. K.* Dimension-order routing algorithms for a family of minimal-diameter circulants // J. Inter. Networks. 2013. V. 14. No. 1. P. 1350002.
14. *Chen B.-X., Meng J.-X., and Xiao W.-J.* A constant time optimal routing algorithm for undirected double-loop networks // LNCS. 2005. V. 3794. P. 308–316.
15. *Camarero C., Martinez C., and Beivide R.* L-networks: a topological model for regular two-dimensional interconnection networks // IEEE Trans. Comput. 2013. V. 62. No. 7. P. 1362–1375.
16. *Hwang F. K.* A complementary survey on double-loop networks // Theoret. Comput. Sci. 2001. No. 263. P. 211–229.
17. *Fiol M. A., Yebra J. L. A., Alegre I., and Valero M.* A discrete optimization problem in local networks and data alignment // IEEE Trans. Comput. 1987. V. 36. No. 6. P. 702–713.
18. *Wong C. K. and Coppersmith D.* A combinatorial problem related to multimodule memory organizations // J. Assoc. Comput. Mach. 1974. V. 21. No. 3. P. 392–402.
19. *Монахов О. Г., Монахова Э. А.* Масштабируемый подход к кодизайну топологий и алгоритмов маршрутизации для семейств оптимальных циркулянтных сетей степени четыре // Дискретн. анализ и исслед. опер. 2025. Т. 32. № 2. С. 88–106.
20. *Beivide R., Herrada E., Balcazar J. L., and Labarta J.* Optimized mesh-connected networks for SIMD and MIMD architectures // Proc. ISCA'87. Pittsburgh, Pennsylvania, USA, 1987. P. 163–170.
21. *Lau F. C. M. and Chen G.* Optimal layouts of midimew networks // IEEE Trans. Parallel Distrib. Syst. 1996. V. 7. No. 9. P. 954–961.
22. *Lezhnev E., Zunin V., Amerikanov A., and Romanov A.* Electronic computer-aided design for low-level modeling of networks-on-chip // IEEE Access. 2024. V. 12. P. 48750–48763.

23. *Romanov A. Y.* Development of routing algorithms in networks-on-chip based on ring circulant topologies // Heliyon. 2019. V. 5. No. 4. Paper e01516.

REFERENCES

1. *Hwang F. K.* A survey on multi-loop networks. Theoret. Comput. Sci., 2003, no. 299, pp. 107–121.
2. *Monakhova E. A.* Strukturnye i kommunikativnye svoystva tsirkulyantnykh setey [Structural and communicative properties of circulant networks]. Prikladnaya Diskretnaya Matematika, 2011, no. 3 (13), pp. 92–115. (in Russian)
3. *Huang X., Ramos A. F., and Deng Y.* Optimal circulant graphs as low-latency network topologies. J. Supercomput. 2022, vol. 78, no. 11, pp. 13491–13510.
4. *Monakhova E. A., Romanov A. Y., and Lezhnev E. V.* Shortest path search algorithm in optimal two-dimensional circulant networks: Implementation for Networks-on-Chip. IEEE Access. 2020, vol. 8. P. 215010–215019.
5. *Liu H., Li X., and Wang S.* Construction of dual optimal bidirectional double-loop networks for optimal routing. Mathematics, 2022, vol. 10, no. 21, paper 4016.
6. *Hoffmann R., Désérable D., and Seredyński F.* Cellular automata rules solving the wireless sensor network coverage problem. Nat. Comput., 2022, vol. 21, pp. 417–447.
7. *Erickson A., Stewart I. A., Navaridas J., and Kiasari A. E.* The stellar transformation. Comput. Netw., 2017, vol. 113, pp. 29–45.
8. *Fei J. and Lu C.* Adaptive sliding mode control of dynamic systems using double loop recurrent neural network structure. IEEE Trans. Neural Netw. Learn. Syst., 2018, vol. 29, pp. 1275–1286.
9. *Monakhova E. A., Monakhov O. G., and Romanov A. Yu.* Routing algorithms in optimal degree four circulant networks based on relative addressing: Comparative analysis for networks-on-chip. IEEE Trans. Network Sci. Eng., 2023, vol. 10, no. 1, pp. 413–425.
10. *Deng Y., Guo M., Ramos A. F., et al.* Optimal low-latency network topologies for cluster performance enhancement. J. Supercomput., 2020, vol. 76, no. 12, pp. 9558–9584.
11. *Muhsen Y. R., Husin N. A., Zolkepli M. B., et al.* 181Routing techniques in network-on-chip based multiprocessor-system-on-chip for IOT: A systematic review. Iraqi J. Comput. Sci. Math., 2024, vol. 5, iss. 1, article 16.
12. *Beivide R., Herrada E., Balcazar J. L., and Arruabarrena A.* Optimal distance networks of low degree for parallel computers. IEEE Trans. Comput., 1991, vol. 40, no. 10, pp. 1109–1124.
13. *Jha P. K.* Dimension-order routing algorithms for a family of minimal-diameter circulants. J. Inter. Networks, 2013, vol. 14, no. 1, pp. 1350002.
14. *Chen B.-X., Meng J.-X., and Xiao W.-J.* A constant time optimal routing algorithm for undirected double-loop networks. LNCS, 2005, vol. 3794, pp. 308–316.
15. *Camarero C., Martinez C., and Beivide R.* L-networks: a topological model for regular two-dimensional interconnection networks. IEEE Trans. Comput., 2013, vol. 62, no. 7, pp. 1362–1375.
16. *Hwang F. K.* A complementary survey on double-loop networks. Theoret. Comput. Sci., 2001, no. 263, pp. 211–229.
17. *Fiol M. A., Yebra J. L. A., Alegre I., and Valero M.* A discrete optimization problem in local networks and data alignment. IEEE Trans. Comput., 1987, vol. 36, no. 6, pp. 702–713.
18. *Wong C. K. and Coppersmith D.* A combinatorial problem related to multimodule memory organizations. J. Assoc. Comput. Mach., 1974, vol. 21, no. 3, pp. 392–402.
19. *Monakhov O. G. and Monakhova E. A.* Masshtabiruemyy podkhod k kodizaynu topologiy i algoritmov marshrutizatsii dlya semeystv optimal'nykh tsirkulyantnykh setey stepeni chetyre

[A scalable approach to co-design of topologies and routing algorithms for families of optimal degree-four circulant networks]. Diskretnyi Analiz i Issledovanie Operatsii, 2025, vol. 32, no. 2, pp. 88–106. (in Russian)

20. *Beivide R., Herrada E., Balcazar J. L., and Labarta J.* Optimized mesh-connected networks for SIMD and MIMD architectures. Proc. ISCA'87, Pittsburgh, Pennsylvania, USA, 1987, pp. 163–170.
21. *Lau F. C. M. and Chen G.* Optimal layouts of midimew networks. IEEE Trans. Parallel Distrib. Syst., 1996, vol. 7, no. 9, pp. 954–961.
22. *Lezhnev E., Zunin V., Amerikanov A., and Romanov A.* Electronic computer-aided design for low-level modeling of networks-on-chip. IEEE Access, 2024, vol. 12, pp. 48750–48763.
23. *Romanov A. Y.* Development of routing algorithms in networks-on-chip based on ring circulant topologies. Heliyon, 2019, vol. 5, no. 4, paper e01516.

СВЕДЕНИЯ ОБ АВТОРАХ

БУЛАВЧУК Александр Михайлович — старший преподаватель Сибирского федерального университета, г. Красноярск. E-mail: **abulavchuk@sfu-kras.ru**

ВЕРДЕНКО Владимир Романович — студент Адыгейского государственного университета, г. Майкоп. E-mail: **vverdenko@gmail.com**

ВОРОНОВ Всеволод Александрович — кандидат технических наук, зав. лабораторией комбинаторной геометрии Кавказского математического центра Адыгейского государственного университета, г. Майкоп; старший научный сотрудник лаборатории комбинаторных и геометрических структур Московского физико-технического института, г. Долгопрудный. E-mail: **v-vor@yandex.ru**

ЗАИКИН Олег Сергеевич — кандидат технических наук, научный сотрудник Математического центра НГУ, ведущий научный сотрудник ИДСТУ СО РАН, г. Иркутск. E-mail: **oleg.zaikin@icc.ru**

ЗОБОВ Антон Игоревич — старший преподаватель РТУ МИРЭА, г. Москва. E-mail: **zobowai@gmail.com**

КУЛАГИН Артем Владимирович — сотрудник ООО «Центр сертификационных исследований», г. Москва. E-mail: **artemcoolag@yandex.ru**

ЛЕЖНЕВ Евгений Владимирович — кандидат технических наук, доцент Национального исследовательского университета «Высшая школа экономики», г. Москва. E-mail: **elezhnev@hse.ru**

МОНАХОВ Олег Геннадьевич — кандидат технических наук, ведущий научный сотрудник Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск. E-mail: **monakhov@rav.sscc.ru**

МОНАХОВА Эмилия Анатольевна — кандидат технических наук, доцент, ведущий научный сотрудник Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск. E-mail: **emilia@rav.sscc.ru**

ПОПКОВ Кирилл Андреевич — доктор физико-математических наук, старший научный сотрудник Института прикладной математики им. М. В. Келдыша РАН, г. Москва. E-mail: **kirill-formulist@mail.ru**

РАЦЕЕВ Сергей Михайлович — доктор физико-математических наук, доцент, профессор кафедры информационной безопасности и теории управления Ульяновского государственного университета, г. Ульяновск. E-mail: **ratseevsm@mail.ru**

РЗАЕВ Эдвард Рамизович — аспирант Национального исследовательского университета «Высшая школа экономики», г. Москва. E-mail: **erzaev@hse.ru**

РОМАНОВ Александр Юрьевич — доктор технических наук, доцент, профессор Национального исследовательского университета «Высшая школа экономики», г. Москва. E-mail: **a.romanov@hse.ru**

ТАРАСОВ Алексей Вячеславович — доктор физико-математических наук, доцент, сотрудник АО «ИТМиВТ», г. Москва. E-mail: **alextar1@mail.ru**

ЧЕРЕДНИК Игорь Владимирович — доцент РТУ МИРЭА, г. Москва. E-mail: **icherednick@mail.ru**