МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2022 № 58

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

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА
«ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА»

Черемушкин А. В., д-р физ.-мат. наук, академик Академии криптографии РФ (главный редактор); Девянин П. Н., д-р техн. наук, чл.-корр. Академии криптографии РФ (зам. гл. редактора); Панкратова И. А., канд. физ.-мат. наук, доц. (отв. секретарь); Агиевич С. В., канд. физ.-мат. наук; Алексеев В. Б., д-р физ.-мат. наук, проф.; Евдокимов А. А., канд. физ.-мат. наук, проф.; Колесникова С. И., д-р техн. наук; Крылов П. А., д-р физ.-мат. наук, проф.; Логачев О. А., д-р физ.-мат. наук, чл.-корр. Академии криптографии РФ; Мясников А. Г., д-р физ.-мат. наук, проф.; Романьков В.А., д-р физ.-мат. наук, проф.; Сафонов К. В., д-р физ.-мат. наук, проф.; Фомичев В. М., д-р физ.-мат. наук, проф.; Харин Ю. С., д-р физ.-мат. наук, чл.-корр. НАН Беларуси; Чеботарев А. Н., д-р техн. наук, проф.; Шоломов Л. А., д-р физ.-мат. наук, проф.

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 28.11.2022. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 14,6. Тираж 300 экз.
Заказ № 5253. Цена свободная. Дата выхода в свет 13.12.2022.

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

Малыгина Е. С., Кунинец А. А. Анализ минимального расстояния АГ-кода, ассоциированного с максимальной кривой рода три 5
Михайлов В. Г., Меженная Н. М. О точности нормальной аппроксимации для распределения числа повторений в стационарной дискретной случайной последовательности 15
Novoselov S. A. On ideal class group computation of imaginary multiquadratic fields 22
Shevlyakov A. Direct powers of algebraic structures and equations 31

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

Дмух А. А., Пасько Д. О. О поиске разностных соотношений для подстановки Alzette с максимальным или близким к нему значением разностной характеристики. 40
Кущинская Л. А. Об одном методе построения однородных плоскостных аппроксимаций фильтрующего генератора 57

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

Иванов Б. Н. Прямой метод вычисления циклов ячеек карты блока простого планарного графа 69
Лобов А. А., Абросимов М. Б. О единственности минимального рёберного 1-расширения гиперкуба Q_4 84
Монахова Э. А., Монахов О. Г. Эффективный алгоритм поиска кратчайших путей в плотных Гауссианских сетях 94

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

Рыбалов А. Н. О генерической сложности проблемы разбиения графа на треугольники 105

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

Булавчук А. М., Семенова Д. В. Применение методов идемпотентной алгебры в генетическом алгоритме для решения задачи календарного планирования 112

СВЕДЕНИЯ ОБ АВТОРАХ 125

CONTENTS

THEORETICAL BACKGROUNDS OF APPLIED DISCRETE MATHEMATICS

Malygina E. S., Kuninets A. A. Analysis of minimal distance of AG-code associated with maximal curve of genus three 5

Mikhailov V. G., Mezhennaya N. M. About the rate of normal approximation for the distribution of the number of repetitions in a stationary discrete random sequence 15

Novoselov S. A. On ideal class group computation of imaginary multiquadratic fields 22

Shevlyakov A. Direct powers of algebraic structures and equations 31

MATHEMATICAL METHODS OF CRYPTOGRAPHY

Dmukh A. A., Pasko D. O. Search for differences for Alzette S-Box with maximum or close to maximum differential characteristic probability 40

Kuschinskaya L. A. The method for constructing uniform planar approximations of the filter generator 57

APPLIED GRAPH THEORY

Ivanov B. N. A direct method for calculating cell cycles of a block map of a simple planar graph 69

Lobov A. A., Abrosimov M. B. About uniqueness of the minimal 1-edge extension of hypercube Q_4 84

Monakhova E. A., Monakhov O. G. Effective algorithm for finding shortest paths in dense Gaussian networks 94

MATHEMATICAL BACKGROUNDS OF INFORMATICS AND PROGRAMMING

Rybalov A. N. The generic complexity of the graph triangulation problem 105

COMPUTATIONAL METHODS IN DISCRETE MATHEMATICS

Bulavchuk A. M., Semenova D. V. Application of idempotent algebra methods in genetic algorithm for solving the scheduling problem 112

BRIEF INFORMATION ABOUT THE AUTHORS 125

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 519.17 DOI 10.17223/20710410/58/1

АНАЛИЗ МИНИМАЛЬНОГО РАССТОЯНИЯ АГ-КОДА, АССОЦИИРОВАННОГО С МАКСИМАЛЬНОЙ КРИВОЙ РОДА ТРИ[1]

Е. С. Малыгина, А. А. Кунинец

Балтийский федеральный университет им. И. Канта, г. Калининград, Россия

E-mail: emalygina@kantiana.ru, artkuninets@yandex.ru

Рассматривается класс алгебро-геометрических кодов, ассоциированных с максимальной кривой рода три. С помощью аппарата функциональных полей устанавливается вид и степень дивизоров, участвующих в построении кода, при которых код является или не является MDS-кодом.

Ключевые слова: *алгебро-геометрический код, минимальное расстояние кода, mds-код, максимальная кривая, функциональное поле, дивизор.*

ANALYSIS OF MINIMAL DISTANCE OF AG-CODE ASSOCIATED WITH MAXIMAL CURVE OF GENUS THREE

E. S. Malygina, A. A. Kuninets

Immanuel Kant Baltic Federal University, Kaliningrad, Russia

We consider a class of algebraic geometry codes associated with a maximal curve of genus three whose number of rational points satisfies the upper Hasse — Weil — Serre bound. It is proved that the number of rational points of such curve is odd and has a classification: the first type includes 4-tuples of conjugate points of multiplicity 1, the second type includes couples conjugate points of multiplicity 2, and the third type includes a single point of multiplicity 4. It is found out for which types of points the divisor of the functional field of the desired curve and consisting of these points is the principle. We consider special cases when $\deg(G) = 2, 4$, and establish the form of a divisor D when AG-code $\mathcal{C}_{\mathscr{L}}(D, G)$ associated with the divisors D and G is MDS-code. It is shown that the AG-code $\mathcal{C}_{\mathscr{L}}(D, G)$ is not an MDS-code if the divisor $D - G$ is principle and $\deg(G) \geqslant 5$. Also, it is proved that $\mathcal{C}_{\mathscr{L}}(D, G)$ is an MDS-code if the divisor D consists only of the first type points of curve conjugated to each other for $\deg(D) \geqslant 8$ and $G = \dfrac{\deg(D) + 2}{2} P_\infty$. Finally, it is shown that the dual equivalent code $\mathcal{C}_{\mathscr{L}}(D, H)^\perp$ to the code $\mathcal{C}_{\mathscr{L}}(D, G)$, which is not MDS, will also not be MDS with conditions $\deg(D) - \alpha < \deg(H) < \deg(D)$, $4 < \deg(G) < \alpha + 4$, $5 < \alpha < \deg(D) - 5$, and D consists only of conjugate points of the first type.

Keywords: *algebraic geometry code, minimal distance, mds-code, maximal curves, function field, divisor.*

[1]Работа выполнена за счёт гранта Российского научного фонда №22-41-0441 (`https://rscf.ru/project/22-41-04411/`).

Введение

Рассмотрим класс алгебро-геометрических кодов, ассоциированных с максимальной кривой рода три. Под линейным кодом $\mathcal{C}$ над простым конечным полем $\mathbb{F}_p$ характеристики p понимаем линейное подпространство векторного пространства $\mathbb{F}_p^n$. Обозначая за n длину кодового слова пространства $\mathcal{C}$, а за $k = [\mathcal{C} : \mathbb{F}_p]$ — его размерность, считаем, что код $\mathcal{C}$ имеет параметры $[n, k]$. Кроме того, на векторном пространстве $\mathcal{C}$ в качестве метрики задаётся вес Хэмминга, то есть число ненулевых координат кодового слова. Ещё одним важным параметром кода является его минимальное расстояние $d(\mathcal{C})$ — минимальный вес Хэмминга среди всех весов Хэмминга кодовых слов кода $\mathcal{C}$. Оно позволяет определить число ошибок $t = \lfloor (d-1)/2 \rfloor$, которое может быть исправлено кодом.

В 1981 г. В. Д. Гоппа в [1] представил класс кодов, ассоциированных с алгебраическими кривыми над конечными полями, которые получили название алгебро-геометрических кодов (АГ-коды). Для построения АГ-кода будем использовать алгебраическую кривую, число точек которой достигает верхней границы Хассе — Вейля — Серра. Такие кривые называются максимальными и имеют большое число рациональных точек. Соответственно чем выше отношение числа точек к роду кривой, тем лучшими характеристиками обладает построенный на кривой код.

Исследование АГ-кодов примечательно тем, что, применяя аппарат теории функциональных полей, зачастую можно вычислить точные значения параметров кода, построенного на алгебраической кривой. Таким образом, анализ минимального расстояния кода проведём с использованием дивизоров функционального поля, ассоциированного с максимальной кривой. Основное преимущество рассмотрения в качестве базовых инструментов исследования именно функциональных полей, а не геометрии алгебраических кривых, обусловлено тем, что функциональные поля кривых инвариантны относительно бирациональных изоморфизмов кривых.

Вычисление значения размерности k АГ-кода можно осуществить посредством вычисления размерности линейного пространства Римана — Роха, ассоциированного с некоторыми дивизорами кривой. Известная теорема Римана — Роха [2] во многих случаях позволяет сделать это точно. В общем случае минимальное расстояние кода удовлетворяет следующим неравенствам: $n+1-g-k \leqslant d \leqslant n+1-k$. Задача вычисления точного значения минимального расстояния АГ-кодов затрагивает исследование так называемых MDS-кодов (Maximum Distance Separable codes), минимальное расстояние которых достигает границы Синглтона, т. е. $d = n+1-k$. В работе представлен анализ некоторых случаев, когда АГ-коды являются/не являются MDS-кодами в зависимости от задания дивизоров функционального поля кривой, с которыми этот код ассоциирован.

В п. 1 представлены предварительные сведения из теории функциональных полей, особое внимание уделено краткой теории дивизоров, поскольку именно они лежат в основе исследования. Пункт 2 посвящён описанию максимальной кривой рода три, на базе которой строится АГ-код, и её точек. В п. 3 отражён основной результат работы. Найдены ограничения на степени дивизоров D и G, ассоциированных с АГ-кодом $\mathcal{C}_{\mathscr{L}}(D, G)$, относительно которых показано, когда $\mathcal{C}_{\mathscr{L}}(D, G)$ является MDS-кодом, а когда нет.

1. Предварительные сведения из теории функциональных полей

На протяжении всех предварительных сведений под K будем понимать произвольное поле, при необходимости внося уточнения.

Определение 1. Алгебраическим функциональным полем F/K от одной переменной называется расширение F поля K, такое, что F является конечным алгебраическим расширением $K(x)$ для некоторого элемента $x \in F$, являющегося трансцендентным над K.

Для краткости будем называть F/K просто функциональным полем. Точками функционального поля являются максимальные идеалы кольца нормирования поля F, определение которого можно посмотреть в [2, Definition 1.1.4]. Под кольцом нормирования точки P будем понимать следующее множество:

$$\mathcal{O}_P = \{z \in F : z^{-1} \notin P\}.$$

Множество всех точек функционального поля F/K будем обозначать $\mathbb{P}_F$. Степень точки P функционального поля F/K определим как степень расширения полей $\deg P = [\mathcal{O}_P/P : K]$.

Напомним, что, согласно [2], всякий элемент $0 \neq z \in F$ имеет единственное представление $z = t^n u$, если $P = t\mathcal{O}_P$, $u \in \mathcal{O}_P^*$ и $n \in \mathbb{Z}$.

Определение 2. С каждой точкой $P \in \mathbb{P}_F$ ассоциируем функцию

$$v_P : F \to \mathbb{Z} \cup \{\infty\},$$

которая играет роль дискретного нормирования функционального поля F/K:

$$v_P(z) = n \ \text{ для } \ z = t^n u, \quad v_P(0) = \infty.$$

Считаем, что точка P имеет нуль в z тогда и только тогда, когда $v_P(z) > 0$, и имеет полюс в z тогда и только тогда, когда $v_P(z) < 0$.

Определение 3. Абелева группа $\mathcal{D}_F$, порождённая точками функционального поля F/K, называется группой дивизоров поля F/K.

Элементы группы $\mathcal{D}_F$ называются дивизорами поля F/K. Дивизор представляет собой формальную сумму точек

$$D = \sum_{P \in \mathbb{P}_F} n_P P, \ \text{ где } \ n_P \in \mathbb{Z} \ \text{ и почти все } \ n_P = 0.$$

Носителем дивизора D является множество

$$\operatorname{supp}(D) = \{P \in \mathbb{P}_F : n_P \neq 0\}.$$

Для $P \in \mathbb{P}_F$ и дивизора D определим $v_P(D) = n_P$. Таким образом,

$$D = \sum_{P \in \operatorname{supp}(D)} v_P(D)\, P.$$

На $\mathcal{D}_F$ определено также частичное упорядочивание, а именно:

$$D_1 \leqslant D_2 \ \Leftrightarrow \ v_P(D_1) \leqslant v_P(D_2) \ \text{ для всех } \ P \in \mathbb{P}_F.$$

Степень дивизора определяется следующим образом:

$$\deg(D) = \sum_{P \in \mathbb{P}_F} v_P(D)\, \deg P.$$

Определение 4. Пусть $f \in F \setminus \{0\}$. Обозначим через Z (через N) множество нулей (полюсов) функции f в $\mathbb{P}_F$. Тогда для f определим её дивизор нулей:

$$(f)_0 = \sum_{P \in Z} v_P(f)\, P;$$

дивизор полюсов:

$$(f)_\infty = \sum_{P \in N} (-v_P(f))\, P;$$

главный дивизор:

$$(f) = (f)_0 - (f)_\infty.$$

Отметим также, что

$$\deg(f)_0 = \deg(f)_\infty = [F : K(f)].$$

Далее определим отношение эквивалентности в группе дивизоров $\mathcal{D}_F$.

Определение 5. Множество дивизоров вида

$$Princ_F = \{(x) : 0 \neq x \in F\}$$

называется группой главных дивизоров поля F/K.

Несложно проверить, что $Princ_F$ действительно является подгруппой группы $\mathcal{D}_F$ относительно операции сложения $(f) + (g) = (fg)$ для $f, g \in F \setminus \{0\}$.

Определение 6. Фактор-группа

$$Cl_F = \mathcal{D}_F / Princ_F$$

называется группой классов дивизоров поля F/K. Элементами Cl_F являются классы $[D]$, задаваемые представителями D. Будем говорить, что два дивизора D и D' из $\mathcal{D}_F$ эквивалентны ($D \sim D'$), если имеет место равенство $D = D' + (f)$ для некоторой ненулевой функции $f \in F$.

Дадим определение пространства Римана — Роха, одного из главных понятий в теории функциональных полей.

Определение 7. Для дивизора $D \in \mathcal{D}_F$ пространством Римана — Роха называется множество вида

$$\mathscr{L}(D) = \{f \in F : (f) \geqslant -D\} \cup \{0\}.$$

Отметим, что $\mathscr{L}(D)$ является конечномерным векторным пространством над полем K, при этом $\dim D = \dim \mathscr{L}(D)$ называется размерностью дивизора D.

Определение 8. Род функционального поля F/K определён следующим образом:

$$g = \max\{\deg(D) - \dim(D) + 1 : D \in \mathcal{D}_F\}.$$

Отметим также, что благодаря некоторым ограничениям, накладываемым на степень дивизора, мы можем точно вычислить размерность пространства Римана — Роха, ассоциированного с этим дивизором, или для простоты размерность самого дивизора. Формула для подсчёта размерности непосредственно следует из известной теоремы Римана — Роха:

Теорема 1 [2, Theorem 1.5.15]. Если D — дивизор F/K и $\deg(D) \geqslant 2g - 1$, то

$$\dim(D) = \deg(D) + 1 - g.$$

2. Максимальная кривая рода три

Определение 9. Пусть C — гладкая неприводимая проективная кривая рода g, определённая над конечным полем $\mathbb{F}_p$. Назовем кривую C оптимальной, если число её рациональных точек удовлетворяет границе Хассе — Вейля — Серра

$$\#C(\mathbb{F}_p) = p + 1 \pm g(C)\lfloor 2\sqrt{p} \rfloor.$$

В случае «−» кривая называется минимальной, в случае «+» — максимальной.

Будем рассматривать максимальные кривые рода три. Максимальность обеспечит построение длинных кодов, а значение рода $g(C) = 3$ позволит задать кривую явно, согласно [3, Предложение 3.2.1]:

$$\begin{cases} z^2 = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \beta_0 y, \\ y^2 = x^3 + ax + b, \end{cases} \tag{1}$$

где $\alpha_0, \alpha_1, \alpha_2, \beta_0, a, b \in \mathbb{F}_p$.

Такие кривые существуют над определёнными конечными полями, а именно над полями с дискриминантами из множества $\{-19, -43, -67, -163\}$ [3]. Под дискриминантом конечного поля $\mathbb{F}_p$ будем понимать число

$$d(\mathbb{F}_p) = \lfloor 2\sqrt{p} \rfloor^2 - 4p.$$

Мы можем упростить уравнение (1), сведя его к переменным x и z. Выражая из первого уравнения системы значение $y = (z^2 - \alpha_2 x^2 - \alpha_1 x - \alpha_0)/\beta_0$ и подставляя его во второе уравнение, получаем

$$z^4 + (-2\alpha_2 x^2 - 2\alpha_1 x - 2\alpha_0)z^2 + (\alpha_2^2 x^4 + (2\alpha_1\alpha_2 - \beta_0^2)x^3 + (2\alpha_0\alpha_2 + \alpha_1^2)x^2 + \\ + (2\alpha_0\alpha_1 - \beta_0^2 a)x + (\alpha_0^2 - \beta_0^2 b)) = 0.$$

Очевидно, что при заданном значении $x \in \mathbb{F}_p$ уравнение может иметь четыре различных простых корня, либо два различных корня кратности 2 каждый, или один корень кратности 4. Сопоставим эти корни точкам кривой согласно трём типам:

- если уравнение имеет четыре различных корня, соответствующих точкам P_1, P_2, P_3 и P_4, то будем относить эти точки к типу I;
- если уравнение имеет два различных корня кратности 2, соответствующих точкам Q_1 и Q_2, то будем относить эти точки к типу II;
- если уравнение имеет единственный корень кратности 4, соответствующий точке P, то будем относить эту точку к типу III.

Отметим, что число рациональных точек рассматриваемой кривой является нечётным.

Утверждение 1. Число рациональных точек кривой $C/\mathbb{F}_p$ всегда нечётно и имеет вид $4s_1 + 2s_2 + 1$.

Доказательство. Согласно границе Хассе — Вейля — Серра, число точек максимальной кривой удовлетворяет условию

$$|C(\mathbb{F}_p)| = p + 1 + 3\lfloor 2\sqrt{p} \rfloor.$$

Мы рассматриваем кривые над конечными полями с нечётными дискриминантами $d = (\lfloor 2\sqrt{p} \rfloor)^2 - 4p$. Учитывая, что характеристика поля p — нечётное число, имеем, что $\lfloor 2\sqrt{p} \rfloor$ нечётно, откуда $|C(\mathbb{F}_p)|$ нечётно.

Принимая во внимание типы точек, получаем

$$|C(\mathbb{F}_p)| = n = 4s_1 + 2s_2 + 1,$$

где s_1— число четвёрок типа I; s_2— число двоек типа II. Заметим, что точка с координатой $(x, 0)$ у рассматриваемой кривой всегда одна. ■

Замечание 1. Отметим, что бесконечно удалённая точка кривой C, которую обозначим P_∞, не является $\mathbb{F}_p$-рациональной.

Согласно [3, Предложение 3.1.2], оптимальная кривая C рода 3 над $\mathbb{F}_p$ является двойным накрытием оптимальной эллиптической кривой. Обозначив это накрытие $f : C \to E$, получаем, что при отображении f бесконечно удалённая точка P_∞ кривой C лежит над бесконечно удалённой точкой ∞ эллиптической кривой E, которая, в свою очередь, лежит над бесконечной точкой проективной прямой $\mathbb{P}^1$. При этом $\deg P_\infty = 2$, что соответствует степени накрытия.

3. Основной результат

Пусть $F = \mathbb{F}_p(C)$ — функциональное поле кривой, заданной уравнением (1), $P_1, P_2, \ldots, P_n$ — попарно различные точки поля $F/\mathbb{F}_p$ степени один и $D = P_1 + \ldots + P_n$. Напомним, что P_∞ — бесконечно удалённая точка поля $F/\mathbb{F}_p$ степени два. Обозначим через G дивизор поля $F/\mathbb{F}_p$, кратный P_∞, тогда $\operatorname{supp}(G) \cap \operatorname{supp}(D) = \varnothing$.

Определение 10. АГ-кодом $\mathcal{C}_{\mathscr{L}}(D, G)$, ассоциированным с дивизорами D и G, является образ гомоморфизма:

$$ev : \begin{cases} \mathscr{L}(G) \to \mathbb{F}_p^n, \\ f \mapsto (f(P_1), \ldots, f(P_n)). \end{cases}$$

Очевидно, что ядро этого гомоморфизма $\operatorname{Ker}(ev) = \mathscr{L}(G - D)$. Если $\deg G < n$, то $\mathscr{L}(G - D) = \{0\}$. Следовательно, ev инъективно и

$$\dim \mathcal{C}_{\mathscr{L}}(D, G) = \dim(G) = \dim \mathscr{L}(G) = \deg(G) + 1 - g = \deg(G) - 2$$

при условии, что $\deg(G) \geqslant 2g - 1 = 5$. Здесь используем утверждение теоремы 1.

Отметим, что, согласно [2, Theorem 2.2.2], $\mathcal{C}_{\mathscr{L}}(D, G)$ является $[n, k, d]$-кодом с параметрами

$$k = \dim(G) - \dim(G - D) \quad \text{и} \quad d \geqslant n - \deg(G).$$

Для вычисления минимального расстояния $d = d(\mathcal{C})$ проведём следующие рассуждения. Пусть $c \in \mathcal{C}_{\mathscr{L}}(D, G)$ — кодовое слово веса δ. Тогда существует функция $f \in \mathscr{L}(G)$, такая, что $ev(f) = c$ и f обнуляется в точках $\{P_i : i \in I\}$ для некоторого $I \subseteq \{1, \ldots, n\}$ и $|I| = n - \delta$.

Если изначально рассмотреть $f \in \mathscr{L}(G - \sum_I P_i) \subset \mathscr{L}(G)$, то по определению пространства Римана — Роха имеем соотношение $(f) \geqslant \sum\limits_I P_i - G$. Кроме того, $\deg\left(\sum\limits_I P_i - G\right) = n - \delta - \deg(G)$. Если $\deg(G) = n - \delta$, то $\deg\left(\sum\limits_I P_i - G\right) = 0$ и $(f) = \sum\limits_I P_i - G$. Соответственно $\sum\limits_I P_i - G$ — главный дивизор.

Поскольку $G - \sum\limits_I P_i < G$, то $\dim\left(G - \sum\limits_I P_i\right) < \dim(G)$. Тогда, очевидно,

$$\dim\left(G - \sum_I P_i\right) + d(\mathcal{C}) < \dim(G) + d(\mathcal{C}) \leqslant n + 1.$$

Согласно вышесказанному и принимая во внимание, что $\deg(G) \geqslant 5$ и $\deg(G) = |\{P_i\}|$, имеем следующее утверждение:

Утверждение 2. Обозначим $D = \sum\limits_{I} P_i$, где $I \subseteq \{1, \ldots, n\}$. Если дивизор $D - G$ является главным при условии, что $\deg(G) \geqslant 5$, то код $\mathcal{C}_{\mathscr{L}}(D, G)$ не является MDS-кодом.

Выясним, какие дивизоры в функциональном поле кривой $F/\mathbb{F}_p$ являются главными.

Лемма 1. Пусть $F = \mathbb{F}_p(C)$ — функциональное поле оптимальной кривой C рода три. Тогда

1) Точка P является точкой типа III тогда и только тогда, когда дивизор $4P - 2P_\infty$ является главным.
2) Точки Q_1 и Q_2 являются точками типа II тогда и только тогда, когда дивизор $2(Q_1 + Q_2) - 2P_\infty$ является главным.
3) Точки P_1, P_2, P_3 и P_4 являются точками типа I тогда и только тогда, когда дивизор $P_1 + P_2 + P_3 + P_4 - 2P_\infty$ является главным.

Доказательство. Отметим, что для $f \in F \setminus \{0\}$ выполняется

$$\deg(f)_0 = \deg(f)_\infty = [F : \mathbb{F}_p(x)] = 4.$$

Приведём доказательство только первого случая, поскольку для оставшихся случаев рассуждения аналогичны.

Достаточность. По условию существует функция $f \in F$, такая, что её главный дивизор имеет вид

$$(f) = 4P - 2P_\infty.$$

Вычислим $\dim(2P_\infty)$. Учитывая, что кривая C не является гиперэллиптической [3, Лемма 3.1.3] и тот факт, что в гиперэллиптическом случае, если $\deg(P_\infty) = 2$, то $\dim(P_\infty) \geqslant 2$ [2, Lemma 6.2.2], получаем $\dim(P_\infty) = 1$. Согласно теореме Клиффорда [2, Theorem 1.6.13], имеем

$$\dim(2P_\infty) \leqslant 1 + \frac{1}{2}\deg(2P_\infty) = 3.$$

Определение оптимальной кривой уравнением (1) соответствует случаю $\dim(2P_\infty) = 3$ [3]. Тогда в качестве f можем рассмотреть функцию $f \in \mathscr{L}(2P_\infty) = \langle 1, x, z\rangle$. Следовательно, $f \in \mathbb{F}_p(x, z)$ и $4P$ является дивизором нулей функции f в $\mathbb{F}_p(x, z)$, а значит, P — нуль функции f с кратностью 4. Таким образом, точка P относится к типу III.

Необходимость. Если точка P над $\mathbb{F}_p(x)$ относится к типу III, значит, P является нулём некоторой функции $f \in \mathbb{F}_p(x, z)$ кратности 4. Тогда дивизор $4P - 2P_\infty$ имеет нулевую степень и по построению является главным, т. е. $(f) = 4P - 2P_\infty$. ■

Замечание 2. Отметим, что различные линейные комбинации представленных выше главных дивизоров также дадут главный дивизор.

Замечание 3. Поскольку в утверждении 2 в записи дивизора D фигурируют точки без учёта их кратности, то D является главным, если P_i, $1 \leqslant i \leqslant 4$, относятся к типу I и образуют четвёрки сопряжённых точек.

Следовательно, $\mathcal{C}_{\mathscr{L}}(D, G)$ является MDS-кодом, если дивизор $D - G$ не является главным, причём дивизор $D = \sum\limits_{I} P_i$, $I \subseteq \{1, \ldots, n\}$, также состоит из четвёрок сопряжённых точек, относящихся к типу I, и $\deg G \geqslant 5$. Однако не ясно, чему в этом случае равна степень дивизора G.

Прежде чем определить вид дивизора G при $\deg(G) \geqslant 5$, чтобы код $\mathcal{C}_{\mathscr{L}}$ являлся MDS-кодом, рассмотрим случаи, когда $\deg(G) < 5$.

Теорема 2. *$\mathcal{C}_{\mathscr{L}}(D, P_\infty)$ является MDS-кодом для некоторого дивизора D степени 2.*

Доказательство. Поскольку $G = P_\infty$, то $\deg(G) = 2$ и $\dim(G) = 1$. Следовательно, базис пространства Римана — Роха $\mathscr{L}(G)$ состоит из единственной функции $\{1\}$. В качестве дивизора D рассмотрим $D = P_1 + \ldots + P_n$, где P_i, $1 \leqslant i \leqslant n$, — точки степени один любого типа. Соответственно порождающая матрица кода имеет вид

$$(\,1(P_1) \quad \ldots \quad 1(P_n)\,) = (\,1 \quad \ldots \quad 1\,).$$

Очевидно, $k = 1$ и вес единственной строки матрицы равен числу точек n дивизора D. Таким образом, минимальное расстояние кода равно $d(\mathcal{C}) = n$, что есть верхняя граница Синглтона. Следовательно, код $\mathcal{C}_{\mathscr{L}}(D, P_\infty)$ является MDS-кодом. ■

Теорема 3. *$\mathcal{C}_{\mathscr{L}}(D, 2P_\infty)$ является MDS-кодом, если дивизор D имеет следующий вид:*

1) *$D = P_1 + P_2 + P_3 + P_4$, где P_i — четвёрка сопряжённых точек;*
2) *$D = P_1 + P_2 + Q_1 + Q_2$, где P_i и Q_i — пары сопряжённых точек типа I или II;*
3) *$D = P_1 + P_2 + Q + S$, где P_i — пара сопряжённых точек типа I или II; $Q \neq S$ — точки любого типа.*

Доказательство. По условию $G = 2P_\infty$ и $\dim(G) = 3$. Соответственно базис пространства Римана — Роха $\mathscr{L}(G)$ равен $\{1, x, z\}$, тогда порождающая матрица кода $\mathcal{C}_{\mathscr{L}}(D, G)$ имеет вид

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x(P) & x(Q) & x(S) & x(T) \\ z(P) & z(Q) & z(S) & z(T) \end{pmatrix},$$

где $P, Q, S, T \in \operatorname{supp}(D)$.

1) Пусть $D = P_1 + P_2 + P_3 + P_4$. Порождающая матрица кода $\mathcal{C}_{\mathscr{L}}(D, G)$ имеет вид

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x(P_1) & x(P_2) & x(P_3) & x(P_4) \\ z(P_1) & z(P_2) & z(P_3) & z(P_4) \end{pmatrix}.$$

Поскольку $\{P_i\}_{1 \leqslant i \leqslant 4}$ образуют четвёрку сопряжённых точек, то $x(P_1) = x(P_2) = x(P_3) = x(P_4)$ и, следовательно, первая и вторая строки матрицы эквивалентны. Это значит, что размерность кода $k = 2$ и $\operatorname{rank}(M) = 3 = d(\mathcal{C})$, и $\mathcal{C}_{\mathscr{L}}(D, G)$ — MDS-код, так как $d(\mathcal{C}) = n + 1 - k$.

2) Пусть $D = P_1 + P_2 + Q_1 + Q_2$, где (P_1, P_2) и (Q_1, Q_2) — пары сопряжённых точек типа I или II. Порождающая матрица кода $\mathcal{C}_{\mathscr{L}}(D, G)$ имеет вид

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x(P_1) & x(P_2) & x(Q_1) & x(Q_2) \\ z(P_1) & z(P_2) & z(Q_1) & z(Q_2) \end{pmatrix}.$$

Имеем $x(P_1) = x(P_2)$ и $x(Q_1) = x(Q_2)$. Следовательно, $k = 3$ и $\operatorname{rank}(M) = 2 = d(\mathcal{C})$. Тогда $\mathcal{C}_{\mathscr{L}}(D, G)$ — MDS-код, так как $d(\mathcal{C}) = n + 1 - k$.

3) $D = P_1+P_2+Q+S$, где (P_1, P_2) — пара сопряжённых точек типа I или II; $Q \neq S$ — точки любого типа. Порождающая матрица кода $\mathcal{C}_{\mathscr{L}}(D, G)$ имеет вид

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x(P_1) & x(P_2) & x(Q) & x(S) \\ z(P_1) & z(P_2) & z(Q) & z(S) \end{pmatrix}.$$

Поскольку $x(P_1) = x(P_2)$, получаем, что $k = 3$ и $\operatorname{rank}(M) = 2 = d(\mathcal{C})$. Тогда $\mathcal{C}_{\mathscr{L}}(D, G)$ — MDS-код. ■

Замечание 4. Отметим, что теорема 3 работает в том случае, когда третья строка матрицы M состоит из различных элементов.

Замечание 5. Если $D = P_1 + P_2 + P_3 + Q$, где $\{P_i\}_{1 \leqslant i \leqslant 3}$ — сопряжённые точки типа I и Q — точка любого типа, то $k = 3$, $d(\mathcal{C}) = 1$ и $\mathcal{C}_{\mathscr{L}}(D, G)$ не является MDS-кодом.

Теперь сформулируем и докажем теорему для случая, когда D состоит из четвёрок сопряжённых точек типа I, а $\deg(G) \neq 2, 4, n$.

Теорема 4. $\mathcal{C}_{\mathscr{L}}(D, G)$ является MDS-кодом, если дивизор D состоит только из четвёрок сопряжённых точек типа I, $\deg(D) = n$, $n \geqslant 8$ и $G = \dfrac{n+2}{2} P_\infty$.

Доказательство. Поскольку число точек n в записи дивизора D равно 0 по модулю 4, то $n = 4k$ для $k \in \mathbb{N}$ и

$$D = P_1^{(1)} + P_2^{(1)} + P_3^{(1)} + P_4^{(1)} + \ldots + P_1^{(k)} + P_2^{(k)} + P_3^{(k)} + P_4^{(k)},$$

где все точки $P_i^{(j)}$, $1 \leqslant i \leqslant 4$, $1 \leqslant j \leqslant k$, являются точками типа I.

Согласно утверждению 2, $\mathcal{C}_{\mathscr{L}}(D, G)$ не является MDS-кодом, если $\deg(G) = n$, поскольку в этом случае дивизор $D - G$ является главным.

Определим, при каком G код $\mathcal{C}_{\mathscr{L}}(D, G)$ будет являться MDS-кодом. Так как

$$\frac{n}{2} P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) \leqslant \left(\frac{n}{2} + 1\right) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}),$$

то

$$\mathscr{L}\left(\frac{n}{2} P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)})\right) \subseteq \mathscr{L}\left(\left(\frac{n}{2} + 1\right) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)})\right).$$

Здесь мы рассматриваем ближайшее по включению пространство, учитывая, что $\dim \mathscr{L}(P_\infty) = 1$. Соответственно

$$\frac{n}{2} P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) = \left(\frac{n}{2} + 1\right) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) - P_\infty,$$

откуда имеем

$$(2k+1) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) = 2k P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) + P_\infty.$$

Поскольку дивизор $2k P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)})$ главный, получаем

$$(2k+1) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)}) \sim P_\infty.$$

Так как P_∞ не является главным дивизором, то дивизор $(2k+1) P_\infty - (P_1^{(1)} + \ldots + P_4^{(k)})$ также неглавный. Следовательно, код $\mathcal{C}_{\mathscr{L}}(D, G)$ является MDS-кодом при $D = P_1^{(1)} + \ldots + P_4^{(k)}$ и $G = \left(\dfrac{n+2}{2}\right) P_\infty$. ■

Рассмотрим более общий случай, когда степень дивизора, с помощью которого вычисляется размерность кода, меньше числа точек дивизора D.

Теорема 5. Пусть $n \geqslant 12$, $\alpha \in \mathbb{N}$, $5 < \alpha < n-5$ и пусть задан дивизор $D = P_1^{(1)} + \ldots + P_4^{(k)}$, где $P_i^{(j)}$ образуют четвёрки сопряжённых точек типа I и $n = 4k$, а также задан дивизор G, где $\operatorname{supp}(G) \cap \operatorname{supp}(D) = \varnothing$ и $4 < \deg(G) < \alpha + 4$. Будем считать, что $\mathcal{C}_{\mathscr{L}}(D, G)$ не является MDS-кодом. Тогда существует дивизор H функционального поля F, такой, что $\operatorname{supp}(H) \cap \operatorname{supp}(D) = \varnothing$ и $n - \alpha < \deg(H) < n$, и код $\mathcal{C}_{\mathscr{L}}(D, H)$ не является MDS-кодом.

Доказательство. От противного. Предположим, что существует дивизор H, удовлетворяющий условиям теоремы, но код $\mathcal{C}_{\mathscr{L}}(D, H)$ является MDS-кодом. Согласно [4, Theorem 2.5], существуют дивизор G, $\operatorname{supp}(G) \cap \operatorname{supp}(D) = \varnothing$ и $0 \leqslant \deg G \leqslant n$, и вектор $a \in \mathbb{F}_p^n$ веса n, такие, что

$$a \cdot \mathcal{C}_{\mathscr{L}}(D, G) = \mathcal{C}_{\mathscr{L}}(D, H)^{\perp}.$$

Для более детального ознакомления со структурой дуального кода см. [2]. Здесь мы только отметим, что если $\mathcal{C}_{\mathscr{L}}(D, H)$ — дуальный код к $\mathcal{C}_{\mathscr{L}}(D, G)$, то

$$\dim(G) + \dim(H) = k.$$

Если $\mathcal{C}_{\mathscr{L}}(D, H)$ — MDS-код, то $\mathcal{C}_{\mathscr{L}}(D, H)^{\perp}$ также является MDS-кодом тогда и только тогда, когда $a \cdot \mathcal{C}_{\mathscr{L}}(D, G)$ — MDS-код, а это выполняется тогда и только тогда, когда $\mathcal{C}_{\mathscr{L}}(D, G)$ — MDS-код.

При этом если $\deg(G) \geqslant 5$ и $\deg(H) \geqslant 5$, что выполняется по условию, то

$$\deg(G) - 2 = \dim(G) = n - \dim(H) = n - (\deg(H) - 2) = n + 2 - \deg(H).$$

Следовательно, $\deg(H) = n + 4 - \deg(G)$. Поскольку $n - \alpha < \deg(H) < n$, то

$$n - \alpha < n + 4 - \deg(G) < n,$$

откуда

$$4 < \deg(G) < \alpha + 4,$$

что противоречит тому, что $\mathcal{C}_{\mathscr{L}}(D, G)$ не является MDS-кодом. ■

ЛИТЕРАТУРА

1. *Goppa V. D.* Geometry and Codes. Kluwer Academic Publishers, 1988.
2. *Stichtenoth H.* Algebraic Function Fields and Codes. Springer Verlag, 1991.
3. *Алексеенко Е. С.* Явные конструкции оптимальных кривых рода три: дис. ... канд. физ.-мат. наук. М., 2016. `http://iitp.ru/upload/content/1203/AES_disser.pdf`.
4. *Stichtenoth H.* Self dual Goppa codes // J. Pure Appl. Algebra. 1988. V. 55. P. 199–211.

REFERENCES

1. *Goppa V. D.* Geometry and Codes. Kluwer Academic Publishers, 1988.
2. *Stichtenoth H.* Algebraic Function Fields and Codes. Springer Verlag, 1991.
3. *Alekseenko E. S.* Yavnye konstruktsii optimal'nykh krivykh roda tri [Explicit Constructions of Optimal Curves of Genus Three]. PHD-thesis, Moscow, 2016. `http://iitp.ru/upload/content/1203/AES_disser.pdf`.
4. *Stichtenoth H.* Self dual Goppa codes. J. Pure Appl. Algebra, 1988, vol. 55, pp. 199–211.

УДК 519.214 DOI 10.17223/20710410/58/2

О ТОЧНОСТИ НОРМАЛЬНОЙ АППРОКСИМАЦИИ ДЛЯ РАСПРЕДЕЛЕНИЯ ЧИСЛА ПОВТОРЕНИЙ В СТАЦИОНАРНОЙ ДИСКРЕТНОЙ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

В. Г. Михайлов*, Н. М. Меженная**

*Математический институт им. В. А. Стеклова Российской академии наук, г. Москва, Россия

**Московский государственный технический университет им. Н. Э. Баумана, г. Москва, Россия

E-mail: mikhail@mi-ras.ru, natalia.mezhennaya@gmail.com

Изучается задача об асимптотической нормальности числа r-кратных повторений знаков в отрезке стационарной (в узком смысле) дискретной случайной последовательности на множестве $\{1, 2, \ldots, N\}$, обладающей свойством равномерно сильного перемешивания. Показано, что в случае, когда коэффициент равномерно сильного перемешивания $\varphi(t)$ при произвольно заданном $\alpha > 0$ убывает как $t^{-6-\alpha}$, расстояние в равномерной метрике между функцией распределения числа повторений и функцией распределения стандартного нормального закона с увеличением длины последовательности n убывает со скоростью $O(n^{-\delta})$ для любого $\delta \in (0; \alpha(32 + 4\alpha)^{-1}))$.

Ключевые слова: *нормальная аппроксимация, число кратных повторений, стационарная случайная последовательность, равномерно сильное перемешивание, расстояние в равномерной метрике.*

ABOUT THE RATE OF NORMAL APPROXIMATION FOR THE DISTRIBUTION OF THE NUMBER OF REPETITIONS IN A STATIONARY DISCRETE RANDOM SEQUENCE

V. G. Mikhailov*, N. M. Mezhennaya**

*Steklov Mathematical Institute of Russian Academy of Sciences, Moscow, Russia

**Bauman Moscow State Technical University, Moscow, Russia

The paper presents the problem of asymptotic normality of the number of r-fold repetitions of characters in a segment of a (strictly) stationary discrete random sequence on the set $\{1, 2, \ldots, N\}$ with the uniformly strong mixing property. It is shown that in the case when the uniformly strong mixing coefficient $\varphi(t)$ for an arbitrarily given $\alpha > 0$ decreases as $t^{-6-\alpha}$, then the distance in the uniform metric between the distribution function of the number of repetitions and the distribution function of the standard normal law decreases at a rate of $O(n^{-\delta})$ with increasing sequence length n for any $\delta \in (0; \alpha(32 + 4\alpha)^{-1}))$.

Keywords: *normal approximation, number of multiple repetitions, stationary random sequence, uniformly strong mixing, distance in uniform metric.*

Введение

В настоящее время большую роль играет задача генерации случайных или псевдослучайных последовательностей как основы имитационного моделирования изучаемых процессов в науке и технике, что делает необходимой проверку соответствия их свойств требованиям модели. Особенно важно это для криптографических применений. Для такой проверки в криптографической литературе разработаны наборы статистических тестов [1, 2].

Например, отдельно изучены свойства статистик, основанных на числах повторений знаков, в том числе кратных, в независимых и конечно зависимых последовательностях случайных величин. Наиболее успешными оказались исследования условий сходимости распределений чисел повторений к распределению Пуассона [3, 4]. В качестве примера построения достаточных условий асимптотической нормальности числа повторений следует привести работу А. М. Шойтова [5].

В последние годы активно велись исследования повторяемости знаков в дискретных цепях Маркова. Например, в работах [6–8] исследована возможность аппроксимации распределения чисел повторений цепочек в цепи Маркова распределением Пуассона. Аналогичная задача о доказательстве центральной предельной теоремы для таких случайных величин решена в [9]. В частности, в [9, теорема 3] установлена оценка скорости сходимости к нормальному закону для числа кратных совпадений знаков в конечной стационарной цепи Маркова. В настоящей работе показано, что вместо цепей Маркова можно рассматривать более широкий класс — стационарные случайные последовательности, удовлетворяющие условию равномерно сильного перемешивания.

1. Оценка скорости сходимости в центральной предельной теореме

Пусть $X_1, \ldots, X_n$ — отрезок стационарной (в узком смысле) случайной последовательности со значениями из множества $\{1, \ldots, N\}$, стационарным распределением $\mathsf{P}[X_1 = k] = p_k \in (0; 1)$, $k = 1, \ldots, N$, $p_1 + \ldots + p_N = 1$, удовлетворяющей условию равномерно сильного перемешивания.

Напомним [10, с. 361], что процесс $\{X_t\}_{-\infty}^{\infty}$ называется стационарным (в узком смысле), если распределения случайных векторов вида $(X_{t_1+h}, \ldots, X_{t_s+h})$ при всех натуральных s не зависят от h.

Стационарная последовательность $\{X_t\}_{-\infty}^{\infty}$ обладает свойством *равномерно сильного перемешивания* [10, с. 391], если

$$\varphi(t) = \sup_{A \in \mathcal{F}_{-\infty}^{0}, B \in \mathcal{F}_{t}^{\infty}} |\mathsf{P}[B|A] - \mathsf{P}[B]| \downarrow 0, \quad t \to \infty,$$

где $\mathcal{F}_a^b$ — σ-алгебра событий, порождённая величинами $X_a, \ldots, X_b$.

В работе [9] доказаны теоремы, описывающие условия асимптотической нормальности при $n \to \infty$ распределения случайной величины

$$\xi_r = \sum_{1 \leqslant j_1 < \ldots < j_r \leqslant n} I\{X_{j_1} = \ldots = X_{j_r}\}$$

— числа r-кратных повторений знаков в отрезке $X_1, \ldots, X_n$ (здесь и далее $I\{A\}$ — индикатор события A). В частности, доказана следующая теорема 1.

Введём случайные величины

$$\zeta_k = \sum_{t=1}^{n} I\{X_t = k\}, \quad \eta_k = \zeta_k - np_k, \quad k = 1, \ldots, N,$$

$$U_n = \frac{1}{(r-1)!} \sum_{k=1}^{N} p_k^{r-1} \eta_k.$$

Рассмотрим центрированную и нормированную случайную величину

$$\xi_r^* = \frac{\xi_r}{n^{r-1}\sqrt{\mathsf{D}U_n}} - \frac{1}{n^{r-1} r! \sqrt{\mathsf{D}U_n}} \sum_{k=1}^{N} \left(\prod_{j=0}^{r-1} (np_k - j) \right).$$

Теорема 1 [9, теорема 3]. Пусть $X_1, \ldots, X_n$ — отрезок простой стационарной цепи Маркова с множеством состояний $\{1, \ldots, N\}$ и положительной матрицей переходных вероятностей. Тогда для любого $0 < \delta < 1/4$

$$\sup_{-\infty < x < \infty} |\mathsf{P}[\xi_r^* \leqslant x] - \Phi(x)| = O\left(n^{-\delta}\right), \quad n \to \infty.$$

Оказывается, что аналогичный результат может быть получен в несколько более общем случае — для стационарных последовательностей, удовлетворяющих условию равномерно сильного перемешивания. Основной результат работы имеет следующий вид:

Теорема 2. Пусть $X_1, \ldots, X_n$ — отрезок стационарной (в узком смысле) случайной последовательности со значениями из множества $\{1, \ldots, N\}$, со стационарным распределением $\mathsf{P}[X_1 = k] = p_k \in (0; 1)$, $k = 1, \ldots, N$, $p_1 + \ldots + p_N = 1$. Пусть среди вероятностей $p_1, \ldots, p_N$ есть различные, а для коэффициента равномерно сильного премешивания φ последовательности $X_1, \ldots, X_n$ при некотором $\alpha > 0$ выполнено условие

$$\varphi(t) \leqslant t^{-6-\alpha}.$$

Тогда для любого $0 < \delta < \dfrac{\alpha}{4(8+\alpha)}$

$$\sup_{-\infty < x < \infty} |\mathsf{P}[\xi_r^* < x] - \Phi(x)| = O\left(n^{-\delta}\right), \quad n \to \infty,$$

где $\Phi(\cdot)$ — функция распределения стандартного нормального закона.

Замечание 1. В условиях теоремы 2 выполнены неравенства $0 < \delta < 1/4$.

2. Доказательство теоремы 2

Ясно, что последовательности

$$I\{X_1 = k\} - p_k, \ \ldots, \ I\{X_n = k\} - p_k$$

при всех $k = 1, \ldots, N$ являются стационарными и обладают свойством равномерно сильного перемешивания. Такими же свойствами обладают последовательности

$$u_1 = \sum_{k=1}^{N} p_k^{r-1} (I\{X_1 = k\} - p_k), \ \ldots, \ u_n = \sum_{k=1}^{N} p_k^{r-1} (I\{X_n = k\} - p_k).$$

Нетрудно проверить, что

$$U_n = \frac{1}{(r-1)!} (u_1 + u_2 + \ldots + u_n).$$

Замечание 2. В равновероятном случае, когда $p_1 = \ldots = p_N = 1/N$, величины U_n, $u_1, u_2, \ldots, u_n$ с вероятностью единица равны нулю. Условия теоремы 2 исключают эту ситуацию.

Нам понадобится ряд вспомогательных утверждений.

Лемма 1 [9, теорема 2]. Пусть выполнены условия теоремы 2 и R_n — натуральное число, удовлетворяющее условию $R_n < n$. Тогда

$$D_n = \sup_{-\infty<x<\infty} \left| \mathsf{P}\left[\frac{U_n}{\sqrt{\mathsf{D}U_n}} < x\right] - \Phi(x) \right| \leqslant \\ \leqslant \sqrt[4]{\frac{2}{\pi}} \left(4N^3 \frac{nR_n^2}{(\sqrt{\mathsf{D}U_n})^3} + \frac{6}{\pi(r-1)!} \frac{n^2\varphi(R_n)}{\sqrt{\mathsf{D}U_n}} + \frac{2N^2}{\pi} \frac{n^2\varphi(R_n)}{\mathsf{D}U_n} \right)^{1/2}. \tag{1}$$

Лемма 2. Пусть выполнены условия теоремы 2. Тогда для любого $0 < \delta < \dfrac{\alpha}{4(8+\alpha)}$

$$D_n = \sup_{-\infty<x<\infty} \left| \mathsf{P}\left[\frac{U_n}{\sqrt{\mathsf{D}U_n}} < x\right] - \Phi(x) \right| = O\left(\frac{1}{n^\delta}\right), \quad n \to \infty.$$

Доказательство. Обозначим

$$A_n = 4N^3 \frac{nR_n^2}{(\sqrt{\mathsf{D}U_n})^3}; \tag{2}$$

$$B_n = \frac{6}{\pi(r-1)!} \frac{n^2\varphi(R_n)}{\sqrt{\mathsf{D}U_n}}; \tag{3}$$

$$C_n = \frac{2N^2}{\pi} \frac{n^2\varphi(R_n)}{\mathsf{D}U_n}. \tag{4}$$

Тогда неравенство (1) можно переписать в виде

$$\sqrt{\frac{\pi}{2}} D_n^2 \leqslant A_n + B_n + C_n. \tag{5}$$

Заметим, что величина U_n является суммой элементов стационарной последовательности. Применим к ней утверждение [10, теорема 18.2.3, с. 413]: если вещественная стационарная последовательность $X_1, \ldots, X_n$ удовлетворяет условию равномерно сильного перемешивания и $\lim\limits_{n\to\infty} \mathsf{D}S_n = \infty$, то

$$\mathsf{D}S_n = nh(n), \tag{6}$$

где $h(n)$ — медленно меняющаяся функция.

Напомним, что функция $h = h(x)$ вещественного аргумента x называется медленно меняющейся, если она удовлетворяет условию

$$\lim_{x\to\infty} \frac{h(tx)}{h(x)} = 1 \quad \text{для всех } t > 0.$$

Для примера отметим, что медленно меняющимися являются функции $(\ln x)^\alpha$ при $\alpha \in (-\infty, \infty)$.

Тогда, возвращаясь к нашей задаче, согласно (6), имеем

$$\mathsf{D}U_n = nh(n), \tag{7}$$

где $h(n)$ — медленно меняющаяся при $n \to \infty$ функция.

Из (2)–(4) и (7) следует, что

$$A_n = 4N^3 \frac{nR_n^2}{(\sqrt{\mathsf{D}U_n})^3} = \frac{R_n^2}{\sqrt{n}}\, h_1(n); \tag{8}$$

$$B_n = \frac{6}{\pi(r-1)!}\, \frac{n^2\varphi(R_n)}{\sqrt{\mathsf{D}U_n}} = \frac{n^{3/2}}{R_n^{6+\alpha}}\, h_2(n); \tag{9}$$

$$C_n = \frac{2N^2}{\pi}\, \frac{n^2\varphi(R_n)}{\mathsf{D}U_n} = \frac{n}{R_n^{6+\alpha}}\, h_3(n). \tag{10}$$

Здесь $h_1(n), h_2(n), h_3(n)$ — медленно меняющиеся при $n \to \infty$ функции.

Подставив оценки (8)–(10) в (5), получим

$$\sqrt{\frac{\pi}{2}} D_n^2 \leqslant \left(\frac{R_n^2}{\sqrt{n}} + \frac{n^{3/2}}{R_n^{6+\alpha}} \right) \tilde{h}(n), \tag{11}$$

где $\tilde{h}(n)$ — некоторая медленно меняющаяся при $n \to \infty$ функция.

Положив

$$R_n = \left[n^{2/(8+\alpha)} \right],$$

получим, что слагаемые в правой части (11) равны по порядку с точностью до медленно меняющихся функций и

$$D_n = O\left(\frac{R_n}{n^{1/4}} \right) = O\left(n^{-\alpha/(4(8+\alpha))} \right), \quad n \to \infty. \tag{12}$$

Из (12) следует утверждение леммы 2. ■

Лемма 3. Пусть выполнены условия теоремы 2 и для распределения случайной величины U_n при некоторых $c > 0$ и $0 < \delta < \alpha/(4(8+\alpha))$ справедливо неравенство

$$\sup_{-\infty<x<\infty} \left| \mathsf{P}\left[\frac{U(n)}{\sqrt{\mathsf{D}U(n)}} < x \right] - \Phi(x) \right| \leqslant \frac{c}{n^\delta}. \tag{13}$$

Тогда найдётся такое $c_1 < \infty$, что

$$\sup_{-\infty<x<\infty} |\mathsf{P}[\xi_r^* \leqslant x] - \Phi(x)| \leqslant \frac{c_1}{n^\delta}. \tag{14}$$

Доказательство. Лемма 3 доказывается аналогично лемме 3 в [9]. Пусть

$$V = \xi_r^* - \frac{U_n}{\sqrt{\mathsf{D}U_n}},$$

$\{v_k : k \geqslant 1\}$ — множество значений величины V. Тогда для $x \in \mathbb{R}$

$$\mathsf{P}[\xi_r^* < x] = \mathsf{P}\left[\frac{U_n}{\sqrt{\mathsf{D}U_n}} + V < x \right] = \mathsf{P}\left[\frac{U_n}{\sqrt{\mathsf{D}U_n}} + \sum_{k\geqslant 1} v_k I\{V = v_k\} < x \right] =$$

$$= \sum_{k\geqslant 1} \mathsf{P}\left[\frac{U_n}{\sqrt{\mathsf{D}U_n}} < x - v_k \right] \mathsf{P}[V = v_k].$$

Воспользуемся оценкой (13). Получаем

$$\begin{gathered}\mathsf{P}[\xi_r^* < x] \leqslant \sum_{k\geqslant 1} \mathsf{P}[V = v_k] \left(\Phi(x - v_k) + \frac{c}{n^\delta}\right) = \\ = \left(\sum_{k\geqslant 1} \mathsf{P}[V = v_k] \left(\Phi(x - v_k) + \frac{c}{n^\delta}\right) - \Phi(x)\right) + \Phi(x).\end{gathered} \tag{15}$$

Так как $\sum\limits_{k\geqslant 1} \mathsf{P}\{V = v_k\} = 1$, то из (15) следует

$$\begin{gathered}\mathsf{P}[\xi_r^* < x] \leqslant \Phi(x) + \frac{c}{n^\delta} + \sum_{k\geqslant 1} \mathsf{P}[V = v_k] \left(\Phi(x - v_k) - \Phi(x)\right) \leqslant \\ \leqslant \Phi(x) + \frac{c}{n^\delta} + \frac{1}{\sqrt{2\pi}} \sum_{k\geqslant 1} |v_k| \, \mathsf{P}[V = v_k] = \Phi(x) + \frac{c}{n^\delta} + \frac{1}{\sqrt{2\pi}} \mathsf{E}|V|.\end{gathered} \tag{16}$$

Согласно [9, оценка (15)], выполнено соотношение $\mathsf{E}|V| = O\left(h(n)n^{-1/2}\right)$, где $h(n)$ — некоторая медленно меняющаяся функция. Поэтому из (16) вытекает, что найдётся такое $c' < \infty$, при котором

$$\mathsf{P}[\xi_r^* < x] \leqslant \Phi(x) + \frac{c'}{n^\delta}. \tag{17}$$

Аналогично доказывается, что

$$\mathsf{P}[\xi_r^* < x] \geqslant \Phi(x) - \frac{c''}{n^\delta}. \tag{18}$$

Из (17) и (18) следует (14) с $c_1 = \max\{c', c''\}$. ■

Этим доказательство теоремы 2 завершено.

Заключение

Получена оценка скорости сходимости распределения числа r-кратных повторений знаков в стационарной случайной последовательности, удовлетворяющей условию равномерно сильного перемешивания с коэффициентом $\varphi(t) \leqslant t^{-6-\alpha}$ при некотором $\alpha > 0$, к нормальному распределению. Установлено, что в этом случае расстояние в равномерной метрике между функцией распределения центрированного и нормированного специальным образом числа повторений и сопровождающим нормальным распределением с увеличением длины последовательности n убывает со скоростью $O(n^{-\delta})$ для любого $\delta \in (0, \alpha(32 + 4\alpha)^{-1}))$.

ЛИТЕРАТУРА

1. *Иванов М. А., Чугунков И. В.* Теория, применение и оценка качества генераторов псевдослучайных последовательностей. М.: КУДИЦ-ОБРАЗ, 2003. 240 с.
2. *Rukhin A., Soto J., Nechvatal J., et al.* A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST, Apr. 2010. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf`.
3. *Михайлов В. Г.* Предельная теорема пуассоновского типа для числа пар почти полностью совпавших цепочек // Теория вероятностей и ее применения. 2008. Т. 53. Вып. 1. С. 59–71.
4. *Михайлов В. Г., Шойтов А. М.* О числах множеств эквивалентных цепочек в последовательности независимых случайных величин // Математические вопросы криптографии. 2013. Т. 4. Вып. 1. С. 77–86.
5. *Шойтов А. М.* Нормальное приближение в задаче об эквивалентных цепочках // Труды по дискретной математике. 2007. Т. 10. С. 326–349.

6. *Михайлов В. Г.* Оценки точности пуассоновской аппроксимации для распределения числа серии повторений длинных цепочек в цепи Маркова // Дискретная математика. 2015. Т. 27. Вып. 4. С. 67–78.
7. *Михайлов В. Г., Шойтов А. М.* О длинных повторениях цепочек в цепи Маркова // Дискретная математика. 2014. Т. 26. Вып. 3. С. 79–89.
8. *Михайлов В. Г., Шойтов А. М.* Многократные повторения длинных цепочек в цепи Маркова // Математические вопросы криптографии. 2015. Т. 6. Вып. 3. С. 117–134.
9. *Михайлов В. Г., Меженная Н. М., Волгин А. В.* Об условиях асимптотической нормальности числа повторений в стационарной случайной последовательности // Дискретная математика. 2021. Т. 33. Вып. 3. С. 64–78.
10. *Ибрагимов И. А., Линник Ю. В.* Независимые и стационарно связанные случайные величины М.: Наука, 1965. 816 с.

REFERENCES

1. *Ivanov M. A. and Chugunkov I. V.* Teoriya, primenenie i otsenka kachestva generatorov psevdosluchaynykh posledovatel'nostey [Theory, Application and Evaluation of the Quality of Pseudo-Random Sequence Generators]. Moscow, KUDITs-OBRAZ, 2003. 240 p. (in Russian)
2. *Rukhin A., Soto J., Nechvatal J., et al.* A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST, Apr. 2010. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf`.
3. *Mikhailov V. G.* A Poisson-type limit theorem for the number of pairs of matching sequences. Theory of Probability and its Appl., 2009, vol. 53, no. 1, pp. 106–116.
4. *Mikhaylov V. G. and Shoytov A. M.* O chislakh mnozhestv ekvivalentnykh tsepochek v posledovatel'nosti nezavisimykh sluchaynykh velichin [On the numbers of equivalent tuples sets in a sequence of independent random variables]. Matematicheskie Voprosy Kriptografii, 2013, vol. 4, iss. 1, pp. 77–86. (in Russian)
5. *Shoytov A. M.* Normal'noe priblizhenie v zadache ob ekvivalentnykh tsepochkakh [Normal approximation in a problem on equivalent tuples]. Trudy po Diskretnoy Matematike, 2007, vol. 10, pp. 326–349. (in Russian)
6. *Mikhailov V. G.* Estimates of accuracy of the Poisson approximation for the distribution of number of runs of long string repetitions in a Markov chain. Discr. Math. Appl., 2016, vol. 26, no. 2, pp. 105–113.
7. *Mikhailov V. G. and Shoitov A. M.* On repetitions of long tuples in a Markov chain. Discr. Math. Appl., 2015, vol. 25, no. 5, pp. 295–303.
8. *Mikhaylov V. G. and Shoytov A. M.* Mnogokratnye povtoreniya dlinnykh tsepochek v tsepi Markova [On multiple repetitions of long tuples in a Markov chain]. Matematicheskie Voprosy Kriptografii, 2015, vol. 6, iss. 3, pp. 117–134. (in Russian)
9. *Mikhaylov V. G., Mezhennaya N. M., and Volgin A. V.* Ob usloviyakh asimptoticheskoy normal'nosti chisla povtoreniy v statsionarnoy sluchaynoy posledovatel'nosti [Conditions for asymptotic normality of the number of repetitions in a discrete stationary random sequence]. Diskretnaya Matematika, 2021, vol. 33, iss. 3, pp. 64–78.
10. *Ibragimov I. A. and Linnik. Yu. V.* Independent and Stationary Sequences of Random Variables. Wolters-Noordhoff, Groningen, Netherlands, 1971.

УДК 511.23 DOI 10.17223/20710410/58/3

ON IDEAL CLASS GROUP COMPUTATION OF IMAGINARY MULTIQUADRATIC FIELDS[1]

S. A. Novoselov

Immanuel Kant Baltic Federal University, Kaliningrad, Russia

E-mail: snovoselov@kantiana.ru

In the paper, we extend Biasse — van Vredendaal (OBS, 2019, vol. 2) implementation and experiments of the class group computation from real to imaginary multiquadratic fields. The implementation is optimized by introducing an explicit prime ideal lift operation and by using LLL reduction instead of HNF computation. We provide examples of class group computation of the imaginary multiquadratic fields of degree 64 and 128, that has been previously unreachable.

Keywords: *multiquadratic field, ideal class group.*

О ВЫЧИСЛЕНИИ ГРУППЫ КЛАССОВ ИДЕАЛОВ МНИМЫХ МУЛЬТИКВАДРАТИЧНЫХ ПОЛЕЙ

С. А. Новоселов

БФУ им. И. Канта, г. Калининград, Россия

Расширены эксперименты Биассе — ван Вредендал (OBS, 2019, vol. 2) по вычислению группы классов идеалов с действительных мультиквадратичных полей на мнимые мультиквадратичные поля. Представлена адаптированная и оптимизированная для работы с мнимыми полями реализация алгоритма Биассе — ван Вредендал. Оптимизации включают в себя введение и использование явных формул для подъёма простых идеалов и замену вычислений эрмитовой нормальной формы на LLL-редукцию. Представлены примеры вычисления группы классов для мнимых мультквадратичных полей степени 64 и 128, недостижимые ранее.

Ключевые слова: *мультквадратичное числовое поле, группа классов идеалов.*

1. Introduction

A multiquadratic field of degree 2^n is defined as

$$K = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n}),$$

where $d_1, \ldots, d_n$ are square-free integers. The field K is called real if all $d_1, \ldots, d_n$ are positive, otherwise it is called imaginary. The class group Cl_K of the multiquadratic field is a factor group of fractional ideals of K modulo principal ideals. The class group computation of a number field is a classical hard problem in algebraic number theory. It is an essential tool in cryptanalysis of the cryptosystems based on arithmetic operations in number fields [1–3] and ideal lattices [4, 5]. Security of the first ones relies on hardness of the short principal

[1]The research was funded by the Russian Science Foundation (project No. 22-41-04411, `https://rscf.ru/en/project/22-41-04411/`).

ideal problem (SPIP) and the discrete logarithm problem. For real multiquadratic fields, the solution to SPIP is provided by the paper [6]. The discrete logarithm problem in number fields can be solved by factoring target ideal into product of prime ideals, adding these obtained primes to the factor base, and running the class group computation algorithm. Cryptosystems based on ideal lattices rely on the hardness of approximate shortest vector problem. The class group computation is also used in [7, 8] as part of the algorithm for solving this problem.

The common way to compute the ideal class group of a number field is to use algorithms [9–11]. These algorithms are subexponential in $\log \Delta_K$, where Δ_K is the discriminant of the field. Biasse and van Vredendaal proposed [12] an algorithm that reduces the problem of the ideal class group computation of a multiquadratic field K to the quadratic subfields of K. Therefore, the time complexity was reduced to subexponential time in $\log D$, where $D = d_1 \cdot \ldots \cdot d_n$ is the largest discriminant of the quadratic subfield. In practice, this allowed Biasse and van Vredendaal to compute class groups of real multiquadratic fields of degrees up to 128. However, provided implementation does not support imaginary multiquadratic fields. In addition, the latter case requires optimizations since the class groups become much larger.

Our contributions:

1) We extend Biasse — van Vredendaal [12] implementation for the class group computation of real multiquadratic fields to imaginary multiquadratic fields[2].
2) For both imaginary and real multiquadratic fields, we provide an optimized algorithm for computing the splitting trees of prime ideals over subfields of the field K. These trees are used in the algorithm to lift solution from subfields to the base field.
3) To reduce memory consumption when computing large class groups, we have replaced Hermite normal form computations with LLL-reduction [13].
4) Using results above, we computed the ideal class group for previously unreachable imaginary fields of degrees 64 and 128. For degree 256, partial information is computed — the result is correct up to factor of size $\leqslant 32208$.

Organization of the paper. In section 2 we give necessary notations. Section 3.1 briefly describes a general approach from [9, 10] for the class group computation and its improvement by Biasse and van Vredendaal [12]. In 3.3 we describe the choice of norm bounds for prime ideals in the factor base. Section 3.4 describes the verification procedure for the result of the class group computation. Sections 4 and 5 contain main results — improvements for the Biasse — van Vredendaal implementation and experiments on the class group computation.

2. Notation

We use the following notations:

— $K = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ is a n-quadratic field where all d_i are pairwise coprime and square-free, $\mathcal{O}_K$ is its ring of integers;
— $\mathrm{Cl}_K = \mathrm{I}_K \,/\, \mathrm{Princ}_K$ — the class group of K, i.e., the factor group of fractional ideals I_K of K modulo principal ideals of $\mathcal{O}_K$;
— h is the order of Cl_K;
— $G_K = \mathrm{Gal}(K/\mathbb{Q})$ is the absolute Galois group of K;
— K_σ is a subfield, that is invariant under the action of an automorphism $\sigma \in G_K$;
— Δ_K is the discriminant of K;

[2]Available here: `https://github.com/novoselov-sa/multiclass-im`.

— $N(I)$, $N(a)$ is the absolute norm of an ideal I or $a \in K$, respectively;
— $N_{L/K}(\alpha)$ is the relative norm of $\alpha \in L$;
— $\mathcal{C}_m$ is a cyclic group of order m.

3. Preliminaries

3.1. General algorithm for class group computation

Assume that we have a factor base $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_d\}$ — a set of all prime ideals of $\mathcal{O}_K$ that generate the class group Cl_K. To build this factor base, we can take all prime ideals of norm $\leqslant 12 \log^2 \Delta_K$, see [14]. In practice, this general bound is very pessimistic and a better heuristic bound can be computed instead using Grenié — Molteni algorithm [15].

The computation of the ideal class group of a number field K can be done by collecting enough relations $(\alpha, e) \in K \times \mathbb{Z}^d$ of the form

$$\alpha \mathcal{O}_K = \prod_{i=1}^{d} \mathfrak{p}_i^{e_i}.$$

One relation is computed by taking a random element $\alpha \in K$ that splits over the factor base S. After collecting enough relations, we form a matrix of relations A, the rows of which are vectors e. The Smith Normal Form $A = UBV$ of this matrix gives us the group structure and the generators of the class group. In particular, we have $B = \mathrm{diag}(b_1, \ldots, b_k)$ and the ideal class group is the following product of cyclic groups:

$$\mathrm{Cl}_K \simeq \mathcal{C}_{b_1} \times \ldots \times \mathcal{C}_{b_k} \simeq \langle \mathfrak{g}_1 \rangle \times \ldots \times \langle \mathfrak{g}_k \rangle,$$

where $\mathfrak{g}_i = \prod_{j=1}^{d} \mathfrak{p}_j^{v'_{i,j}}$ for $V^{-1} = (v'_{i,j})$. In addition, we have $\mathfrak{p}_i = \prod_{j=1}^{k} \mathfrak{g}_j^{v_{i,j}}$. It is well known that the whole process of computing the ideal class group takes subexponential time [9] in $\log \Delta_K$.

3.2. Algorithm of Biasse and van Vredendaal

For an arbitrary number field, the complexity of the class group computation is subexponential in $\log \Delta_K$. However, as it was shown by Biasse and van Vredendaal [12], for multiquadratic fields the problem of the class group computation can be efficiently reduced to finding relations in certain quadratic subfields of K, which is a much simpler task, and then lifting them to K. It results in an algorithm of complexity subexponential in $\log D$, where $D = d_1 \cdot \ldots \cdot d_n$ is the largest discriminant of quadratic subfields, which is always smaller than Δ_K.

3.3. Choice of norm bound

Choosing the correct bound for the norms of the prime ideals that constitute the factor base is also a non-trivial task. Taking all primes satisfying Bach's bound [14] or the bound computed using the Grenié — Molteni script [15] appears to be impractical due to high memory consumption (the factor base turns out to be too large for both real and imaginary cases). Instead, in our experiments, we used the following idea. Since Biasse — van Vredendaal algorithm lifts the relations from quadratic subfields of K back to the base field K, choosing primes with norms greater than the corresponding bounds for these quadratic subfields does not give us new relations to be lifted. So we first compute the bounds for all quadratic subfields with Grenié — Molteni script [15]. We take the maximum of these bounds plus a small constant suggested by the experiments (these are of orders 200 for $n \leqslant 5$).

3.4. Verification

To verify class group computation, we checked the obtained class number following the approach from [10, Step D]. Assuming Generalized Riemann Hypothesis (GRH), one can verify the result of class group computation for the field K in polynomial time (in $\log \Delta_K$ and $\deg K$) by computing the product hR_K with enough precision, where R_K is a regulator [16, Def. 4.9.8] of K. This product can be computed using the following class number formula [16, Th. 4.9.12]:

$$\lim_{s\to 1}(s-1)\zeta_K(s) = 2^{r_1}(2\pi)^{r_2}\frac{hR_K}{w_K\sqrt{|\Delta_K|}}.$$

Here $\zeta_K(s) = \prod_{\mathfrak{p}} \frac{1}{1-(N(\mathfrak{p}))^{-s}}$ is the Dedekind zeta function of K, w_K is a number of roots of unity, r_1 is a number of real embeddings of K, and r_2 is such that $2r_2$ is a number of complex embeddings of K. The left side of this formula (Euler product) can be computed in polynomial time (assuming GRH) following [17, 18]. The other parts of the class number formula (with exception of h and R_K) can be efficiently computed for multiquadratic fields. This gives us the (approximated) product εhR_K. The term ε represents error in the computation of Euler product.

To verify the class group computation, we compare the computed product εhR_K with the tentative product h^*R^*, where h^* is the tentative class number — result of our class number computation, and R^* is the approximation of regulator computed using [6]. If $|\varepsilon| < \sqrt{2}$, then it remains to check that $h^*R_K^* < \sqrt{2}\varepsilon hR_K$, since $h^*R_K^*$ is an integral multiple of hR. In our experiments we computed Euler products using the method zeta_log_residue from Hecke package [19] for Julia [20]. This method implements the approximation of Euler products from [18].

4. Optimizations and improvements for the Biasse — van Vredendaal algorithm

4.1. The case of imaginary multiquadratic fields

The algorithm of Biasse — van Vredendaal and its implementation [12] is given for real multiquadratic fields. However, as claimed in the paper [12, p. 104], the algorithm works for any fields, where we can choose two different automorphisms of order 2. Indeed, after cumbersome, but rather technical fixes in the Biasse — van Vredendaal implementation, we were able to run the class group computation algorithm for imaginary multiquadratic fields. To be more precise, in our implementation we take into account a different rank of the unit group as in imaginary multiquadratic field it is half of a real multiquadratic field of the same degree. Additionally, we augment the implementation with a procedure that generates trees describing the splitting of a prime ideal over the subfields — a missing piece of the implementation from [12]. This is the subject of the next section. After running algorithm for imaginary fields, we were unable to compute the class groups for high degree fields due to high memory consumption. This was caused by HNF computations. So we replaced HNF computation with LLL [13] reduction.

4.2. Generation of splitting trees

The main drawback of the Biasse — van Vredendaal implementation [12] is the lack of code that computes trees, describing splitting of prime ideals over subfields. These trees contain for each prime ideal $\mathfrak{P}$ of a subfield K_γ of the field K the exponent vector $\mathbf{x} = (x_1, \ldots, x_d)$ such that $\mathfrak{P}\mathcal{O}_K = \prod_{i=1}^{d} \mathfrak{p}_i^{x_i}$. This information allows us efficiently lift prime

ideals and thus prime ideal products from K_γ to K. The prime ideals in these trees are also used as a factor base for computations in the subfields. Our implementation produces the splitting picture for prime ideals using a certain representation of these ideals, as well as the operation for lifting ideal from the subfields to the base field. We give the details in the next section.

4.3. Representation of prime ideals

Let $\theta = \sqrt{d_1} + \ldots + \sqrt{d_n}$ and $f = \prod\limits_{\gamma \in G_K} (x - \gamma(\theta))$. Let $K = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n}) = \mathbb{Q}(x)/f\mathbb{Q}(x)$. If we consider only primes p such that $p \nmid [\mathcal{O}_K : \mathbb{Z}[\theta]]$, then every prime ideal $\mathfrak{p} \mid p\mathcal{O}_K$ may be written in one of the two following forms:

1) $(p, \theta + \overline{\gamma(\theta)})$ for some $\gamma \in G_K$ if all $d_1, \ldots, d_n$ are quadratic residues modulo p;
2) $(p, (\theta - \overline{\gamma(\theta')})^2 - \overline{\gamma(\theta'')^2})$ for $\gamma \in G_K$, $\theta' = \sum\limits_{\sqrt{d_i} \in \mathbb{F}_p} \sqrt{d_i}$, and $\theta'' = \sum\limits_{\sqrt{d_i} \notin \mathbb{F}_p} \sqrt{d_i}$.

Here $\overline{z}$ denotes the reduction of z modulo p. This representation of primes is unique and there is one-to-one correspondence between primes and automorphisms G_K for the case 1 and for the automorphisms in $G_K/\pm 1$ for the case 2. The representation follows from the standard prime decomposition theorem [16, Th. 4.8.13] and from the fact that the multiquadratic fields are Galois extensions of $\mathbb{Q}$.

4.4. Fast ideal above operation

For $K = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$, the class group computation algorithm of Biasse — van Vredendaal [12] uses only three subfields $K_\sigma = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_{n-1}})$, $K_\tau = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_{n-2}}, \sqrt{d_n})$, and $K_{\sigma\tau} = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_{n-2}}, \sqrt{d_{n-1}}\sqrt{d_n})$ fixed by the automorphisms:

1) $\sigma : \sqrt{d_n} \mapsto -\sqrt{d_n}$;
2) $\tau : \sqrt{d_{n-1}} \mapsto -\sqrt{d_{n-1}}$;
3) $\sigma\tau : \sqrt{d_n} + \sqrt{d_{n-1}} \mapsto -\sqrt{d_n} - \sqrt{d_{n-1}}$.

For fast generation of the trees describing splitting of prime ideals over the subfields, we used an explicit ideal above (Lift) operation for prime ideals presented in Table 1. It is defined for prime ideals that are given in the form as described in Section 4.3. For the field $K_{\tau\sigma}$ we omitted the cases $\sqrt{d}_n \in \mathbb{F}_p, \sqrt{d}_{n-1} \notin \mathbb{F}_p$ or $\sqrt{d}_n \notin \mathbb{F}_p, \sqrt{d}_{n-1} \in \mathbb{F}_p$. They can be given in analogous way and differ from the presented case by signs only. Table 1 was derived using the fact that for an unramified ideal $\mathfrak{p}$ of $\mathcal{O}_{K_\gamma}$ and the ideal $\mathfrak{P} = (p, \beta)$ of $\mathcal{O}_K$ such that $\mathfrak{P} \mid \mathfrak{p}\mathcal{O}_K$ we have [21, Alg. 2.5.3] that $\mathfrak{p} = (p, \mathrm{N}_{K/K_\gamma}(\beta))$. Notations in the Table 1: $\mathfrak{p} \mid (p)$, $\mathfrak{p} = (p, \alpha)$ is a prime ideal of $\mathcal{O}_{K_\gamma}$ for some prime number p; $\mathfrak{p}\mathcal{O}_K = \mathfrak{P}_1\mathfrak{P}_2$, in the case $\mathfrak{P}_2 = (1)$ we omit $\mathfrak{P}_2$; $\mathfrak{P}_1 = (p, \beta_1)$, $\mathfrak{P}_2 = (p, \beta_2)$.

The described Lift operation allows us to do the fast generation of the splitting trees for primes $\mathfrak{p} \mid (p)$, where $p \nmid [\mathcal{O}_K : \mathbb{Z}[\theta]]$ and $p \nmid [\mathcal{O}_{K_\gamma} : \mathbb{Z}[\theta_\gamma]]$ for each subfield K_γ of K.

Table 1

Explicit lift of prime ideals from subfields K_γ of K

Subfield	α	Conditions	β_1, β_2
K_σ	$\theta_\sigma + \overline{\mu(\theta_\sigma)}$	$\sqrt{d}_n \in \mathbb{F}_p$	$\theta + \overline{\mu(\theta_\sigma)} + \overline{\sqrt{d_n}}$, $\theta + \overline{\mu(\theta_\sigma)} - \overline{\sqrt{d_n}}$
K_σ	$\theta_\sigma + \overline{\mu(\theta_\sigma)}$	$\sqrt{d}_n \notin \mathbb{F}_p$	$(\theta + \mu(\theta_\sigma))^2 - \overline{d_n}$
K_σ	$(\theta_\sigma - \overline{\mu(\theta'_\sigma)})^2 - \overline{\mu(\theta''_\sigma)^2}$	$\sqrt{d}_n \in \mathbb{F}_p$	$(\theta_\sigma - \overline{\mu(\theta'_\sigma) + \sqrt{d_n}})^2 - \overline{\mu(\theta''_\sigma)^2}$, $(\theta_\sigma - \overline{\mu(\theta'_\sigma) - \sqrt{d_n}})^2 - \overline{\mu(\theta''_\sigma)^2}$
K_σ	$(\theta_\sigma - \overline{\mu(\theta'_\sigma)})^2 - \overline{\mu(\theta''_\sigma)^2}$	$\sqrt{d}_n \notin \mathbb{F}_p$	$(\theta_\sigma - \overline{\mu(\theta'_\sigma)})^2 - \overline{(\mu(\theta''_\sigma) + \sqrt{d_n})^2}$, $(\theta_\sigma - \overline{\mu(\theta'_\sigma)})^2 - \overline{(\mu(\theta''_\sigma) - \sqrt{d_n})^2}$
K_τ	$\theta_\tau + \overline{\mu(\theta_\tau)}$	$\sqrt{d_{n-1}} \in \mathbb{F}_p$	$\theta + \overline{\mu(\theta_\tau)} + \overline{\sqrt{d_{n-1}}}$, $\theta + \overline{\mu(\theta_\tau)} - \overline{\sqrt{d_{n-1}}}$
K_τ	$\theta_\tau + \overline{\mu(\theta_\tau)}$	$\sqrt{d_{n-1}} \notin \mathbb{F}_p$	$(\theta + \overline{\mu(\theta_\tau)})^2 - \overline{d_{n-1}}$
K_τ	$(\theta_\tau - \overline{\mu(\theta'_\tau)})^2 - \overline{\mu(\theta''_\tau)^2}$	$\sqrt{d_{n-1}} \in \mathbb{F}_p$	$(\theta_\sigma - \overline{\mu(\theta'_\tau) + \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_\tau)^2}$, $(\theta_\tau - \overline{\mu(\theta'_\tau) - \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_\tau)^2}$
K_τ	$(\theta_\tau - \overline{\mu(\theta'_\tau)})^2 - \overline{\mu(\theta''_\tau)^2}$	$\sqrt{d_{n-1}} \notin \mathbb{F}_p$	$(\theta_\tau - \overline{\mu(\theta'_\tau)})^2 - \overline{(\mu(\theta''_\tau) + \sqrt{d_{n-1}})^2}$, $(\theta_\tau - \overline{\mu(\theta'_\tau)})^2 - \overline{(\mu(\theta''_\tau) - \sqrt{d_{n-1}})^2}$
$K_{\sigma\tau}$	$\theta_{\sigma\tau} + \overline{\mu(\theta_{\sigma\tau})}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \in \mathbb{F}_p$, $\mu(\sqrt{d_{n-1}d_n}) = \sqrt{d_{n-1}d_n}$	$\theta + \overline{\mu(\theta_{\sigma\tau})} - \overline{\mu(\sqrt{d_{n-1}d_n})} + \overline{\sqrt{d_{n-1}}} - \overline{\sqrt{d_n}}$ $\theta + \overline{\mu(\theta_{\sigma\tau})} - \overline{\mu(\sqrt{d_{n-1}d_n})} - \overline{\sqrt{d_{n-1}}} + \overline{\sqrt{d_n}}$
$K_{\sigma\tau}$	$\theta_{\sigma\tau} + \overline{\mu(\theta_{\sigma\tau})}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \in \mathbb{F}_p$, $\mu(\sqrt{d_{n-1}d_n}) = -\sqrt{d_{n-1}d_n}$	$\theta + \overline{\mu(\theta_{\sigma\tau})} - \overline{\mu(\sqrt{d_{n-1}d_n})} + \overline{\sqrt{d_{n-1}}} + \overline{\sqrt{d_n}}$ $\theta + \overline{\mu(\theta_{\sigma\tau})} - \overline{\mu(\sqrt{d_{n-1}d_n})} - \overline{\sqrt{d_{n-1}}} - \overline{\sqrt{d_n}}$
$K_{\sigma\tau}$	$\theta_{\sigma\tau} + \overline{\mu(\theta_{\sigma\tau})}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \notin \mathbb{F}_p$, $\mu(\sqrt{d_{n-1}d_n}) = \sqrt{d_{n-1}d_n}$	$(\theta + \overline{\mu(\theta_{\sigma\tau}) - \sqrt{d_{n-1}d_n}})^2 - \overline{(d_{n-1} + d_n)^2}$
$K_{\sigma\tau}$	$\theta_{\sigma\tau} + \overline{\mu(\theta_{\sigma\tau})}$	$\sqrt{d}_{n-1}, \sqrt{d}_n \notin \mathbb{F}_p$, $\mu(\sqrt{d_{n-1}d_n}) = -\sqrt{d_{n-1}d_n}$	$(\theta + \overline{\mu(\theta_{\sigma\tau}) + \sqrt{d_{n-1}d_n}})^2 - \overline{(d_{n-1} + d_n)^2}$
$K_{\sigma\tau}$	$(\theta_{\sigma\tau} - \overline{\mu(\theta'_{\sigma\tau})})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \in \mathbb{F}_p$ $\mu(\sqrt{d_{n-1}d_n}) = \sqrt{d_{n-1}d_n}$	$(\theta - \overline{\mu(\theta'_{\sigma\tau}) - \sqrt{d_{n-1}d_n} + \sqrt{d_n} + \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$ $(\theta - \overline{\mu(\theta'_{\sigma\tau}) - \sqrt{d_{n-1}d_n} - \sqrt{d_n} - \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$
$K_{\sigma\tau}$	$(\theta_{\sigma\tau} - \overline{\mu(\theta'_{\sigma\tau})})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \in \mathbb{F}_p$ $\mu(\sqrt{d_{n-1}d_n}) = -\sqrt{d_{n-1}d_n}$	$(\theta - \overline{\mu(\theta'_{\sigma\tau}) + \sqrt{d_{n-1}d_n} + \sqrt{d_n} - \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$ $(\theta - \overline{\mu(\theta'_{\sigma\tau}) + \sqrt{d_{n-1}d_n} - \sqrt{d_n} + \sqrt{d_{n-1}}})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$
$K_{\sigma\tau}$	$(\theta_{\sigma\tau} - \overline{\mu(\theta_{\sigma\tau})})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \notin \mathbb{F}_p$ $\mu(\sqrt{d_{n-1}d_n}) = \sqrt{d_{n-1}d_n}$	$(\theta - \overline{\mu(\theta'_{\sigma\tau}) - \sqrt{d_{n-1}d_n}})^2 - \overline{(\mu(\theta''_{\sigma\tau}) + \sqrt{d_n} + \sqrt{d_{n-1}})^2}$ $(\theta - \overline{\mu(\theta'_{\sigma\tau}) - \sqrt{d_{n-1}d_n}})^2 - \overline{(\mu(\theta''_{\sigma\tau}) - \sqrt{d_n} - \sqrt{d_{n-1}})^2}$
$K_{\sigma\tau}$	$(\theta_{\sigma\tau} - \overline{\mu(\theta_{\sigma\tau})})^2 - \overline{\mu(\theta''_{\sigma\tau})^2}$	$\sqrt{d_n}, \sqrt{d_{n-1}} \notin \mathbb{F}_p$ $\mu(\sqrt{d_{n-1}d_n}) = -\sqrt{d_{n-1}d_n}$	$(\theta - \overline{\mu(\theta'_{\sigma\tau}) + \sqrt{d_{n-1}d_n}})^2 - \overline{(\mu(\theta''_{\sigma\tau}) + \sqrt{d_n} - \sqrt{d_{n-1}})^2}$ $(\theta - \overline{\mu(\theta'_{\sigma\tau}) + \sqrt{d_{n-1}d_n}})^2 - \overline{(\mu(\theta''_{\sigma\tau}) - \sqrt{d_n} + \sqrt{d_{n-1}})^2}$

5. Experiments

For experiments, we used implementation of [12], which we adapted and optimized for imaginary multiquadratic fields as described above. In our experiments we used the fields of the form

$$K = \mathbb{Q}(\sqrt{-3}, \sqrt{-7}, \sqrt{-11}, \sqrt{-19}, \sqrt{-23}, \ldots),$$

where $d_1, \ldots, d_n$ are primes such that $d_i \equiv 1 \pmod 4$ for all i. Results of computations are presented in Table 2. Computations were done on one core of Xeon Silver 4210R processor clocked at 2.4 GHz on computer with 629 GB RAM. The experiments were run on Sage 9.4 [22].

Table 2

Class group computation experiments

$[K:\mathbb{Q}]$	Multiclass-im	Sage	Cl_K
8	25	0.188	$\mathcal{C}_3$
16	175	0.378	$\mathcal{C}_8 \times \mathcal{C}_{48}$
32	$2.67 \cdot 10^3$	30	$\mathcal{C}_2 \times \mathcal{C}_4^3 \times \mathcal{C}_{24} \times \mathcal{C}_{48}^2 \times C_{3360}$
64	$4.484 \cdot 10^4$	$> 5.898 \cdot 10^4$	$\mathcal{C}_2^2 \times \mathcal{C}_4^9 \times \mathcal{C}_8^3 \times \mathcal{C}_{16} \times \mathcal{C}_{48} \times \mathcal{C}_{96}^2 \times \mathcal{C}_{192}^2 \times$ $\times \mathcal{C}_{6720}^2 \times \mathcal{C}_{927360}$
128	$6.279 \cdot 10^4$		$\mathcal{C}_2^6 \times \mathcal{C}_4^{11} \times \mathcal{C}_8^{15} \times \mathcal{C}_{16}^9 \times \mathcal{C}_{32} \times \mathcal{C}_{96}^4 \times \mathcal{C}_{192}^2 \times$ $\times \mathcal{C}_{960} \times \mathcal{C}_{1920} \times \mathcal{C}_{13440}^5 \times \mathcal{C}_{443520} \times$ $\times \mathcal{C}_{20401920} \times \mathcal{C}_{554955114954240}$
256	$3.428 \cdot 10^5$		$\mathcal{C}_2^{15} \times \mathcal{C}_4^{33} \times \mathcal{C}_8^{19} \times \mathcal{C}_{16}^{15} \times \mathcal{C}_{48}^{10} \times \mathcal{C}_{96}^{19} \times \mathcal{C}_{480} \times$ $\times \mathcal{C}_{960}^7 \times \mathcal{C}_{6720} \times \mathcal{C}_{13440}^4 \times \mathcal{C}_{40320} \times \mathcal{C}_{80640}^2 \times$ $\times \mathcal{C}_{11531520}^3 \times \mathcal{C}_{219098880} \times \mathcal{C}_{438197760} \times \mathcal{C}_{514005972480} \times$ $\times \mathcal{C}_{16648653448627200} \times \mathcal{C}_{27884713764834445351472 6400}$

Timings for Sage in Table 2 are given for the class_group(proof = False) method. Similar to the case of real multiquadratic fields [12, Table 1], we have slower computations for small degrees. But algorithm of Biasse — van Vredendaal allows us to compute class groups for fields of degrees $\geqslant 64$, while Sage does not.

Computation for $\deg K = 64$

In this case, we were not able to compute class group using built-in Sage methods. For the factor base, we took prime ideals above the rational primes $3, \ldots, 173$. The result was verified using the class number formula by the method from Section 3.4 as follows. The tentative class number is $h^* = 5866281825408383486089249161216000$. The logarithm of Euler product (from class number formula) computed using Hecke [19] is equal to 11.2 ± 0.0136. We have $\left|\log \dfrac{h^* R^*}{\varepsilon h R}\right| \approx 0.00347$, $\dfrac{h^* R^*}{\varepsilon h R} \in [0.996, 1.003]$, so $\dfrac{h^* R^*}{\varepsilon h R} < \sqrt{2}$ as desired for verification of the class number.

Computation for $\deg K = 128$

To form the factor base, we took only prime ideals above rational primes p such that $p \nmid [\mathcal{O}_M : \mathbb{Z}[\theta_M]]$ for subfields M of K that were used in the class group computation algorithm. Note that the algorithm does not use all subfields of the field M, but a subset of subfields. Namely, we used prime ideals that are above the following rational primes:

$$p \in \{197, 223, 239, 257, 263, 271, 277, 331, 347, 353, 359, 367, 373, 379, 409\}.$$

Since the factor base is relatively small, we did not expect to obtain $h^* = h$. Nevertheless, the verification test was successfully passed. The logarithm of Euler product obtained using Hecke [19] is 24 ± 0.261, $\left|\log \dfrac{h^* R^*}{\varepsilon h R}\right| \approx 0.16$, $\dfrac{h^* R^*}{\varepsilon h R} \in [0.85, 1.17] < \sqrt{2}$. Thus, our computation returned the correct class number.

Computation for $\deg K = 256$

Here, we used prime ideals above the following rational primes

$$p \in \{233, 263, 347, 353, 359, 373, 421, 443, 467, 503, 509\}$$

in the factor base. At this point the verification fails and we report in the Table 2 only a multiple of the class number. Euler product computed using Hecke [19] is equal to 51 ± 0.116. This leads to $\left|\log \frac{h^* R^*}{\varepsilon h R}\right| \approx 10.38$ and so h^* contains an extra factor of size $\leqslant 32208$.

REFERENCES

1. *Buchmann J. and Paulus S.* A one way function based on ideal arithmetic in number fields. LNCS, 1997, vol. 1294, pp. 385–394.
2. *Biehl I., Buchmann J., Hamdy S., and Meyer A.* A signature scheme based on the intractability of computing roots. Des. Codes Cryptogr., 2002, vol. 25, pp. 223–236.
3. *Meyer A., Neis S., and Pfahler T.* First implementation of cryptographic protocols based on algebraic number fields. LNCS, 2001, vol. 2119, pp. 84–103.
4. *Lyubashevsky V., Peikert C., and Regev O.* On ideal lattices and learning with errors over rings. LNCS, 2010, vol. 6110, pp. 1–23.
5. *Lyubashevsky V. and Micciancio D.* Generalized compact knapsacks are collision resistant. LNCS, 2006, vol. 4052, pp. 144–155.
6. *Bauch J., Bernstein D. J., de Valence H., et al.* Short generators without quantum computers: the case of multiquadratics. Ann. Intern. Conf. on the Theory and Applications of Cryptographic Techniques, 2017, vol. 84, no. 291, pp. 27–59. Source code: `https://multiquad.cr.yp.to/software.html`.
7. *Pellet-Mary A., Hanrot G., and Stehlé D.* Approx-SVP in ideal lattices with pre-processing. LNCS, 2019, vol. 11477, pp. 685–716.
8. *Bernard O. and Roux-Langlois A.* Twisted-PHS — Using the product formula to solve Approx-SVP in ideal lattices. LNCS, 2020, vol. 12492, pp. 349–380.
9. *Buchmann J.* A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. Séminaire de Théorie des Nombres (Paris 1988/1989), Progr. Math., Birkhäuser, Boston, 1990, no. 91, pp. 27–41.
10. *Cohen H., Diaz Y Diaz F., and Olivier M.* Subexponential algorithms for class group and unit computations. J. Symbolic Computation, 1997, vol. 24, pp. 433–441.
11. *Biasse J.-F.* An L(1/3) Algorithm for Discrete Logarithm Computation and Principality Testing in Certain Number Fields. arXiv preprint, 2012. `https://arxiv.org/abs/1204.1292`.
12. *Biasse J.-F. and van Vredendaal C.* Fast multiquadratic S-unit computation and application to the calculation of class groups. The Open Book Series, 2019, vol. 2, pp. 103–118. Source code: `https://scarecryptow.org/publications/multiclass.html`.
13. *Lenstra A. K., Lenstra H. W., and Lovasz L.* Factoring polynomials with rational coefficients. Math. Ann., 1982, vol. 261, pp. 515–534.
14. *Bach E.* Explicit bounds for primality testing and related problems. Mathematics of Computation, 1990, vol. 55, no. 191, pp. 355–380.
15. *Grenié L. and Molteni G.* Explicit bounds for primality testing and related problems. Mathematics of Computation, 2018, no. 313, pp. 2483–2511.
16. *Cohen H.* A Course in Computational Algebraic Number Theory. Berlin, Springer Verlag, 1993.
17. *Bach E.* Improved approximations for Euler products. Number Theory, CMS Conf. Proc., 1995, vol. 15, pp. 13–28.
18. *Belabas K. and Friedman E.* Computing the residue of the Dedekind zeta function. Mathematics of Computation, 2015, vol. 84, no. 291, pp. 357–369.

19. *Fieker C., Hart W., Hofmann T., and Johansson F.* Nemo/Hecke: computer algebra and number theory packages for the Julia programming language. Proc. ISSAC'17, 2017, pp. 157–164. https://www.thofma.com/Hecke.jl/dev/.
20. *Bezanson J., Edelman A., Karpinski S., and Shah V. B.* Julia: A fresh approach to numerical computing. SIAM Review, 2017, vol. 59, no. 1, pp. 65–98.
21. *Cohen H.* Advanced Topics in Computational Number Theory. N.Y., Springer Science + Business Media, 2000.
22. *The Sage Developers.* SageMath, the Sage Mathematics Software System (Version 9.4). 2022, https://www.sagemath.org.

УДК 512.53 DOI 10.17223/20710410/58/4

DIRECT POWERS OF ALGEBRAIC STRUCTURES AND EQUATIONS[1]

A. Shevlyakov

Sobolev Institute of Mathematics SB RAS, Omsk, Russia
Dostoevsky Omsk State University, Omsk, Russia

E-mail: art.shevlyakov@gmail.com

We study systems of equations over graphs, posets and matroids. We give the criteria when a direct power of such algebraic structures is equationally Noetherian. Moreover, we prove that any direct power of any finite algebraic structure is weakly equationally Noetherian.

Keywords: *graphs, matroids, finite algebraic structures, direct powers, equationally Noetherian algebraic structures.*

ПРЯМЫЕ СТЕПЕНИ АЛГЕБРАИЧЕСКИХ СИСТЕМ И УРАВНЕНИЯ НАД НИМИ

А. Н. Шевляков

Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия
Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия

Изучаются системы уравнений над графами, матроидами и частично упорядоченными множествами. Доказаны критерии нетеровости по уравнениям прямых степеней алгебраических систем указанных типов. Кроме того, доказано, что прямая степень произвольной конечной алгебраической системы является слабо нетеровой по уравнениям.

Ключевые слова: *графы, матроиды, конечные алгебраические системы, прямые степени, нетеровость по уравнениям.*

1. Introduction

Let $\mathbf{K}$ be an arbitrary class of mathematical objects. One of the main problem of mathematics is to describe "simple" and "hard" objects in $\mathbf{K}$. One can do it in different ways using various techniques of algebra, geometry, calculus, etc. In the paper, we make an attempt to classify "simple" and "hard" algebraic structures by universal algebraic geometry (UAG).

Following [1], UAG is a discipline of model theory, and it deals with equations over arbitrary algebraic structures. There are many notions of UAG which allow us to separate algebraic structures with "simple" and "hard" equational properties. The main feature here is the equationally Noetherian property. Recall that an algebraic structure $\mathcal{A}$ is equationally Noetherian if any system of equations $\mathbf{S}$ is equivalent over $\mathcal{A}$ to a finite subsystem. Roughly speaking, if an algebraic structure $\mathcal{A}$ is equationally Noetherian, then its equational properties are said to be "simple". Otherwise, we assume that $\mathcal{A}$ has a complicated equational theory.

[1]The author was supported by the RSF-grant no. 22-21-00745.

Indeed, the Noetherian property is a central notion of UAG, and papers [1–3] contain the series of results which establish nice properties of equationally Noetherian algebraic structures. However, for *finite* algebraic structures the Noetherian property gives the trivial partition into "simple" and "hard" classes, since all finite algebraic structures are equationally Noetherian.

Thus, we have to propose an alternative approach in the division of finite algebraic structures into the classes with "simple" and "hard" equational properties. Our approach satisfies the following:

1) we deal with lattices of algebraic sets over a given algebraic structures (a set Y is algebraic over an algebraic structure $\mathcal{A}$ if Y is the solution set of an appropriate system of equations);
2) we use the common operations of UAG (direct products, substructures, ultraproducts etc.);
3) the partition into "simple" and "hard" algebraic structures is implemented by a list of first-order formulas Φ such that

$$\mathcal{A} \text{ is "simple"} \Leftrightarrow \mathcal{A} \text{ satisfies } \Phi.$$

In other words, the "simple" class is axiomatizable by the formulas Φ.

Namely, we offer to consider infinite direct powers $\Pi\mathcal{A}$ of an algebraic structure $\mathcal{A}$ and study the Diophantine equations over $\Pi\mathcal{A}$ instead of Diophantine equations over $\mathcal{A}$ (an equation $E(X)$ is said to be Diophantine over an algebraic structure $\mathcal{B}$ if $E(X)$ may contain occurrences of any element from $\mathcal{B}$). The decision rule in our approach is the following:

$$\mathcal{A} \text{ is "simple"} \Leftrightarrow \text{all direct powers of } \mathcal{A} \text{ are equationally Noetherian ;} \qquad (1)$$

otherwise, an algebraic structure $\mathcal{A}$ is said to be "hard".

Some results of the type (1) were obtained in [4], where we describe all groups, rings and monoids satisfying (1). For example, a group (ring) satisfies (1) iff it is abelian (respectively, with zero multiplication).

The current paper continues the study [4], and in Sections 3–5 we consider equations over the important classes of relational algebraic structures: graphs, partial orders and matroids. For each of these classes we describe algebraic structures that satisfies (1).

However, the most complicated and nontrivial part of the paper is Section 6. It contains the series of general results that hold for any direct power of any finite algebraic structure $\mathcal{A}$. In particular, we prove that any infinite system of equations $\mathbf{S}$ over $\Pi\mathcal{A}$ is equivalent to a finite system $\mathbf{S}'$ (here we do not claim $\mathbf{S}' \subseteq \mathbf{S}$). Thus, we prove that any direct power of a finite algebraic structure is weakly equationally Noetherian.

2. Basic definitions

Following [1–3], we give the main definitions of universal algebraic geometry.

Let $\mathcal{L}$ be a countable language and $\mathcal{A}$ be an algebraic structure of the language $\mathcal{L}$ ($\mathcal{L}$-structure). We consider languages of the following types: $\mathcal{L}_g = \{E^{(2)}\}$ (graph language), $\mathcal{L}_p = \{\leqslant^{(2)}\}$ (partial order language), $\mathcal{L}_m = \{P_1^{(1)}, P_2^{(2)}, \ldots\}$ (matroid language). An *equation in* $\mathcal{L}$ ($\mathcal{L}$-*equation*) is an atomic formula over $\mathcal{L}$. The examples of equations in various languages are the following: $E(x, y)$, $E(x, x)$ (language $\mathcal{L}_g$); $x \leqslant y$, $x \leqslant x$ (language $\mathcal{L}_p$); $P_1(x)$, $P_2(x, y)$ (language $\mathcal{L}_m$). Notice that the expression $x = y$ is an equation in any language $\mathcal{L}$.

A *system of $\mathcal{L}$-equations* ($\mathcal{L}$-*system* for shortness) is an arbitrary set of $\mathcal{L}$-equations. Notice that we consider only systems in a finite set of variables $X = \{x_1, x_2, \ldots, x_n\}$. The set of all solutions of $\mathbf{S}$ in an $\mathcal{L}$-structure $\mathcal{A}$ is denoted by $\mathrm{V}_{\mathcal{A}}(\mathbf{S}) \subseteq \mathcal{A}^n$. A set $Y \subseteq \mathcal{A}^n$ is said to be *algebraic over* $\mathcal{A}$ if there exists an $\mathcal{L}$-system $\mathbf{S}$ with $Y = \mathrm{V}_{\mathcal{A}}(\mathbf{S})$. If the solution set of an $\mathcal{L}$-system $\mathbf{S}$ is empty, $\mathbf{S}$ is said to be *inconsistent*. Two $\mathcal{L}$-systems $\mathbf{S}_1, \mathbf{S}_2$ are called *equivalent over an $\mathcal{L}$-structure* $\mathcal{A}$ if $\mathrm{V}_{\mathcal{A}}(\mathbf{S}_1) = \mathrm{V}_{\mathcal{A}}(\mathbf{S}_2)$. This equivalence relation is denoted by $\mathbf{S}_1 \sim \mathbf{S}_2$.

An $\mathcal{L}$-structure $\mathcal{A}$ is *$\mathcal{L}$-equationally Noetherian* if any infinite $\mathcal{L}$-system $\mathbf{S}$ is equivalent over $\mathcal{A}$ to a finite subsystem $\mathbf{S}' \subseteq \mathbf{S}$. The class of equationally Noetherian $\mathcal{L}$-structures is denoted by $\mathbf{N}$.

It is easy to prove that an $\mathcal{L}$-structure $\mathcal{A}$ is equationally Noetherian iff for any $\mathcal{L}$-system $\mathbf{S}$ there exists a number m such that the set of the first m equations of $\mathbf{S}$ is equivalent to $\mathbf{S}$ (here we essentially use the countability of the language $\mathcal{L}$).

Generalizations of the Noether property have been introduced in [3]. An $\mathcal{L}$-structure $\mathcal{A}$ is *weakly $\mathcal{L}$-equationally Noetherian* if any infinite $\mathcal{L}$-system $\mathbf{S}$ is equivalent over $\mathcal{A}$ to a finite system $\mathbf{S}'$ (here we do not claim $\mathbf{S}' \subseteq \mathbf{S}$). The class of weakly equationally Noetherian $\mathcal{L}$-structures is denoted by $\mathbf{N}'$. Obviously, $\mathbf{N} \subseteq \mathbf{N}'$.

Let $\mathcal{A}$ be an $\mathcal{L}$-structure. By $\mathcal{L}(\mathcal{A})$ we denote the language $\mathcal{L} \cup \{a : a \in \mathcal{A}\}$ extended by new constant symbols which correspond to elements of $\mathcal{A}$. The language extension allows us to use constants in equations. The examples of equations in extended languages are the following (below $\mathcal{G}$, $\mathcal{M}$ are graph and matroid respectively): $E(x, a)$ (language $\mathcal{L}_g(\mathcal{G})$ and $a \in \mathcal{G}$); $P_2(a, x)$, $P_3(x, b, c)$, $P_4(a, x, y, b)$ (language $\mathcal{L}_m(\mathcal{M})$ and $a, b, c \in \mathcal{M}$). Obviously, the class of $\mathcal{L}(\mathcal{A})$-equations is wider than the class of $\mathcal{L}$-equations, so an $\mathcal{L}$-equationally Noetherian $\mathcal{L}$-algebra may lose this property in the language $\mathcal{L}(\mathcal{A})$.

Let $\mathcal{A}$ be an $\mathcal{L}$-structure. An element of a direct power $\Pi\mathcal{A} = \prod_{i \in I} \mathcal{A}$ is denoted by a sequence in square brackets $[a_i \mid i \in I]$. Functions and relations over $\Pi\mathcal{A}$ have the coordinate-wise definition. For example, any relation $R^m \in \mathcal{L}$ is defined on $\Pi\mathcal{A}$ as follows:

$$R\left([a_i^{(1)} \mid i \in I], [a_i^{(2)} \mid i \in I], \ldots, [a_i^{(m)} \mid i \in I]\right) \Leftrightarrow R\left(a_i^{(1)}, a_i^{(2)}, \ldots, a_i^{(m)}\right) \text{ for each } i \in I.$$

The map $\pi_k \colon \Pi\mathcal{A} \to \mathcal{A}$ is called a *projection onto the k-th coordinate* if $\pi_k([a_i \mid i \in I]) = a_k$.

Let $E(X)$ be an $\mathcal{L}(\Pi\mathcal{A})$-equation over a direct power $\Pi\mathcal{A}$. We may rewrite $E(X)$ in the form $E(X, \overrightarrow{\mathbf{C}})$, where $\overrightarrow{\mathbf{C}}$ is an array of constants occurring in the equation $E(X)$. One can introduce the *projection of an equation* onto the i-th coordinate as follows:

$$\pi_i(E(X)) = \pi_i(E(X, \overrightarrow{\mathbf{C}})) = E(X, \pi_i(\overrightarrow{\mathbf{C}})),$$

where $\pi_i(\overrightarrow{\mathbf{C}})$ is an array of the i-th coordinates of the elements from $\overrightarrow{\mathbf{C}}$. For example, an $\mathcal{L}_g(\Pi\mathcal{G})$-equation $E(x, [a_1, a_2, a_3, \ldots])$ has the following projections:

$$E(x, a_1),$$
$$E(x, a_2),$$
$$E(x, a_3),$$
$$\ldots$$

Similarly, a matroid equation $P_4(x, [a_1, a_2, a_3, \ldots], y, [b_1, b_2, b_3, \ldots])$ has the projections

$$P_4(x, a_1, y, b_1),$$
$$P_4(x, a_2, y, b_2),$$
$$P_4(x, a_3, y, b_3),$$
$$\ldots$$

Let us take an $\mathcal{L}(\Pi\mathcal{A})$-system $\mathbf{S} = \{E_j(X) : j \in J\}$. The i-th projection of $\mathbf{S}$ is the $\mathcal{L}(\mathcal{A})$-system $\pi_i(\mathbf{S}) = \{\pi_i(E_j(X)) : j \in J\}$. The projections of an $\mathcal{L}(\Pi\mathcal{A})$-system $\mathbf{S}$ allow to describe the solution set of $\mathbf{S}$ by

$$\mathrm{V}_{\Pi\mathcal{A}}(\mathbf{S}) = \{[P_i \mid i \in I] : P_i \in \mathrm{V}_{\mathcal{A}}(\pi_i(\mathbf{S}))\}. \tag{2}$$

Lemma 1. Let $\mathbf{S} = \{E_j(X) : j \in J\}$ be an $\mathcal{L}(\Pi\mathcal{A})$-system over $\Pi\mathcal{A}$. The system $\mathbf{S}$ is consistent iff so are all projections $\pi_i(\mathbf{S})$. Moreover, if $\mathcal{A}$ is $\mathcal{L}(A)$-equationally Noetherian, then an inconsistent $\mathcal{L}(\Pi\mathcal{A})$-system $\mathbf{S}$ is equivalent to a finite subsystem.

Proof. The first assertion follows directly from (2). Suppose $\mathcal{A}$ is $\mathcal{L}$-equationally Noetherian and $\pi_i(\mathbf{S})$ is inconsistent. Hence, $\pi_i(\mathbf{S})$ is equivalent to its finite inconsistent subsystem $\{\pi_i(E_j(X)) : j \in J'\}$, $|J'| < \infty$, and the subsystem $\mathbf{S}' = \{E_j(X) : j \in J'\} \subseteq \mathbf{S}$ is also inconsistent. ■

3. Graphs

Recall that a *graph* is an algebraic structure of the language $\mathcal{L}_g = \{E^{(2)}\}$ satisfying the following axioms:

$$\forall x\, \neg E(x, x) \text{ (no loops)},$$
$$\forall x \forall y\, E(x, y) \to E(y, x) \text{ (symmetry)}.$$

Theorem 1. An infinite direct power $\Pi\mathcal{G} = \prod\limits_{i \in I} \mathcal{G}$ of a graph G is $\mathcal{L}_g(\Pi\mathcal{G})$-equationally Noetherian iff $\mathcal{G}$ satisfies the quasi-identity

$$\forall x_1 \forall x_2 \forall x_3 \forall x_3\ (E(x_1, x_2) \wedge E(x_2, x_3) \wedge E(x_3, x_4) \to E(x_4, x_1)). \tag{3}$$

Proof. Let us prove the "if" part of the statement.

Let $\mathbf{S}$ be an $\mathcal{L}_g(\Pi\mathcal{G})$-system in variables $X = \{x_1, \ldots, x_n\}$. One can rewrite $\mathbf{S}$ as a finite union of systems

$$\mathbf{S} = \bigcup_{j=1}^{n} \mathbf{S}_j \bigcup \mathbf{S}_0,$$

where $\mathbf{S}_j = \{E(x_j, \mathbf{c}_k) : k \in K_j\}$ and $\mathbf{S}_0$ is the system of equations of the following types: $E(x_i, x_j)$, $x_i = x_j$, $x_i = \mathbf{c}_j$. Obviously, the system $\mathbf{S}_0$ is equivalent to a finite subsystem. Hence, it is sufficient to prove that each system $\mathbf{S}_j$ in one variable x_j is equivalent to a finite subsystem.

Let us write the coordinate-wise versions of the system $\mathbf{S}_j$:

$$\pi_i(\mathbf{S}_j) = \{E(x_j, \pi_i(\mathbf{c}_k)) : k \in K_j\},\ i \in I,$$

where $\pi_i(\mathbf{c}_k)$ is the i-th coordinate of an element $\mathbf{c}_k$.

If for each i the equations $\{E(x_j, \pi_i(\mathbf{c}_k)) : k \in K_j\}$ have the same solution sets, then $\mathbf{S}_j$ is equivalent to a single equation $E(x_j, \mathbf{c}_k) \in \mathbf{S}_j$ for arbitrary $k \in K_j$. Otherwise, there exists an index i such that

$$Y_1 = \mathrm{V}_{\mathcal{G}}(E(x_j, \pi_i(\mathbf{c}_{k_1}))) \neq \mathrm{V}_{\mathcal{G}}(E(x_j, \pi_i(\mathbf{c}_{k_2}))) = Y_2$$

for some $k_1, k_2 \in K_j$.

If $Y_1 \cap Y_2 = \varnothing$, then $\mathbf{S}_j$ is inconsistent and obviously equivalent to the subsystem $\{E(x_j, \mathbf{c}_{k_1}), E(x_j, \mathbf{c}_{k_2})\}$. Below we may assume $Y_1 \nsubseteq Y_2$ and one can take elements $b_1, b_2 \in \mathcal{G}$ such that $b_1 \in Y_1 \setminus Y_2$, $b_2 \in Y_1 \cap Y_2$, i.e., $E(b_1, \pi_i(\mathbf{c}_{k_1}))$, $E(b_2, \pi_i(\mathbf{c}_{k_1}))$ and $E(b_2, \pi_i(\mathbf{c}_{k_2}))$.

Since the quasi-identity (3) is true in $\Pi\mathcal{G}$, we have $E(b_1, \pi_i(\mathbf{c}_{k_2}))$, which contradicts the choice of the element b_1.

Let us prove the "only if" part of the statement. Assume the quasi-identity (3) does not hold in $\mathcal{G}$, i.e., there exist elements a_1, a_2, a_3, a_4 with $E(a_1, a_2)$, $E(a_2, a_3)$, $E(a_3, a_4)$, $\neg E(a_4, a_1)$. Consider the $\mathcal{L}_g(\Pi\mathcal{G})$-system $\mathbf{S}$ of the following equations:

$$E(x, [a_2, a_2, a_2 \ldots]),$$
$$E(x, [a_4, a_2, a_2 \ldots]),$$
$$E(x, [a_4, a_4, a_2 \ldots]),$$
$$\ldots$$

Let $\mathbf{S}_n$ be the subsystem of $\mathbf{S}$ formed by the first n equations of $\mathbf{S}$.

The point $\mathbf{a} = [\underbrace{a_3, a_3, \ldots, a_3}_{n-1 \text{ times}}, a_1, a_1, \ldots]$ satisfies $\mathbf{S}_n$ but $\mathbf{a}$ does not satisfy the $(n+1)$-th equation of $\mathbf{S}$. Thus, $\mathbf{S}_n$ is not equivalent to $\mathbf{S}$ for any n, and $\Pi\mathcal{G}$ is not $\mathcal{L}_g(\Pi\mathcal{G})$-equationally Noetherian. ■

Corollary 1. If a graph $\mathcal{G}$ contains a triangle (i.e., there exist vertices $x_1, x_2, x_3 \in \mathcal{G}$ with $E(x_1, x_2)$, $E(x_2, x_3)$, $E(x_3, x_1)$), then $\Pi\mathcal{G}$ is not $\mathcal{L}_g(\Pi\mathcal{G})$-equationally Noetherian.

Proof. Obviously, the condition of Theorem 1 fails for such graphs, since there are no loops in $\mathcal{G}$. ■

Let $\mathbf{K} = \{\mathcal{G} : \Pi\mathcal{G} \in \mathbf{N}\}$ be the set of all graphs with equationally Noetherian direct powers. Theorem 1 gives that the class $\mathbf{K}$ is axiomatizable. The class $\mathbf{K}$ may be also described by forbidden graphs and distance functions.

Let us give the explicit examples of graphs $\mathcal{G} \in \mathbf{K}$.

One can directly prove that the disjoint union $\mathcal{G} = \mathcal{G}_1 \sqcup \mathcal{G}_2$ has an equationally Noetherian direct power $\Pi\mathcal{G}$ if both graphs satisfy the quasi-identity (3). Thus, there arises a question: is there a *connected* graph $\mathcal{G}$ with n vertices such that any direct power $\Pi\mathcal{G}$ is $\mathcal{L}_g(\Pi\mathcal{G})$-equationally Noetherian?

The answer is positive. Let us define the following graph $\mathcal{G}$ with the vertex set $\{x_0, x_1, \ldots, x_n, x_{n+1}\}$ and edges $\{E(x_0, x_i), E(x_i, x_{n+1}) : 1 \leqslant i \leqslant n\}$. The direct check gives that $\mathcal{G}$ satisfies (3), contains $n+2$ vertices, and $\mathcal{G}$ is connected.

4. Partial orders

A *partial order* $\mathcal{P}$ is an algebraic structure of the language $\mathcal{L}_p = \{\leqslant^{(2)}\}$ such that $\mathcal{P}$ satisfies the following axioms:

$$\forall x\ (x \leqslant x),$$
$$\forall x \forall y\ (x \leqslant y) \wedge (y \leqslant x) \to (x = y),$$
$$\forall x \forall y\ (x \leqslant y) \wedge (y \leqslant z) \to (x \leqslant z).$$

A partial order $\mathcal{P}$ is said to be *non-trivial* if there exists a pair $a, b \in \mathcal{P}$ such that $a < b$ (i.e., $a \leqslant b$ and $a \neq b$).

Theorem 2. Let $\mathcal{P}$ be a non-trivial partial order and $\Pi\mathcal{P}$ be an infinite direct power of $\mathcal{P}$. Then $\Pi\mathcal{P}$ is not $\mathcal{L}_p(\Pi\mathcal{P})$-equationally Noetherian.

Proof. Since $\mathcal{P}$ is non-trivial, there exist $a, b \in \mathcal{P}$ with $a < b$. It is sufficient to show that an infinite direct power $\Pi\mathcal{E} \subseteq \Pi\mathcal{P}$ of the partial order $\mathcal{E} = \{a, b\}$ is not $\mathcal{L}_p(\Pi\mathcal{E})$-equationally Noetherian.

Indeed, one should consider the following infinite $\mathcal{L}_p(\Pi\mathcal{E})$-system $\mathbf{S}$:

$$\begin{gathered} x \leqslant [b, b, b, \ldots,] \\ x \leqslant [a, b, b, \ldots,] \\ x \leqslant [a, a, b, \ldots,] \\ \ldots \end{gathered}$$

Obviously, the unique solution of $\mathbf{S}$ is $[a, a, a, \ldots,]$. However, the solution set of any finite subsystem of $\mathbf{S}$ contains a point $[\underbrace{a, a, a, \ldots, a}_{n \text{ times}}, b, b, b, \ldots]$ for sufficiently large n. Thus, $\mathbf{S}$ is not equivalent to any finite subsystem. ■

5. Matroids

One can consider a *matroid* $\mathcal{M}$ as an algebraic structure of an infinite language $\mathcal{L}_m = \{P_1^{(1)}, P_2^{(2)}, P_3^{(3)}, \ldots\}$, where each predicate symbol P_n has the following interpretation:

$$P_n(x_1, \ldots, x_n) \Leftrightarrow \text{ the set } \{x_1, \ldots, x_n\} \text{ is independent in } \mathcal{M}.$$

Moreover, any matroid satisfies the following axioms:

$$\forall x_1 \ldots \forall x_n \left(\bigvee_{i \neq j} (x_i = x_j) \to \neg P_n(x_1, \ldots, x_n) \right),$$

$$\forall x_1 \ldots \forall x_n \left(P_n(x_1, \ldots, x_n) \to \bigwedge_{i=1}^{n} P_{n-1}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \right) \ (n > 1),$$

$$\forall x_1 \ldots \forall x_n, \forall y_1 \ldots \forall y_{n+1} \left(P_n(x_1, \ldots, x_n) \wedge P_{n+1}(y_1, \ldots, y_{n+1}) \to \bigvee_{i=1}^{n+1} P_{n+1}(x_1, \ldots, x_n, y_i) \right),$$

$$\forall x_1 \ldots \forall x_n \ (P_n(x_1, \ldots, x_n) \to P_n(x_{\sigma(1)}, \ldots, x_{\sigma(n)})) \text{ for any permutation } \sigma.$$

Notice that a direct power $\Pi\mathcal{M}$ of a matroid $\mathcal{M}$ is not necessarily a monoid itself. However, here we study direct powers of matroids, since the algebraic geometry over $\Pi\mathcal{M}$ may clarify algebraic and geometric properties of the original matroid $\mathcal{M}$.

Lemma 2. Let $\mathcal{M}$ be a matroid with $P_3(a, b, c)$ for some $a, b, c \in \mathcal{M}$. Then any infinite direct power $\Pi\mathcal{M}$ is not $\mathcal{L}_m(\Pi\mathcal{M})$-equationally Noetherian.

Proof. Let us consider a system $\mathbf{S}$ of $\mathcal{L}_m(\Pi\mathcal{M})$-equations

$$\begin{gathered} P_2(x, [a, a, a, \ldots]), \\ P_2(x, [b, a, a, \ldots]), \\ P_2(x, [b, b, a, \ldots]), \\ \ldots \end{gathered}$$

Denote by $\mathbf{S}_n$ the first n equations of $\mathbf{S}$. Clearly, $\mathbf{S}_n$ is satisfied by the point

$$[\underbrace{c, c, \ldots, c}_{n \text{ times}}, b, b, \ldots].$$

However, this point does not belong to the solution set of $\mathbf{S}$, since the predicate

$$P_2([\underbrace{c, c, \ldots, c}_{n \text{ times}}, b, b, \ldots], [\underbrace{b, b, \ldots, b}_{n+1 \text{ times}}, a, a, \ldots])$$

is not true for the $(n+1)$-th coordinate. ■

According to Lemma 2, any matroid $\mathcal{M}$ with $\Pi\mathcal{M}$-equationally Noetherian direct power $\Pi\mathcal{M}$ may be represented by a graph $\mathcal{G}(\mathcal{M})$ such that

1) the vertex set of $\mathcal{G}(M)$ coincides with the set $\mathcal{M}$;
2) $P_2(a, b) \Leftrightarrow E(a, b)$.

Hence, such matroids may be classified by the analogue of Theorem 2.

Theorem 3. A direct power $\Pi\mathcal{M}$ of a matroid $\mathcal{M}$ is $\mathcal{L}_m(\mathcal{M})$-equationally Noetherian iff $\mathcal{M}$ satisfies the following axioms:

$$\forall x \forall y \forall z \; \neg P_3(x, y, z),$$
$$\forall x_1 \forall x_2 \forall x_3 \forall x_4 \; \left(P_2(x_1, x_2) \wedge P_2(x_2, x_3) \wedge P_2(x_3, x_4) \rightarrow P_2(x_4, x_1)\right).$$

Proof. The proof immediately follows from Lemma 2, Theorem 2 and the correspondence $\mathcal{M} \leftrightarrow \mathcal{G}(\mathcal{M})$. ■

6. Direct powers of finite structures

Let us prove a general fact about direct powers of arbitrary finite algebraic structures. The proof of the following theorem is rather complicated, so its main steps are explained in Example 1.

Theorem 4. Let $\mathcal{A}$ be a finite $\mathcal{L}$-structure. Then any direct power $\Pi\mathcal{A} = \prod\limits_{i \in I} \mathcal{A}$ is weakly $\mathcal{L}(\Pi\mathcal{A})$-equationally Noetherian.

Proof. Let $\mathbf{S} = \{E_j(X, \overrightarrow{\mathbf{C}_\mathbf{j}}) : j \in J\}$ be an infinite $\mathcal{L}(\Pi\mathcal{A})$-system over $\Pi\mathcal{A}$ and $\pi_i(\mathbf{S}) = \{E_j(X, \pi_i(\overrightarrow{\mathbf{C}_\mathbf{j}})) : j \in J\}$ $(i \in I)$ be the projections of $\mathbf{S}$ onto all coordinates of $\Pi\mathcal{A}$. Notice that any system $\pi_i(\mathbf{S})$ is a system of $\mathcal{L}(\mathcal{A})$-equations over $\mathcal{A}$.

Since $\mathcal{A}$ is finite, there exists a finite number of equations $M = \{E_j(X, \pi_i(\overrightarrow{\mathbf{C}_\mathbf{j}})) : (i, j) \in$ $\in K\}$ $(|K| < \infty)$ such that any $E_j(X, \pi_i(\overrightarrow{\mathbf{C}_\mathbf{j}})) \in \bigcup\limits_{i \in I} \pi_i(\mathbf{S})$ is equivalent over $\mathcal{A}$ to an appropriate equation from M. Hence, each $\pi_i(\mathbf{S})$ is equivalent to a subsystem $\mathbf{S}'_i \subseteq M$ over $\mathcal{A}$. The idea of the further proof is the following: we try to wrap all systems $\mathbf{S}'_i$ into a finite number of equations $\mathbf{S}'$ over $\Pi\mathcal{A}$.

Let us define an $\mathcal{L}(\Pi\mathcal{A})$-system $\mathbf{S}'$ by the following procedure.

S t e p 0 . Put

$$\mathbf{S}_0 = \bigcup_{(i,j) \in K} E_j(X, \overrightarrow{\mathbf{C}_\mathbf{j}}) \subseteq \mathbf{S}$$

$(|\mathbf{S}_0| \leqslant |K|)$ and $\mathbf{S}' = \mathbf{S}_0$. The main property of $\mathbf{S}_0$ is the following: each equation from M occurs in some projection of an equation from $\mathbf{S}_0$. Let us arbitrarily enumerate equations in the set M, i.e., each equation from M has the number $s \in [1, |K|]$.

S t e p s $(1 \leqslant s \leqslant |K|)$. Let us take the s-th equation $E_j(X, \pi_i(\overrightarrow{\mathbf{C_j}}))$ from M and define the following sets of indexes: $I_0 = \{l \in I : E_j(X, \pi_i(\overrightarrow{\mathbf{C_j}})) \in \mathbf{S}'_l\}$, $I_1 = I \setminus I_0$. In other words, I_0 is the set of all indexes l such that the given equation from M occurs in the system $\mathbf{S}'_l$. Define a set $M_s = \{D_l(X) : l \in I\}$ of $\mathcal{L}(\mathcal{A})$-equations as follows:

$$D_l(X) = \begin{cases} E_j(X, \pi_i(\overrightarrow{\mathbf{C_j}})), & \text{if } l \in I_0, \\ E_j(X, \pi_l(\overrightarrow{\mathbf{C_j}})), & \text{if } l \in I_1. \end{cases}$$

The sense of the set M_s is the following. If a system $\mathbf{S}'_l$ contains $E_j(X, \pi_i(\overrightarrow{\mathbf{C_j}})) \in M$, then we take this equation as the k-th projection in M_s. Otherwise, the l-th projection in M_s is taken from the equation $E_j(X, \overrightarrow{\mathbf{C_j}}) \in \mathbf{S}_0$.

The $\mathcal{L}(A)$-equations M_s may be wrapped into the $\mathcal{L}(\Pi\mathcal{A})$-equation $D_s(X, \overrightarrow{\mathbf{D_s}})$ such that $\pi_l(D_s(X, \overrightarrow{\mathbf{D_s}})) = D_l(X)$.

We put $\mathbf{S}' := \mathbf{S}' \cup D_s(X, \overrightarrow{\mathbf{D_s}})$ and go to the following step $(s+1)$.

By the definition of the system $\mathbf{S}'$, the i-th projection $\pi_i(\mathbf{S}')$ contain all equations from $\mathbf{S}'_i \sim \pi_i(\mathbf{S})$. Hence, $\pi_i(\mathbf{S}') \sim \pi_i(\mathbf{S})$ over $\mathcal{A}$, and finally $\mathbf{S}' \sim \mathbf{S}$ over $\Pi\mathcal{A}$. ■

The following example explains the technique and denotations of Theorem 4.

Example 1. Let $\mathcal{G}$ be the graph with vertices $\{a, b, c\}$ and edges $E(a, b)$, $E(b, c)$, $E(c, a)$ (i.e., $\mathcal{G}$ is a complete graph). Let us consider an infinite $\mathcal{L}(\Pi\mathcal{G})$-system $\mathbf{S}$ of equations:

$$E(x, [a, a, a, a, a, a, \ldots]),$$
$$E(x, [b, a, a, a, a, a, \ldots]),$$
$$E(x, [b, c, a, a, a, a, \ldots]),$$
$$E(x, [b, c, b, a, a, a, \ldots]),$$
$$E(x, [b, c, b, c, a, a, \ldots]),$$
$$\ldots$$

The projections $\pi_i(S)$ are the following (we omit in the projections equations which occur earlier):

$$\pi_1(\mathbf{S}) = \{E(x, a), E(x, b)\},$$
$$\pi_2(\mathbf{S}) = \{E(x, a), E(x, c)\},$$
$$\pi_3(\mathbf{S}) = \{E(x, a), E(x, b)\},$$
$$\pi_4(\mathbf{S}) = \{E(x, a), E(x, c)\},$$
$$\ldots$$

The set M consists of the equations $E(x, a)$, $E(x, b)$, $E(x, c)$ (any equation from $\bigcup_{i=1}^{n} \pi_i(\mathbf{S})$ is equivalent to one of the given equations). Since the third equation of $\mathbf{S}$ contains all equations from M as projections, we may put $\mathbf{S}_0 = \{E(x, [b, c, a, a, a, a, \ldots])\}$ (the set K here is $\{(1, 3), (2, 3), (3, 3)\}$). For the projections $\pi_i(\mathbf{S})$ we have

$$\begin{aligned} \pi_{2k+1}(\mathbf{S}) &\sim \{E(x, a), E(x, b)\} = \mathbf{S}'_{2k+1}, \\ \pi_{2k}(\mathbf{S}) &\sim \{E(x, a), E(x, c)\} = \mathbf{S}'_{2k}. \end{aligned}$$

Now we construct the final system $\mathbf{S}'$ with $|\mathbf{S}_0| + |M| = 1 + 3 = 4$ equations. First, we put $\mathbf{S}' = \mathbf{S}_0$ and make the following three steps:

1) We take $E(x,a) \in M$. Since this equation occurs in any system $\mathbf{S}'_i$ ($I_0 = \mathbb{N}$, $I_1 = \varnothing$), we add to $\mathbf{S}'$ the equation $E(x,[a,a,a,a,a,\ldots])$.
2) Take $E(x,b) \in M$. Since $E(x,b)$ occurs in the systems $\mathbf{S}_i$ with odd i ($I_0 = \{1,3,\ldots\}$, $I_1 = \{2,4,\ldots\}$), we should add to $\mathbf{S}'$ an equation of the form $E(x,[b,*,b,*,b,*,\ldots])$. The elements for even positions are taken from the equation from $\mathbf{S}_0$, and we obtain the equation $E(x,[b,c,b,a,b,a,\ldots])$. The last equation is added to $\mathbf{S}'$.
3) For the equation $E(x,c) \in M$ we make dual operations. Since $E(x,c)$ occurs in the systems $\mathbf{S}_i$ with even i ($I_0 = \{2,4,\ldots\}$, $I_1 = \{1,3,\ldots\}$), we should add to $\mathbf{S}'$ an equation of the form $E(x,[*,c,*,c,*,c,\ldots])$. The elements for odd positions are taken from the equation from $\mathbf{S}_0$, and we obtain the equation $E(x,[b,c,a,c,a,c,\ldots])$. Also, we add the last equation to $\mathbf{S}'$.

Thus, the final system $\mathbf{S}'$ consists of the following equations:

$$E(x,[b,c,a,a,a,a,\ldots]),$$
$$E(x,[a,a,a,a,a,a,\ldots]),$$
$$E(x,[b,c,b,a,b,a,\ldots]),$$
$$E(x,[b,c,a,c,a,c,\ldots]).$$

It is easy to see that all projections $\pi_i(\mathbf{S}')$ are equivalent over $\mathcal{G}$ to the systems $\mathbf{S}'_i$. Thus, $\mathbf{S}'$ is equivalent to $\mathbf{S}$.

The ideas of Theorem 4 allow us to estimate uniformly the minimal number of equations in the finite system $\mathbf{S}'$.

Corollary 2. Let $\mathbf{S}$ be a system of $\mathcal{L}(\Pi\mathcal{A})$-equations in n variables over a direct power $\Pi\mathcal{A}$ of a finite $\mathcal{L}$-structure $\mathcal{A}$, $|\mathcal{A}| = k$. Then $\mathbf{S}$ is equivalent to a system $\mathbf{S}'$ with at most 2^{k^n+1} equations.

Proof. Since we deal with equations in n variables, all algebraic sets over $\mathcal{A}$ are the subsets of the affine space $\mathcal{A}^n$, $|\mathcal{A}^n| = k^n$. Hence, there exists at most 2^{k^n} different algebraic sets over $\mathcal{A}$. Since the set M in Theorem 4 consists of pairwise non-equivalent equations, we have $|M| \leqslant 2^{k^n}$.

The final system $\mathbf{S}'$ consists of at most $|M| + |M| = 2|M|$ equations ($|\mathbf{S}_0| = |M|$, and $|M|$ iterations of the procedure add to $\mathbf{S}'$ exactly $|M|$ equations). Thus, we obtain $|\mathbf{S}'| \leqslant 2 \cdot 2^{k^n} = 2^{k^n+1}$. ■

REFERENCES

1. *Daniyarova E. Yu., Myasnikov A. G., and Remeslennikov V. N.* Unification theorems in algebraic geometry. Algebra Discr. Math., 2008, vol. 1, pp. 80–112.
2. *Daniyarova E. Yu., Myasnikov A. G., and Remeslennikov V. N.* Algebraic geometry over algebraic structures, II: Foundations. J. Math. Sci., 2012, vol. 183, pp. 389–416.
3. *Daniyarova E. Yu., Myasnikov A. G., and Remeslennikov V. N.* Algebraic geometry over algebraic structures, III: Equationally noetherian property and compactness. Southern Asian Bull. Math., 2011, vol. 35, no. 1, pp. 35–68.
4. *Shevlyakov A. N. and Shahryari M.* Direct products, varieties, and compactness conditions. Groups Complexity Cryptology, 2017, vol. 9, no. 2, pp. 159–166.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 003.26.09+004.021 DOI 10.17223/20710410/58/5

О ПОИСКЕ РАЗНОСТНЫХ СООТНОШЕНИЙ ДЛЯ ПОДСТАНОВКИ ALZETTE С МАКСИМАЛЬНЫМ ИЛИ БЛИЗКИМ К НЕМУ ЗНАЧЕНИЕМ РАЗНОСТНОЙ ХАРАКТЕРИСТИКИ

А. А. Дмух, Д. О. Пасько

Академия криптографии Российской Федерации, г. Москва, Россия

E-mail: graphin@rambler.ru, pasko.do@yandex.ru

Предлагается подход «разностная встреча посередине» для построения согласованных систем локальных разностных соотношений для подстановки Alzette, который позволяет получать системы с максимальными или близкими к максимальным разностными характеристиками. С помощью этого подхода расширены результаты по оценке разностных характеристик подстановки Alzette, полученные разработчиками подстановки, при этом с меньшей трудоёмкостью.

Ключевые слова: *подстановка, Alzette, разностная характеристика, разностный метод.*

SEARCH FOR DIFFERENCES FOR ALZETTE S-BOX WITH MAXIMUM OR CLOSE TO MAXIMUM DIFFERENTIAL CHARACTERISTIC PROBABILITY

A. A. Dmukh, D. O. Pasko

Academy of Cryptography of the Russian Federation, Moscow, Russia

In this paper, we describe a “differential meet-in-the-middle” method for obtaining differences for 64-bit ARX permutation Alzette with maximum or close to maximum differential characteristic probability (DCP). The method is based on testing the high-probability differences in the middle rounds of Alzette and extending them to the previous and following rounds. Using this method, we obtain 7 differences for 4-rounds Alzette with DCP 2^{-6}, 1 difference for 5-rounds Alzette with DCP 2^{-10}, and 1 difference for 6-rounds Alzette with DCP 2^{-18}. Same differences for 4 and 5 rounds were obtained by the developers of Alzette as the differences with maximum DCP, but our method has lower complexity: taking the calculation of probability for a round difference as a single operation, it’s 36 operations (4 rounds), 135 operations (5 rounds) and 486 operations (6 rounds) for our method and more than $1.29 \cdot 10^8$ operations (4 rounds), $2 \cdot 1.29 \cdot 10^8$ operations (5 rounds) and $1.03 \cdot 10^{14}$ operations (6 rounds) for Alzette developers’ method. Also, we obtain 6 differences for 7-rounds Alzette with DCP 2^{-27} and 11 differences for 8-rounds Alzette with DCP 2^{-35} with complexity $\leqslant 5 \cdot 10^{13}$ operations for both cases. For these number of rounds by the developers of Alzette were obtained only the higher bounds for maximum DCP: 2^{-24} (7 rounds) and 2^{-32} (8 rounds). Our estimations of Alzette developers’ method complexity is

$\geqslant 2.97 \cdot 10^{16}$ operations for 7-rounds Alzette and $\geqslant 2.97 \cdot 10^{16} + 4.75 \cdot 10^{12}$ operations for 8-rounds Alzette.

Keywords: *permutation, Alzette, differential characteristic, differential method.*

Введение

В ноябре 2019 г. коллективом авторов в работе [1] была предложена подстановка Alzette, действующая на двоичных векторах длины 64 и предназначенная к использованию в низкоресурсных блочных шифрах и криптографических примитивах на их основе. Подстановка Alzette представляет собой итеративную ARX-конструкцию, т. е. в ней на каждой итерации (раунде) используются лишь следующие операции над двоичными векторами длины 32: модульного сложения по $\bmod 2^{32}$ ($\boxplus$), циклического сдвига ($\ggg\ a$ — в данном случае циклический сдвиг вправо на a позиций, Rotation) и побитового сложения ($\oplus$, XOR). Выбор ARX-конструкции для построения подстановки обусловлен удобством и эффективностью реализации указанных операций на платформах с ограниченными ресурсами.

Конкретный выбор количества итераций и значений циклических сдвигов для подстановки Alzette в [1], по словам авторов, был сделан для достижения разумного баланса между эффективностью реализации и приемлемыми криптографическими характеристиками. Таким образом, в [1] для подстановки Alzette остановились на количестве итераций, равном 4, и значениях циклических сдвигов, приведённых на рис. 1, где $x, y, u, v, c \in V_{32}$.

Подстановка Alzette из $\mathrm{S}_{V_{64}}$ является дальнейшим развитием ARX-конструкций, предназначенных для использования в низкоресурсных криптографических примитивах: подстановки из криптопримитива SATURNIN [2] (используется подстановка из $\mathrm{S}_{V_{16}}$, получаемая из нескольких слоев 4-битных подстановок, перемежаемых слоями максимально рассеивающих линейных преобразований соответствующей размерности) и подстановки SPARX [3] (ARX-конструкция, реализующая подстановку из $\mathrm{S}_{V_{32}}$).

В отличие от подстановок из [2, 3], для которых некоторые (в работе [2] — все) криптографические характеристики подсчитаны точно на ЭВМ, для подстановки Alzette в работе [1] ряд характеристик, в частности разностные и линейные, оценены на ЭВМ с применением подхода, предложенного в [4]. Данный подход использует ряд предположений и допущений. В частности, предполагается марковость ARX-конструкции, хотя сами же авторы работы [4] показывают, что алгоритм Speck, для которого получены значения для оценок наилучших разностных и линейных характеристик для некоторого числа итераций, свойством марковости не обладает. Таким образом, строгих теоретических обоснований полученных в [1] оценок для наилучших разностных и линейных характеристик не приводится.

В данной работе с использованием результатов [5, 6] предложен подход по проверке оценок максимального значения разностной характеристики подстановки Alzette, полученных в [1], который, помимо прочего, позволяет предъявлять согласованные системы локальных разностных соотношений [5] (далее — с.с.л.р.с.), соответствующие заданным разностным соотношениям, с максимальными значениями разностных характеристик.

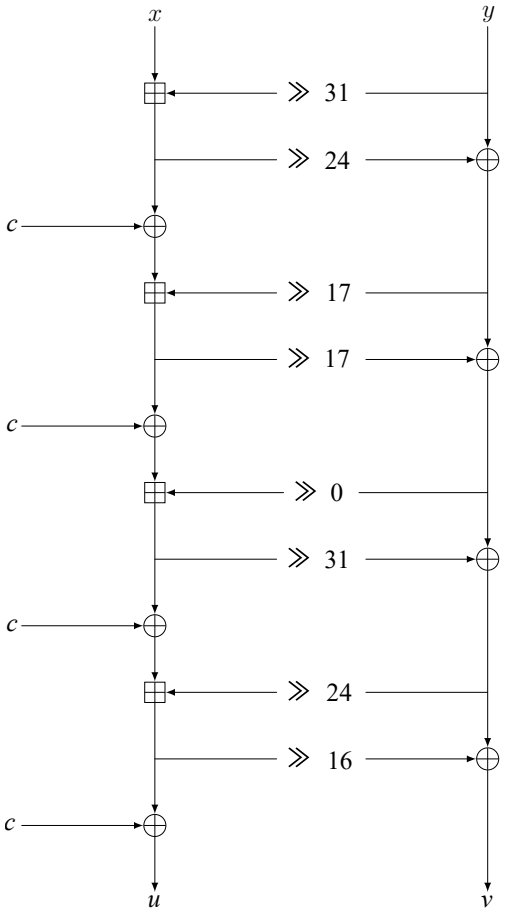

Рис. 1. Подстановка Alzette

1. Разностные характеристики подстановки Alzette

Как и в [1], будем искать разностные соотношения для подстановки Alzette $(A, B) \in \in V_{64} \times V_{64}$ с максимально возможными значениями разностной характеристики $P^{\mathrm{ALZ}}_{A,B} = \mathsf{P}[\mathrm{ALZ}(x) \oplus \mathrm{ALZ}(x \oplus A) = B]$, где вероятность $\mathsf{P}[\dots]$ вычисляется в предположении случайного равновероятного распределения x на V_{64}, а $\mathrm{ALZ}(x) \in V_{64}$ — результат применения подстановки Alzette к вектору x.

Представим подстановку Alzette как четыре итерации блочного ARX-шифра. В таком случае, в соответствии с [5], с.с.л.р.с. можно строить последовательно, например от первой итерации к последней. При этом необходимо строить локальные разностные соотношения (относительно побитового сложения $\oplus$) для всех трёх используемых операций: $\boxplus$, $\ggg a$ и $\oplus$. Если для последних двух операций построение таких соотношений (в том числе удовлетворяющих условию согласованности и имеющих значение разностной характеристики 1) — тривиальная задача, то для построения локальных вероятностных соотношений модульного сложения (и, главное, оценки их разностных характеристик) воспользуемся результатами, полученными в [5, 6]. В соответствии с [5], вместо точного значения вероятности $P^{\mathrm{ALZ}}_{A,B}$ рассматривается её оценка — разностная характеристика $\tilde{P}^{\mathrm{ALZ}}_{A,B}$ с.с.л.р.с. подстановки Alzette, равная произведению разностных характеристик локальных разностных соотношений, приходящихся на каждой итерации на модульное сложение.

На рис. 2 дано альтернативное (более удобное для построения с.с.л.р.с.) представление итерации с номером i, $i = 1, \dots, 4$, подстановки Alzette. Здесь сложе-

ние с константой $c \in V_{32}$ не изображено, так как оно не влияет ни на построение локальных разностных соотношений, ни на значения разностных характеристик; $X_L^{(i-1)}, X_R^{(i-1)}, X_L^{(i)}, X_R^{(i)}, Z^{(i-1)} \in V_{32}$ — промежуточные разности; r_i, s_i — значения циклических сдвигов влево (они приведены в табл. 1).

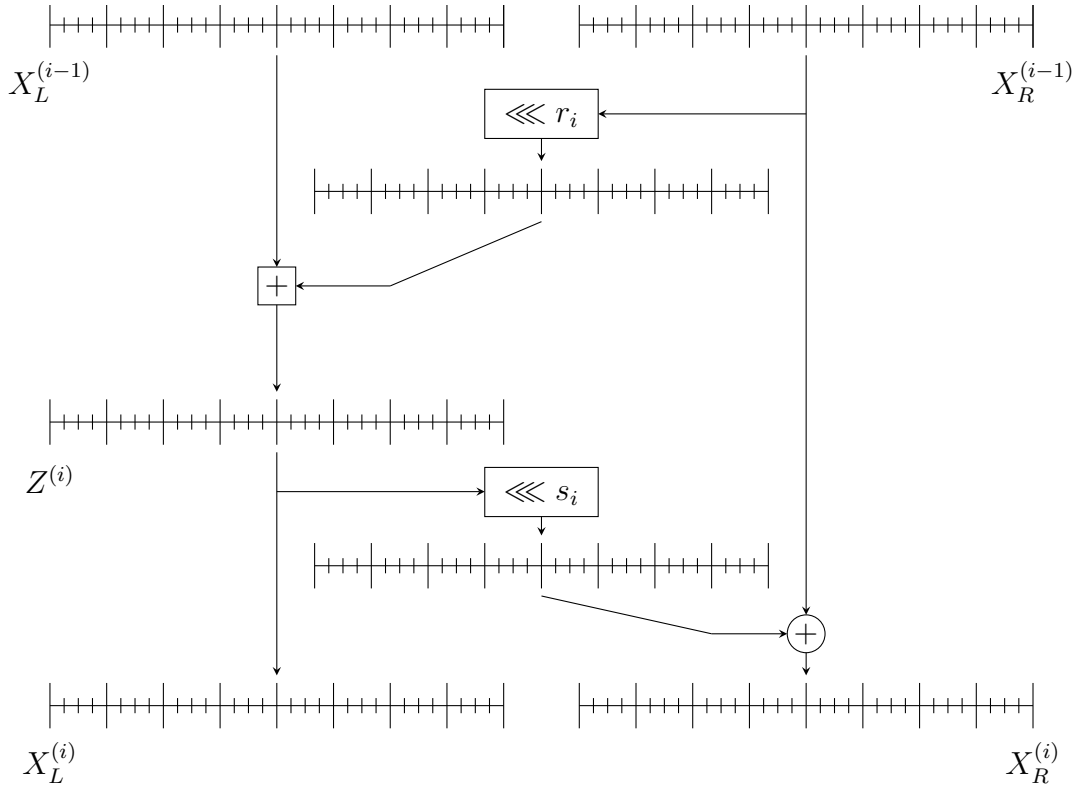

Рис. 2. Альтернативное представление i-й итерации подстановки Alzette

Таблица 1
Значения сдвигов

i	r_i	s_i
1	1	8
2	15	15
3	0	1
4	8	16

Определим, следуя [6], разностную характеристику для модульного сложения ($xdp^+(\alpha, \beta \to \gamma)$ — exclusive-or differential probability):

$$xdp^+(\alpha, \beta \to \gamma) = \mathsf{P}\left[\alpha, \beta \xrightarrow{\boxplus} \gamma\right] = \mathsf{P}\left[((x \oplus \alpha) \boxplus (y \oplus \beta)) \oplus (x \boxplus y) = \gamma\right],$$

где вероятность $\mathsf{P}[\dots]$ вычисляется в предположении случайного равновероятного распределения x, y на V_n; $x = (x_{n-1}, \dots, x_0)$, $y = (y_{n-1}, \dots, y_0)$. Отметим, что в обозначениях работы [5] $xdp^+(\alpha, \beta \to \gamma) = p^+_{\oplus\alpha, \oplus\beta, \oplus\gamma}$.

Для вычисления конкретных значений разностных характеристик локальных разностных соотношений, приходящихся в подстановке Alzette на модульное сложение, удобно воспользоваться результатами работы [6], в которой приводится представление вероятности $xdp^+(\alpha, \beta \to \gamma)$ как формального ряда над моноидом восьмеричных слов

с коэффициентами из поля действительных чисел. Для заданных $\alpha, \beta, \gamma \in V_n$ обозначим $w = (w_{n-1}, \ldots, w_0)$, где $w_i = \alpha_i \cdot 4 + \beta_i \cdot 2 + \gamma_i$, $i = 0, 1, \ldots, n-1$. Таким образом, можно задать $xdp^+(\alpha, \beta \to \gamma)$ как функцию от множества всех восьмеричных слов длины n в интервал $[0, 1] \subset \mathbb{R}$

Теорема 1 [6, теорема 2.1]. Для любого набора $(\alpha, \beta, \gamma) \in V_{3n}$

$$xdp^+(\alpha, \beta \to \gamma) = \overrightarrow{L} \cdot A_{w_{n-1}} \cdot A_{w_{n-2}} \cdot \ldots \cdot A_{w_0} \cdot C^{\downarrow},$$

где $\overrightarrow{L} = (1, 1)$; $C^{\downarrow} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$; $A_1 = A_2 = A_4 = \begin{pmatrix} 0 & 1/2 \\ 0 & 1/2 \end{pmatrix}$; $A_3 = A_5 = A_6 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \end{pmatrix}$; $A_7 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$.

Из теоремы 1, в частности, следует, что $xdp^+(\alpha, \beta \to \gamma) = 1$ только в одном из четырёх случаев, приведённых в табл. 2 (e_{n-1} — вектор из V_n, у которого единственная координата с номером $n-1$ равна единице, все остальные — нулю).

Таблица 2

Разностные соотношения сложения $\boxplus$, имеющие вероятность 1

α	0	e_{n-1}	e_{n-1}	0
β	e_{n-1}	0	e_{n-1}	0
γ	e_{n-1}	e_{n-1}	0	0

Так как для подстановки Alzette $n = 32$ (т. е. достаточно велико), то при расчёте разностных характеристик с использованием теоремы 1 для уменьшения трудоёмкости их вычисления полезно использовать следующие свойства матриц A_k, $k = 0, \ldots, 7$.

Утверждение 1. Для матриц A_k, $k = 0, \ldots, 7$:

1) $A_0^t = A_0$, $A_7^t = A_7$ для любого $t \geqslant 1$;
2) если обозначить $A_{124} = A_1 = A_2 = A_4 = \begin{pmatrix} 0 & 1/2 \\ 0 & 1/2 \end{pmatrix}$ и $A_{356} = A_3 = A_5 = A_6 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \end{pmatrix}$, то $A_{124}^t = \begin{pmatrix} 0 & 1/2^t \\ 0 & 1/2^t \end{pmatrix}$, $A_{356}^t = \begin{pmatrix} 1/2^t & 0 \\ 1/2^t & 0 \end{pmatrix}$ для любого $t \geqslant 1$;
3) $A_0^s \cdot A_7^t = A_7^t \cdot A_0^s = A_{124}^s \cdot A_0^t = A_{356}^s \cdot A_7^t = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ для любых $s, t \geqslant 1$;
4) $A_{124} \cdot A_7^t = A_{124}$, $A_{356} \cdot A_0^t = A_{356}$ для любого $t \geqslant 1$;
5) $A_0^t \cdot A_{356} = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix}$, $A_7^t \cdot A_{356} = \begin{pmatrix} 0 & 0 \\ 1/2 & 0 \end{pmatrix}$, $A_0^t \cdot A_{124} = \begin{pmatrix} 0 & 1/2 \\ 0 & 0 \end{pmatrix}$, $A_7^t \cdot A_{124} = \begin{pmatrix} 0 & 0 \\ 0 & 1/2 \end{pmatrix}$ для любого $t \geqslant 1$;
6) $A_{124}^s \cdot A_{356}^t = \begin{pmatrix} 1/2^{s+t} & 0 \\ 1/2^{s+t} & 0 \end{pmatrix}$, $A_{356}^t \cdot A_{124}^s = \begin{pmatrix} 0 & 1/2^{s+t} \\ 0 & 1/2^{s+t} \end{pmatrix}$ для любых $s, t \geqslant 1$.

Доказательство.

Пункт 1 доказывается индукцией по $i \geqslant 2$ с базисом индукции при $i = 2$: $A_0^2 = A_0$, $A_7^2 = A_7$.

Пункт 2 доказывается индукцией по $i \geqslant 2$ с базисом индукции при $i = 2$: $A_{124}^2 = \begin{pmatrix} 0 & 1/4 \\ 0 & 1/4 \end{pmatrix}$, $A_{356}^2 = \begin{pmatrix} 1/4 & 0 \\ 1/4 & 0 \end{pmatrix}$.

Пункт 3 следует из пп. 1 и 2.

Пункт 4 следует из п. 1 и легко проверяемых равенств $A_{124} \cdot A_7 = A_{124}$ и $A_{356} \cdot A_0 = A_{356}$.

Пункт 5 следует из п. 1 и легко проверяемых равенств $A_0 \cdot A_{356} = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix}$, $A_7 \cdot A_{356} = \begin{pmatrix} 0 & 0 \\ 1/2 & 0 \end{pmatrix}$, $A_0 \cdot A_{124} = \begin{pmatrix} 0 & 1/2 \\ 0 & 0 \end{pmatrix}$, $A_7 \cdot A_{124} = \begin{pmatrix} 0 & 0 \\ 0 & 1/2 \end{pmatrix}$.

Пункт 6 следует из п. 2. ■

Замечание 1. Если через A обозначить $A_{w_{n-1}} \cdot A_{w_{n-2}} \cdot \ldots \cdot A_{w_0} = A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$, то $\overrightarrow{L} \cdot A \cdot C^{\downarrow} = a_{1,1} + a_{2,1}$, следовательно, при построении разностного соотношения при заданных α и β для выполнения условия $xdp^{+}(\alpha, \beta \to \gamma) \neq 0$ необходимо выбирать вектор γ таким образом, чтобы избегать в произведении $A_{w_{n-1}} \cdot A_{w_{n-2}} \cdot \ldots \cdot A_{w_0}$ сомножителей, которые приводят к одновременному обнулению элементов в первом столбце.

Замечание 2. В работе [1] с использованием подхода из [4], реализованного на ЭВМ, найдены семь разностных соотношений для подстановки Alzette (четыре итерации) с максимальным значением разностной характеристики соответствующей с.с.л.р.с., равной 2^{-6}, а также одно разностное соотношение для подстановки Alzette на пять итераций (четыре итерации плюс дополнительная первая итерация) с максимальным значением разностной характеристики с.с.л.р.с., равной 2^{-10}. Эти разностные соотношения приведены в шестнадцатеричном виде в табл. 3.

Таблица 3

Разностные соотношения подстановки Alzette, найденные в [1]

№ п/п	Число итераций	$A \in V_{64}$	$B \in V_{64}$	$-\log_2\left(\tilde{P}^{\mathrm{ALZ}}_{A,B}\right)$
1	4	(8000010000000080)	(8040410041004041)	6
2	4	(8000010000000080)	(80C04100410040C1)	6
3	4	(0080400180400000)	(8000018081808001)	6
4	4	(0080400180400000)	(8000008080808001)	6
5	4	(A0008140000040A0)	(8000010001008001)	6
6	4	(8002010000010080)	(0101000000030101)	6
7	4	(8002010000010080)	(0301000000030301)	6
8	5	(A0008140000040A0)	(8201010200018283)	10

2. Разностная встреча посередине

Для построения с.с.л.р.с. с максимальным (или близким к нему) значением разностной характеристики для подстановки Alzette на четыре, пять и шесть итераций в соответствии с теоремой 1 и замечанием 1 предложим следующий подход, который условно назовём «разностная встреча посередине». На второй (или на третьей) итерации подстановки Alzette в качестве промежуточных разностей $X_L^{(1)}$, $X_R^{(1)} \lll r_2$, $Z^{(2)}$ (или $X_L^{(2)}$, $X_R^{(2)} \lll r_3$, $Z^{(3)}$) будем брать только векторы из табл. 1 с дальнейшим достраиванием системы локальных разностных соотношений «вверх» и «вниз», пытаясь одновременно максимизировать разностные характеристики с.с.л.р.с. на других итерациях с учётом уже сделанного выбора промежуточных разностей. При этом следует отметить, что хотя при равенстве $\alpha = \beta = \gamma = 0$ (столбец 5 табл. 1) также достигается максимальное значение локальной разностной характеристики (равное 1), такие варианты мы рассматривать не будем, так как это означает, что $X_L^{(i-1)} = X_R^{(i-1)} = 0$ при

$i = 2$ или 3, что, в свою очередь, приводит к тривиальному (нулевая разность на входе с вероятностью 1 переходит в нулевую разность на выходе) разностному соотношению для всей подстановки Alzette.

Используя подход «разностная встреча посередине» без применения ЭВМ, удалось получить все семь разностных соотношений на четыре итерации с максимальным значением разностной характеристики с.с.л.р.с., равной 2^{-6}, приведённых в [1], одно разностное соотношение на пять итераций с максимальным значением разностной характеристики с.с.л.р.с., равной 2^{-10} (также найденное в [1]), и одно разностное соотношение на шесть итераций с максимальным значением разностной характеристики с.с.л.р.с., равной 2^{-18}. Отметим, что в [1] только упоминается, что максимальное значение разностной характеристики с.с.л.р.с. на шести итерациях равно 2^{-18}, но ни одной с.с.л.р.с., на которой оно достигается, не приведено. При этом для меньшего числа итераций — четыре и пять — все соответствующие разностные соотношения приведены. Вероятно, для шести итераций авторы [1] получили не точное значение максимальной разностной характеристики с.с.л.р.с., а лишь её оценку сверху, потому что, начиная с семи итераций, в работе [1] прямо указывается, что предъявленные максимальные значения разностных характеристик (также без указания соответствующих им разностных соотношений) являются лишь оценками сверху.

3. Разностная встреча посередине для шести итераций подстановки Alzette

Подробно распишем построение и получение значений разностных характеристик с.с.л.р.с. для разностного соотношения на шесть итераций. Для оставшихся разностных соотношений на четыре итерации приведём с.с.л.р.с. вида

$$X_L^{(0)}, X_R^{(0)}, X_L^{(1)}, X_R^{(1)}, X_L^{(2)}, X_R^{(2)}, X_L^{(3)}, X_R^{(3)}, X_L^{(4)}, X_R^{(4)},$$

по которым с использованием теоремы 1 можно проверить равенство разностной характеристики максимальному значению 2^{-6}.

Начнём с разностного соотношения на шесть итераций. На рис. 3–8 во всех двоичных векторах длины 32 для простоты отмечены (символом «•») только единичные координаты. С учётом замечания 1 значение $xdp^+(\alpha, \beta \to \gamma)$ будем характеризовать в виде произведения 32 матриц (без дополнительного умножения на $\overrightarrow{L}$ и $C^{\downarrow}$), помня, что для получения значения $xdp^+(\alpha, \beta \to \gamma)$ необходимо сложить строки результирующей матрицы и выбрать старшую координату.

На рис. 3 и 8 над $X_L^{(0)}, X_R^{(0)}$ и $X_L^{(6)}, X_R^{(6)}$ записаны их шестнадцатеричные представления. Из построения с.с.л.р.с. на шесть итераций видно, что она содержит в себе с.с.л.р.с. на четыре итерации с максимальным значением разностной характеристики 2^{-6} (№ 5 из табл. 3), а также с.с.л.р.с. на пять итераций (№ 8 из табл. 3), значения $X_L^{(4)}, X_R^{(4)}$ и $X_L^{(5)}, X_R^{(5)}$ для которого записаны на рис. 6 и 7 в шестнадцатеричном виде.

Таким образом, для разностного соотношения на шесть итераций получаем, что $P^{\mathrm{ALZ}}_{(\mathrm{A0008140000040A0,434081024080A323})} = \frac{1}{16}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{16}\cdot\frac{1}{256} = 2^{-18}$, а для разностного соотношения на пять и четыре итерации $P^{\mathrm{ALZ}}_{(\mathrm{A0008140000040A0,8201010200018283})} = \frac{1}{16}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{16} = 2^{-10}$, $P^{\mathrm{ALZ}}_{(\mathrm{A0008140000040A0,8000010001008001})} = \frac{1}{16}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2} = 2^{-6}$ соответственно.

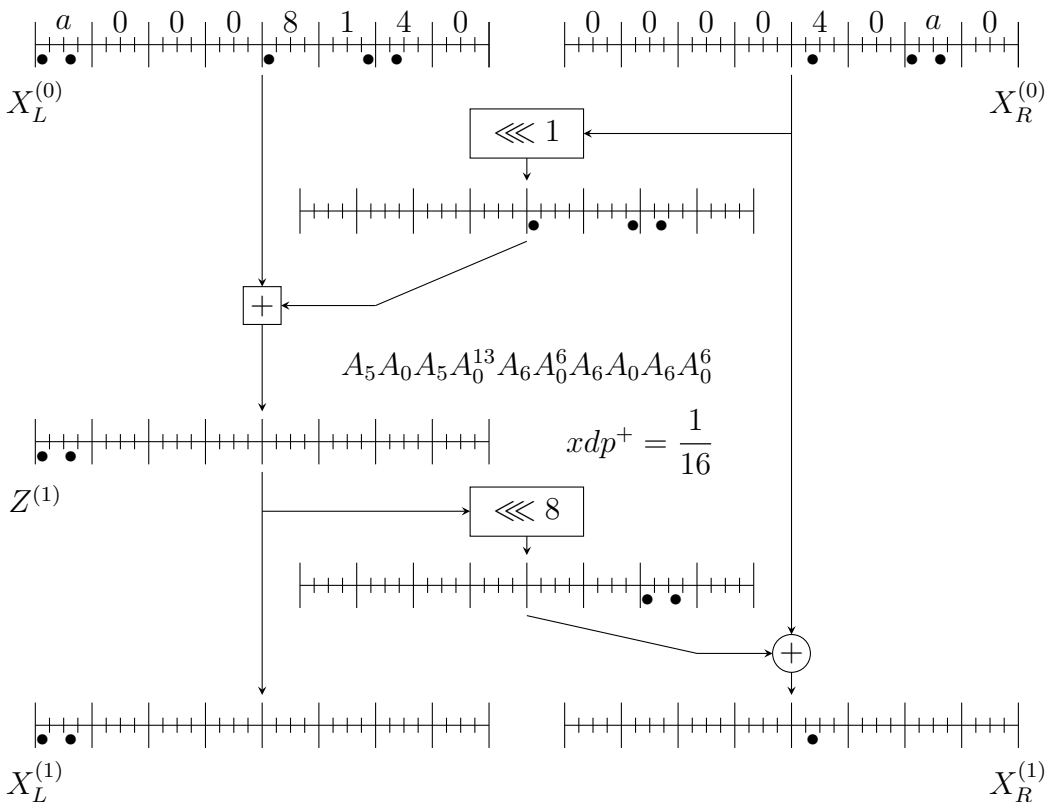

Рис. 3. Итерация 1

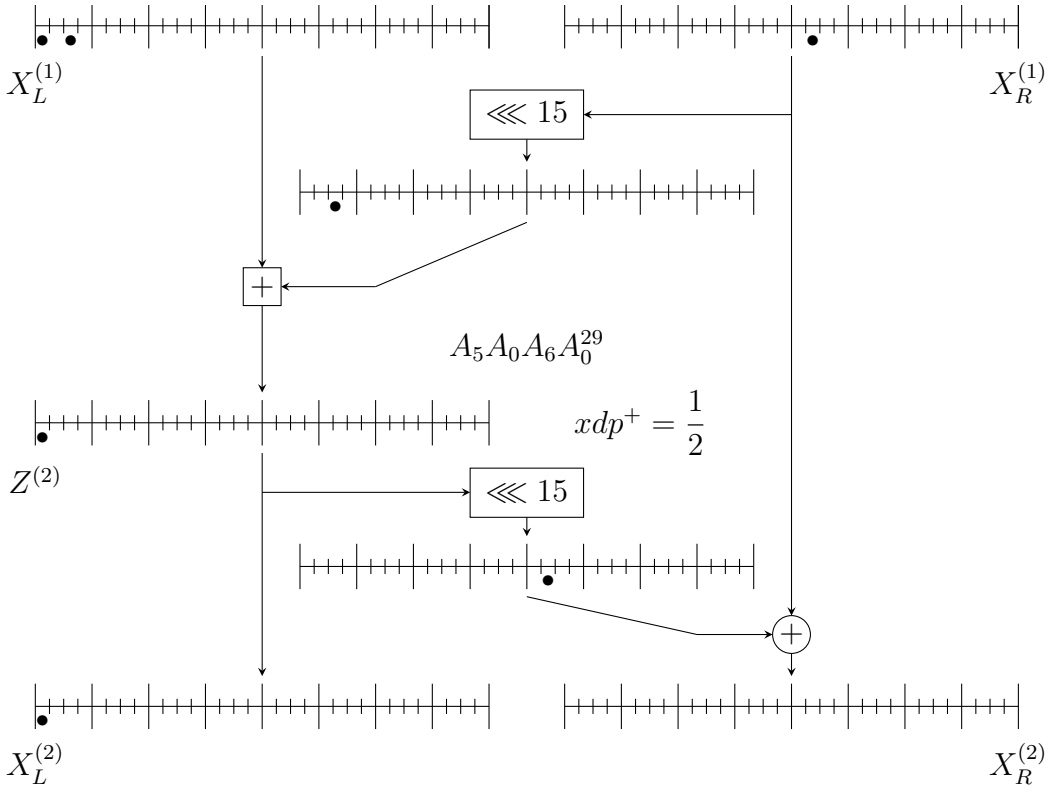

Рис. 4. Итерация 2

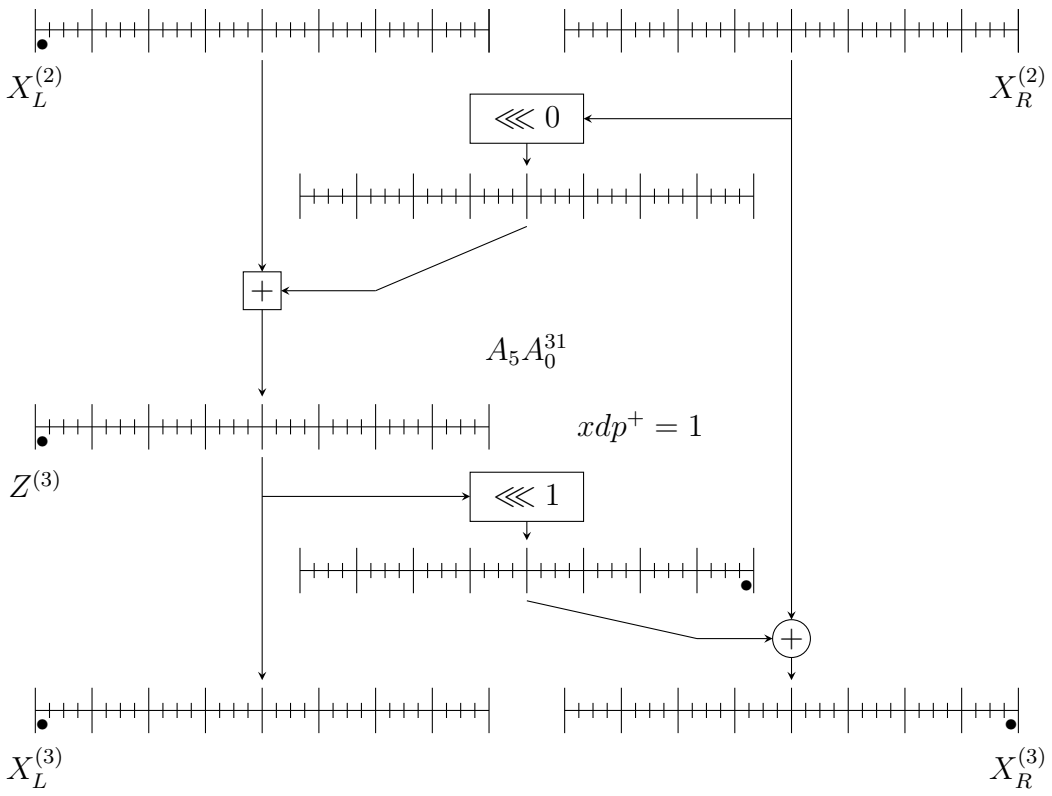

Рис. 5. Итерация 3

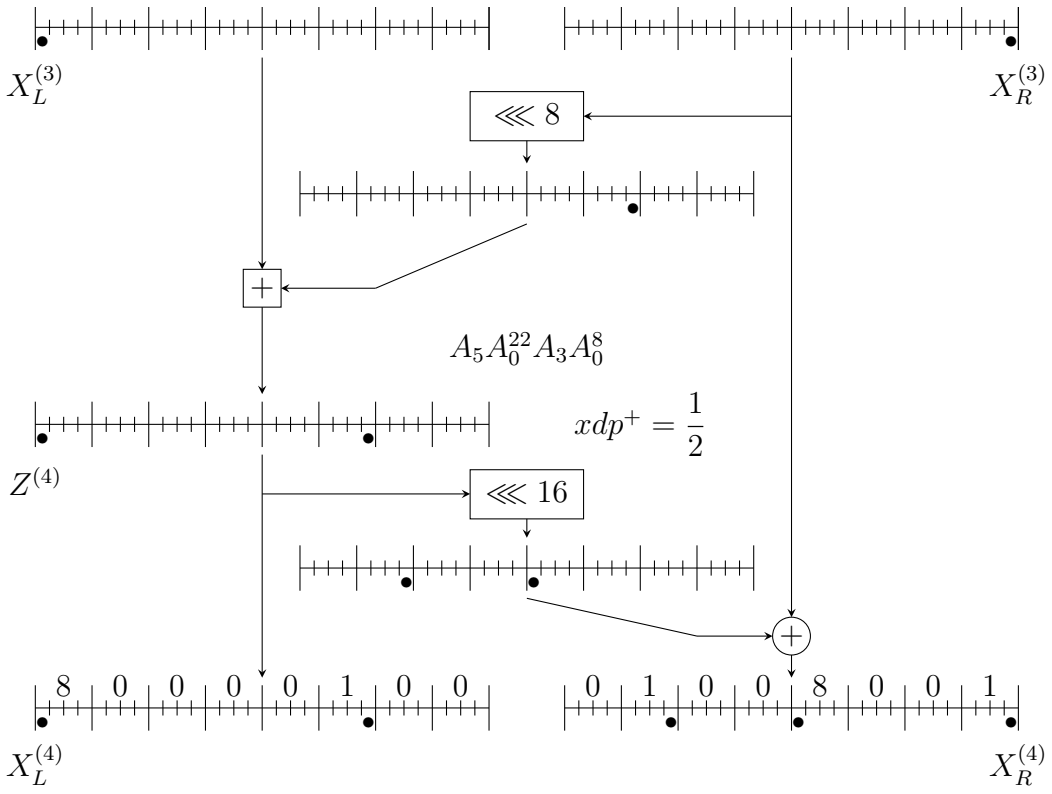

Рис. 6. Итерация 4

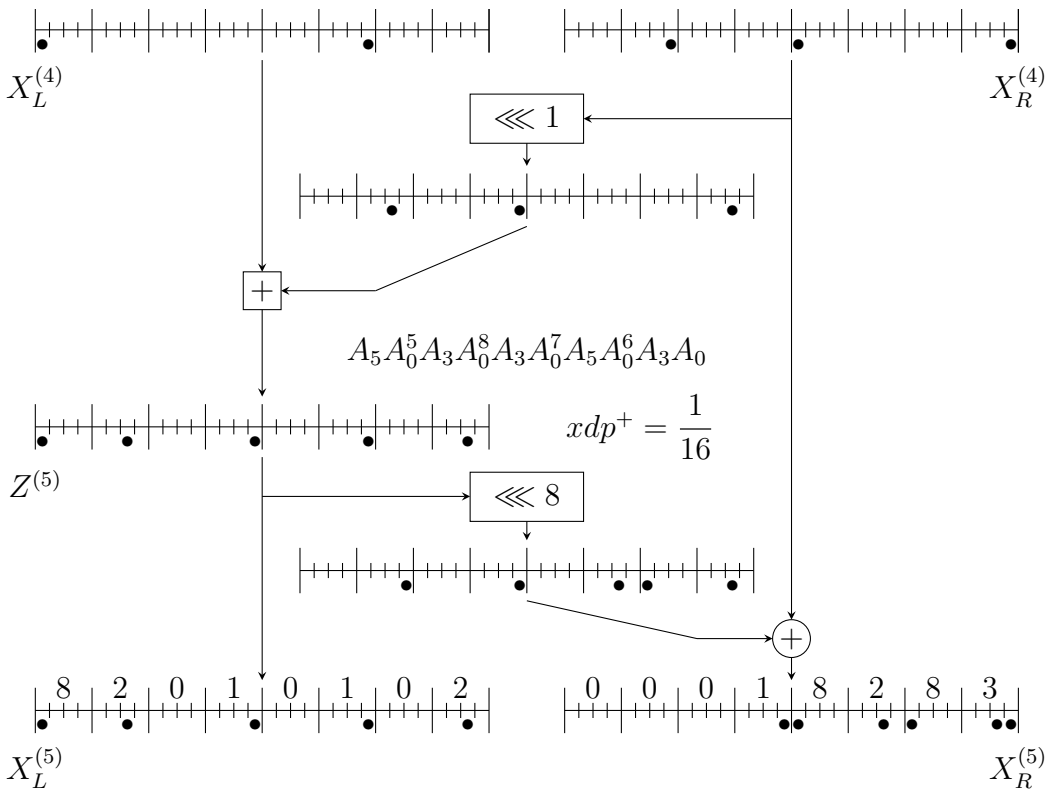

Рис. 7. Итерация 5

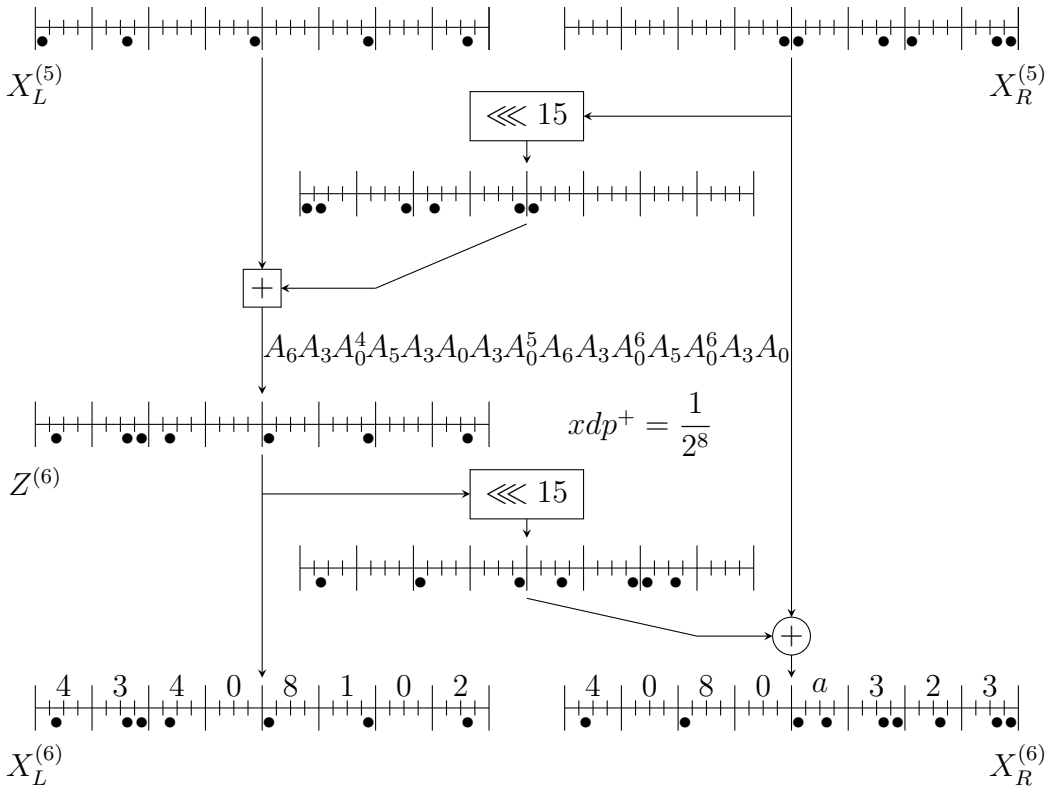

Рис. 8. Итерация 6

В табл. 4 приведены в шестнадцатеричном виде с.с.л.р.с. для оставшихся разностных соотношений с максимальным значением разностной характеристики (№ 1–4, 6, 7 из табл. 3).

Таблица 4

Разностные соотношения подстановки Alzette на четыре итерации

Соотношение № 1	Соотношение № 2
$\left(X_L^{(0)}, X_R^{(0)}\right) = (80000100, 00000080)$ $Z^{(1)} = X_L^{(1)} = (80000000), X_R^{(1)} = (00000000)$ $Z^{(2)} = X_L^{(2)} = (80000000), X_R^{(2)} = (00004000)$ $Z^{(3)} = X_L^{(3)} = (80004000), X_R^{(3)} = (0000\mathrm{C}001)$ $Z^{(4)} = X_L^{(4)} = (80404100), X_R^{(4)} = (41004041)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{16} = 2^{-6}$	$\left(X_L^{(0)}, X_R^{(0)}\right) = (80000100, 00000080)$ $Z^{(1)} = X_L^{(1)} = (80000000), X_R^{(1)} = (00000000)$ $Z^{(2)} = X_L^{(2)} = (80000000), X_R^{(2)} = (00004000)$ $Z^{(3)} = X_L^{(3)} = (80004000), X_R^{(3)} = (0000\mathrm{C}001)$ $Z^{(4)} = X_L^{(4)} = (80\mathrm{C}04100), X_R^{(4)} = (410040\mathrm{C}1)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{16} = 2^{-6}$
Соотношение № 3	Соотношение № 4
$\left(X_L^{(0)}, X_R^{(0)}\right) = (00804001, 80400000)$ $Z^{(1)} = X_L^{(1)} = (00004000), X_R^{(1)} = (80000000)$ $Z^{(2)} = X_L^{(2)} = (00000000), X_R^{(2)} = (80000000)$ $Z^{(3)} = X_L^{(3)} = (80000000), X_R^{(3)} = (80000001)$ $Z^{(4)} = X_L^{(4)} = (80000180), X_R^{(4)} = (81808001)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{8} \cdot \frac{1}{2} \cdot 1 \cdot \frac{1}{4} = 2^{-6}$	$\left(X_L^{(0)}, X_R^{(0)}\right) = (00804001, 80400000)$ $Z^{(1)} = X_L^{(1)} = (00004000), X_R^{(1)} = (80000000)$ $Z^{(2)} = X_L^{(2)} = (00000000), X_R^{(2)} = (80000000)$ $Z^{(3)} = X_L^{(3)} = (80000000), X_R^{(3)} = (80000001)$ $Z^{(4)} = X_L^{(4)} = (80000080), X_R^{(4)} = (80808001)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{8} \cdot \frac{1}{2} \cdot 1 \cdot \frac{1}{4} = 2^{-6}$
Соотношение № 5	Соотношение № 6
$\left(X_L^{(0)}, X_R^{(0)}\right) = (80020100, 00010080)$ $Z^{(1)} = X_L^{(1)} = (80000000), X_R^{(1)} = (00010000)$ $Z^{(2)} = X_L^{(2)} = (00000000), X_R^{(2)} = (00010000)$ $Z^{(3)} = X_L^{(3)} = (00010000), X_R^{(3)} = (00030000)$ $Z^{(4)} = X_L^{(4)} = (01010000), X_R^{(4)} = (00030101)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{4} \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{8} = 2^{-6}$	$\left(X_L^{(0)}, X_R^{(0)}\right) = (80020100, 00010080)$ $Z^{(1)} = X_L^{(1)} = (80000000), X_R^{(1)} = (00010000)$ $Z^{(2)} = X_L^{(2)} = (00000000), X_R^{(2)} = (00010000)$ $Z^{(3)} = X_L^{(3)} = (00010000), X_R^{(3)} = (00030000)$ $Z^{(4)} = X_L^{(4)} = (03010000), X_R^{(4)} = (00030301)$ $\tilde{P}^{\mathrm{ALZ}}_{\left(x_L^{(0)}, x_R^{(0)}\right), \left(x_L^{(4)}, x_R^{(4)}\right)} = \frac{1}{4} \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{8} = 2^{-6}$

4. Разностная встреча посередине для семи и восьми итераций подстановки Alzette

С использованием ЭВМ проведён поиск с.с.л.р.с. для семи (4 плюс 3 первых итерации) и восьми (4 плюс 4) итераций подстановки Alzette. Непосредственно поиск проводился для восьми итераций, а системы для семи итераций строились из найденных систем. В [1] для такого числа итераций приведены только оценки сверху на минус двоичный логарифм максимума разностной характеристики — соответственно 24 и 32.

Алгоритм является разновидностью описанной «разностной встречи посередине». Предварительно были сформированы наборы разностей для сложения $\boxplus$, имеющие вероятности 1, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}. Алгоритм состоит из трёх этапов. Результатами первого и второго этапов являются частичные системы для итераций с третьей по шестую и с третьей по восьмую соответственно. На третьем этапе системы, прошедшие первые два этапа, достраиваются до полных систем для итераций с первой по восьмую.

Первый этап. Фиксируем значения $p_3, p_6 \in \{2^{-k} : k = 0, \ldots, 4\}$. На итерации 3 перебираем все разности, имеющие вероятность p_3, а на итерации 6 — все разности, имеющие вероятность p_6. При их фиксации разности на итерациях 4 и 5 вычисляются однозначно. Таким образом получаем с.с.л.р.с. на итерациях 3–6. Проверяем, что вероятность этой системы больше $p_{\max}$. Если условие не выполняется, то выбираем

следующую пару разностей на третьей и шестой итерациях. Системы, для которых условие выполняется, переходят на второй этап.

В т о р о й э т а п. Для каждой из систем, полученных на первом этапе, перебираем все возможные выходные разности сложения $\boxplus$ на седьмой итерации (входные разности определяются по выходным разностям шестой итерации). Полученные системы на итерации 3–7, разностная характеристика которых превышает $p_{\max}$, достраиваются до системы на восемь итераций так, чтобы максимизировать вероятность системы на итерациях 3–8. Полученные системы на итерации 3–8, разностная характеристика которых превышает $p_{\max}$, переходят на третий этап.

Т р е т и й э т а п аналогичен второму, только перебираются все возможные разности на второй итерации и достраивается (исходя из максимизации разностной характеристики результирующей системы) разность на первой итерации. Построенные таким образом системы, имеющие вероятность больше $p_{\max}$, являются результатом работы алгоритма.

В качестве $p_{\max}$ была выбрана полученная в [1] граница на разностную характеристику системы локальных разностных соотношений на девять итераций, равная 2^{-36}. В табл. 5 приведены сочетания вероятностей p_3, p_6 опробованных пар разностей.

Т а б л и ц а 5

$-\log_2(p_3)\,(-\log_2(p_6))$	$-\log_2(p_6)\,(-\log_2(p_3))$
0 и 1	0 и 1
0 и 1	2
0 и 1	3
0 и 1	4
2	0 и 1
2	2
2	3
2	4

Системы на семь итераций строятся из систем на восемь итераций следующим образом: отбрасывается локальное разностное соотношение восьмой итерации, а соотношение на седьмой итерации заменяется соотношением с максимальной вероятностью выполнения.

Всего получено 11 с.с.л.р.с. на восемь итераций, имеющих разностную характеристику 2^{-35}. Из них построено 6 с.с.л.р.с. на семь итераций, имеющих разностную характеристику 2^{-27}. Полученные соотношения приведены в табл. 6. В табл. 7 в качестве примера приведена одна из полученных систем на восемь итераций.

Таблица 6

Разностные соотношения подстановки Alzette на семь и восемь итераций

№ п/п	Число итераций	$A \in V_{64}$	$B \in V_{64}$	$-\log_2 \tilde{P}^{\mathrm{ALZ}}_{A,B}$
1	7	(00A1508020508040)	(10102080507001A0)	27
2	7	(0021108060108040)	(10102080507001A0)	27
3	7	(00A1508020508040)	(10102080507001A0)	27
4	7	(0021108060108040)	(10102080507001A0)	27
5	7	(210002410080C121)	(410020200340A0E0)	27
6	7	(210002410080C121)	(410020200340A0E0)	27
7	8	(A080410180C000A0)	(81804140C1418141)	35
8	8	(210002410080C121)	(002040204060A0C0)	35
9	8	(210002410080C121)	(00A0002000602040)	35
10	8	(2101000001810020)	(0181018081010080)	35
11	8	(00A1508020508040)	(2010001050602030)	35
12	8	(00A1508020508040)	(0010005010200030)	35
13	8	(0021108060108040)	(0010005010200030)	35
14	8	(4200028501008142)	(0040804080C14180)	35
15	8	(4200028501008142)	(0140004000C04080)	35
16	8	(80001081801000C0)	(80C000C080404080)	35
17	8	(0021108060108040)	(2010001050602030)	35

Таблица 7

Пример системы локальных разностных соотношений на восемь итераций

№ итерации i	$X^{(i-1)} = \left(X_L^{(i-1)}, X_R^{(i-1)}\right)$	$X^{(i-1)} = \left(X_L^{(i)}, X_R^{(i)}\right)$	$-\log_2 P\left(X^{(i-1)} \to X^{(i)}\right)$
1	(21000241, 0080C121)	(20008001, 0000C001)	8
2	(20008001, 0000C001)	(80000001, 00000001)	4
3	(80000001, 00000001)	(80000000, 00000000)	1
4	(80000000, 00000000)	(80000000, 00008000)	0
5	(80000000, 00008000)	(80010000, 01008080)	1
6	(80010000, 01008080)	(C0410080, 8140E0A0)	4
7	(C0410080, 8140E0A0)	(40802020, 0040A0E0)	10
8	(40802020, 0040A0E0)	(00204020, 4060A0C0)	7

5. Оценка трудоёмкости построения разностных соотношений для подстановки Alzette

Трудоёмкость предложенного подхода «разностная встреча посередине» для четырёх итераций подстановки Alzette равна $3^2 \cdot 4 = 36$ вычислений (в соответствии с теоремой 1) вероятностей локальных разностных соотношений на двоичных векторах длины 32 (существенно менее трудоёмкими операциями циклического сдвига и сложения по $\bmod 2$ двоичных векторов длины 32 мы пренебрегаем), для пяти и шести итераций указанная трудоёмкость равна $3^3 \cdot 5 = 135$ и $3^4 \cdot 6 = 486$ операций соответственно (причём это максимальная трудоёмкость — без учёта уже известных результатов по локальным разностным соотношениям для четырёх итераций).

Трудоёмкость алгоритма для восьми итераций в основном определяется трудоёмкостью первого этапа. Для каждой пары опробуемых локальных разностных соотношений необходимо вычислить вероятности локальных разностных соотношений для итераций 4 и 5. Число опробуемых пар рассчитаем при помощи формулы из [6, теорема 2.3]. Если обозначить через $D_n(j)$ число разностных соотношений разрядности n,

имеющих вероятность 2^{-j}, $j \in \{0, \ldots, n\}$, то

$$D_n(j) = 4 \cdot 6^j \binom{n-1}{j}.$$

Тогда общее число опробуемых пар равно

$$\sum_{p_3, p_6} D_{32}(-\log_2(p_3)) D_{32}(-\log_2(p_6)) \approx 2{,}26 \cdot 10^{13}.$$

С учётом двух операций для каждой разности и этапов 2 и 3 общую трудоёмкость нахождения с.с.л.р.с. на восемь итераций можно оценить сверху величиной $5 \cdot 10^{13}$.

Системы на семь итераций строятся из систем на восемь итераций фактически вручную, поэтому в качестве оценки трудоёмкости их построения можно использовать оценку для восьми итераций.

Для сравнения оценим трудоёмкость алгоритма поиска/оценки максимальных разностных характеристик, которым пользовались авторы работы [1]. В [1] не описан алгоритм и не оценена его трудоёмкость, поэтому все выводы сделаны на основе анализа программной реализации [8].

Суть алгоритма состоит в следующем. Последовательно, от младшего разряда к старшему, опробуются локальные разностные соотношения первой итерации и для каждого «частичного» соотношения вычисляется его вероятность. Далее эта вероятность умножается на разностную характеристику наилучшей с.с.л.р.с. на оставшийся «хвост» из итераций 2, 3 и т. д. — таким образом получается оценка сверху на характеристику наилучшей системы локальных разностных соотношений на заданное число итераций, имеющей опробуемое в данный момент начало. Если эта оценка меньше заданной границы p_b, то мы опробуем следующее значение текущего разряда «частичного» локального соотношения или возвращаемся на один разряд назад. Если же оценка не меньше границы p_b, то переходим к следующем разряду. После получения полного локального соотношения на первой итерации повторяем аналогичный процесс на второй итерации с той разницей, что в качестве оценки характеристики текущей опробуемой системы используется произведение вероятности (уже полной) локального соотношения на первой итерации, вероятность «частичного» соотношения на второй итерации и разностная характеристика наилучшей системы на «хвост» из итераций 3, 4 и т. д. Ещё одно отличие: на первой итерации перебираются все три вектора α, β, γ, а на последующих — только γ. В конце концов мы или получаем систему на заданное число итераций с разностной характеристикой не меньше p_b, или убеждаемся, что систем с такой характеристикой на заданное число итераций нет. В последнем случае уменьшаем p_b в 2 раза и повторяем весь алгоритм. В качестве начального значения p_b для r итераций используется значение характеристики наилучшей системы на $(r-1)$ итераций.

Описанный алгоритм предназначен для обоснования криптографических свойств подстановки Alzette при криптографическом синтезе, то есть в случае нахождения системы разностных соотношений с разностной характеристикой p_b утверждается, что систем с бóльшей разностной характеристикой (для данного числа итераций) не существует. Алгоритм разностной встречи посередине ориентирован на криптографический анализ, то есть на нахождение системы разностных соотношений с как можно большей разностной характеристикой за как можно меньшее число операций. Поэтому чтобы сравнить трудоёмкости этих двух алгоритмов, будем рассчитывать трудоёмкость алгоритма из [1], исходя из поиска конкретных систем разностных соотношений.

Оценим снизу трудоёмкость алгоритма из работы [1] следующим образом. Зафиксируем число итераций r. Пусть вероятность искомой системы разностных соотношений на r-й итераций составляет $p^{(1)}$, а наибольшая вероятность разностного соотношения на $(r-1)$ итераций, начиная со второй, — $p^{(2)}$. Это означает, что прежде чем мы дойдём до искомого разностного соотношения, имеющего вероятность $p^{(1)}$, мы должны будем проверить $k = -\log_2\left(p^{(1)}\right) + \log_2\left(p^{(2)}\right)$ ложных границ $p_b = p^{(2)}$, $p^{(2)}/2$, $p^{(2)}/4$, $\ldots$, $p^{(2)}/2^{k-1}$. Для каждой из этих ложных границ мы будем как минимум перебирать все разности первой итерации, имеющие «разрешённую» вероятность $p_1 \in \left\{1, 1/2, \ldots, p^{(1)}/p^{(2)}\right\}$, и для каждой из них вычислять эту вероятность 32 раза (при добавлении каждого разряда разности). Таким образом, будет произведено не менее $S(k)$ вычислений вероятности разностного соотношения, где

$$S(k) = 32 \sum_{i=0}^{k-1} \sum_{j=0}^{i} D_{32}(j). \tag{1}$$

Как видно из формулы (1), величина $S(k)$ зависит только от k, по сути — от отношения $p^{(1)}/p^{(2)}$, поэтому если при разных r значения k одинаковые, то и величины $S(k)$ тоже одинаковы. В этом случае чтобы подчеркнуть, что $S(k)$ больше при большем r, будем учитывать, что значение $p^{(2)}$ должно быть определено по аналогичному алгоритму с использованием значения $p^{(3)}$ — наибольшей вероятности разностного соотношения на $(r-2)$ итераций, начиная с третьей. Соответствующая трудоёмкость будет не меньше чем $S(k')$, где $k' = -\log_2\left(p^{(2)}\right) + \log_2\left(p^{(3)}\right)$.

В табл. 8 приведены значения $-\log_2\left(p^{(i)}\right)$, $i = 1, 2, 3$. В ней:

— значения $p^{(1)}$ — разностные характеристики конкретных найденных систем разностных соотношений;
— значения $p^{(2)}$ для четырёх, пяти и шести итераций и $p^{(3)}$ для пяти и восьми итераций взяты из [1, табл. 2] (см. табл. 3 и 6);
— в качестве $p^{(2)}$ для восьми итераций взято значение границы из [1, табл. 2];
— значения k — разности значений второй и третьей колонок;
— значения k' — разности значений третьей и четвёртой колонок;
— значения $p^{(3)}$ и k' приведены только для такого числа итераций, у которых k совпадает с k для предыдущего числа.

Таблица 8

Число итераций	$-\log_2(p^{(1)})$	$-\log_2(p^{(2)})$	$-\log_2(p^{(3)})$	k	k'
4	6	2	–	4	–
5	10	6	2	4	4
6	18	10	–	8	–
7	27	17	–	10	–
8	35	25	18	10	7

Окончательно получаем: если обозначить через S_r оценку снизу трудоёмкости алгоритма из работы [1], то имеет место следующее выражение:

$$S_r = \begin{cases} S(k), & r \in \{4, 6, 7\}, \\ S(k) + S(k'), & r \in \{5, 8\}, \end{cases}$$

где k, k' зависят от r и берутся из табл. 8. Численные значения S_r и оценки сверху на трудоёмкость разностной встречи посередине приведены в табл. 9.

Таблица 9

Число итераций r	S_r	Трудоёмкость РВП (оценка сверху)
4	$1{,}29 \cdot 10^8$	36
5	$2 \cdot 1{,}29 \cdot 10^8$	135
6	$1{,}03 \cdot 10^{14}$	486
7	$2{,}97 \cdot 10^{16}$	$5 \cdot 10^{13}$
8	$2{,}97 \cdot 10^{16} + 4{,}75 \cdot 10^{12}$	$5 \cdot 10^{13}$

Заключение

В работе с использованием предложенного подхода «разностная встреча посередине» для подстановки Alzette с четырьмя и пятью итерациями получены все согласованные системы локальных разностных соотношений, соответствующие максимальному значению разностной характеристики этой подстановки. Для подстановки Alzette с шестью итерациями получена одна система с максимальной разностной характеристикой. Разностные соотношения для четырёх и пяти итераций, соответствующие данным согласованным системам (без указания самих этих систем), получены ранее разработчиками подстановки Alzette с использованием вычислений на ЭВМ.

Построены согласованные системы локальных разностных соотношений для семи и восьми итераций подстановки Alzette, разностные характеристики которых лишь незначительно отличаются от оценок сверху, полученных в [1]. Показано, что в задаче построения разностного соотношения с как можно бóльшим значением разностной характеристики за как можно меньшее число операций разностная встреча посередине обладает меньшей трудоёмкостью, чем алгоритм, использованный в [1] для обоснования криптографических характеристик подстановки Alzette.

Подход «разностная встреча посередине», использующий математический аппарат формальных рядов с коэффициентами из поля действительных чисел, позволяет относительно просто находить согласованные системы локальных разностных соотношений, имеющие максимальное значение разностной характеристики. Полученные результаты могут быть использованы, например, при оценке характеристик криптографических примитивов, построенных с использованием подстановки Alzette.

ЛИТЕРАТУРА

1. *Beierle C., Biryukov A., Cardoso dos Santos L., et al.* Alzette: A 64-bit ARX-box. Cryptology ePrint Archive. Report 2019/1378. 2019. `https://eprint.iacr.org/2019/1378`.
2. *Canteaut A., Duval S., Leurent G., et al.* Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/saturnin-spec-round2.pdf`.
3. *Dinu D., Perrin L., Udovenko A., et al.* Design strategies for ARX with provable bounds: Sparx and LAX // LNCS. 2016. V. 10031. P. 484–513.
4. *Biryukov A., Velichkov V., and Corre Y. L.* Automatic search for the best trails in ARX: Application to block cipher Speck // LNCS. 2016. V. 9783. P. 289–310.
5. *Малышев Ф. М.* Вероятностные характеристики разностных и линейных соотношений для неоднородной линейной среды // Математические вопросы криптографии. 2019. Т. 10. Вып. 1. С. 41–72.
6. *Wallèn J.* On the differential and linear properties of addition. Research Report A84. Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, 2003. 58 p.

7. *Малышев Ф. М., Трифонов Д. И.* Рассеивающие свойства XSLP-шифров // Математические вопросы криптографии. 2016. Т. 7. Вып. 3. С. 47–60.
8. https://github.com/cryptolu/sparkle.

REFERENCES

1. *Beierle C., Biryukov A., Cardoso dos Santos L., et al.* Alzette: A 64-bit ARX-box. Cryptology ePrint Archiv, Report 2019/1378, 2019. https://eprint.iacr.org/2019/1378.
2. *Canteaut A., Duval S., Leurent G., et al.* Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/saturnin-spec-round2.pdf.
3. *Dinu D., Perrin L., Udovenko A., et al.* Design strategies for ARX with provable bounds: Sparx and LAX. LNCS, 2016, vol. 10031, pp. 484–513.
4. *Biryukov A., Velichkov V., and Corre Y. L.* Automatic search for the best trails in ARX: Application to block cipher Speck. LNCS, 2016, vol. 9783, pp. 289–310.
5. *Malyshev F. M.* Veroyatnostnye kharakteristiki raznostnykh i lineynykh sootnosheniy dlya neodnorodnoy lineynoy sredy [Probabilistic characteristics of differential and linear relations for nonhomogeneous linear medium]. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, iss. 1, pp. 41–72. (in Russian)
6. *Wallèn J.* On the differential and linear properties of addition. Research Report A84, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, 2003. 58 p.
7. *Malyshev F. M. and Trifonov D. I.* Rasseivayushchie svoystva XSLP-shifrov [Diffusion properties of XSLP-ciphers]. Matematicheskie Voprosy Kriptografii, 2016, vol. 7, iss. 3, pp. 47–60. (in Russian)
8. https://github.com/cryptolu/sparkle.

УДК 519.7 DOI 10.17223/20710410/58/6

ОБ ОДНОМ МЕТОДЕ ПОСТРОЕНИЯ ОДНОРОДНЫХ ПЛОСКОСТНЫХ АППРОКСИМАЦИЙ ФИЛЬТРУЮЩЕГО ГЕНЕРАТОРА

Л. А. Кущинская

Московский государственный университет им. М. В. Ломоносова, г. Москва, Россия

E-mail: lyudmila.kuschinskaja@yandex.ru

Плоскостные аппроксимации фильтрующих генераторов могут быть использованы для восстановления его начального состояния по отрезку выходной последовательности. Представлены результаты исследования одного метода построения плоскостных аппроксимаций специального вида.

Ключевые слова: *криптоанализ, восстановление ключа, фильтрующий генератор, плоскостная аппроксимация.*

THE METHOD FOR CONSTRUCTING UNIFORM PLANAR APPROXIMATIONS OF THE FILTER GENERATOR

L. A. Kuschinskaya

Lomonosov Moscow State University, Moscow, Russia

We study the possibility of constructing an approximation of filter generator to restore initial state $u^* \in V_n$ from its output sequence $z_i = f(A^i(u^*)) \in \{0,1\}$, $i = 0, \ldots, N-1$, where $A : V_n \to V_n$ is non-degenerate linear mapping, f is balanced Boolean function. The triple $(m, L_0, \mathbb{T})$ is a key element in the construction of the approximation, where $m \in \mathbb{N}$, L_0 is coset of the space V_n, $\mathbb{T} = (t_0, t_1, \ldots, t_m)$, $t_0 = 0$, $t_0 \leqslant t_1 < \ldots < t_m$. Let $(m, L_0, \mathbb{T})$ be a triple and for $b_1, \ldots, b_m \in \{0,1\}$ the probability that $f(v) = b_i$ is greater than $1/2$ for a random equiprobable choice of a vector v from $L_i = A^{t_i - t_{i-1}}(L_{i-1})$, $i = 1, \ldots, m$. Then a finite number of such triples with pairwise distinct sets L_0 makes it possible to restore the key with a complexity that is much less than the complexity of enumerating keys in some cases. In this paper, we study the possibility of constructing approximations of a special form, where all cosets L_0 have the same dimension, their union is equal to V_n, and the values of m are the same for all described triples. Expressions for the optimal values of the parameters k and δ are obtained for some enumeration method for constructing the approximations. It is shown that for $k = \left\lceil \log_2 \left((Q - \sqrt{Q^2 - \pi_0 2^{n+2}})/2\pi_0 \right) \right\rceil$ and $\delta \approx \lceil t_0 \sqrt{\Omega} \rceil$ it is possible to achieve the minimum length of the generator output sequence required to construct such approximations for a given value of the upper complexity Q and lower reliability π_0 of the initial state recovery method, where $\Omega = 2^k$, $t_0 \approx 1.19061$.

Keywords: *cryptanalysis, key recovery, filter generator, planar approximation.*

Введение

Классическими структурными элементами потоковых шифров являются фильтрующий и комбинирующий генераторы. Эти генераторы нашли широкое применение в современных потоковых шифрах, используемых на практике, например E0 [1],

A5/1 [2], SNOW 2.0 [3], Grain [4]. Анализ криптографических свойств подобных конструкций представляет существенный интерес [5 – 9].

В 2017 г. в работе [10] предложен метод восстановления ключа фильтрующего генератора, основанный на понятии плоскостной аппроксимации функции усложнения. Свойства используемой плоскостной аппроксимации напрямую влияют на численные характеристики метода и позволяют в ряде случаев восстанавливать ключ с трудоёмкостью, существенно меньшей трудоёмкости полного опробования ключей. В [10] описан только этап восстановления ключа в предположении, что необходимая для этого плоскостная аппроксимация уже построена. Вопрос построения плоскостных аппроксимаций ранее не исследовался.

В данной работе исследуется вопрос построения плоскостных аппроксимаций специального вида, которые названы *однородными*. Плоскостная аппроксимация определяет три основных численных характеристики метода восстановления ключа: трудоёмкость, надёжность и необходимое количество известных битов выходной последовательности генератора. Рассматривается задача построения однородной плоскостной аппроксимации, при которой метод восстановления ключа достигает заданных значений трудоёмкости и надёжности при минимально возможном требуемом количестве известных битов выходной последовательности. Предложена математическая модель работы переборного метода решения задачи, в рамках которой получены оптимальные значения его параметров.

1. Основные определения и обозначения

Пусть $\mathbb{F}_2$ — поле из 2 элементов, $V_n = \mathbb{F}_2^n$ — линейное пространство размерности n над полем $\mathbb{F}_2$. *Носителем вектора* $x = (x_0, \ldots, x_{n-1}) \in V_n$ называется множество $\operatorname{supp}(x) = \{i \in \{0, \ldots, n-1\} : x_i = 1\}$. Тот факт, что $L \subseteq V_n$ является подпространством пространства V_n, будем обозначать так: $L < V_n$; линейную оболочку векторов $v^{(1)}, \ldots, v^{(k)}$ из V_n обозначим $L(v^{(1)}, \ldots, v^{(k)})$ [11]. *Плоскостью* в пространстве V_n будем называть смежный класс по подпространству этого пространства, а её размерностью — размерность этого подпространства.

Булевой функцией f от n переменных называется отображение $f : V_n \to \mathbb{F}_2$. Множество всех булевых функций от n переменных будем обозначать через $\mathcal{F}_n$. Носителем функции $f \in \mathcal{F}_n$ называется множество $1_f = \{x \in V_n : f(x) = 1\}$. *Весом* $\operatorname{wt}(f)$ булевой функции $f \in \mathcal{F}_n$ называется мощность её носителя.

Через $\mathbb{N}_0$ обозначим множество $\mathbb{N} \cup \{0\}$. Под *фильтрующим генератором* будем понимать отображение из $\mathbb{N}_0 \times V_n$ в $\mathbb{F}_2$, определяющееся невырожденным линейным отображением $A : V_n \to V_n$ и уравновешенной булевой функцией $f \in \mathcal{F}_n$, которое ставит в соответствие числу i и вектору $u^* \in V_n$ бит $z_i = f(A^i(u^*))$. Вектор u^* будем называть *ключом* или *начальным заполнением* фильтрующего генератора, а последовательность битов $z_0, z_1, \ldots$ — его *выходной последовательностью*. Результатом зашифрования открытого текста $x \in V_N$ на ключе $u^* \in V_n$ с помощью потокового шифра, построенного на основе фильтрующего генератора, является вектор $y \in V_N$, такой, что $y_i = x_i \oplus z_i$ для любого $i \in \{0, \ldots, N-1\}$. Другими словами, $y = x \oplus z$, где $z = (z_0, z_1, \ldots, z_{N-1}) \in V_N$ — начальный отрезок длины N выходной последовательности фильтрующего генератора.

Определение 1 [12]. Пусть имеется случайная выборка из n элементов без возвращения из конечной совокупности мощности N, при этом ровно D из них обладают заданным свойством. Пусть случайная величина x — число элементов из выборки, обладающих заданным свойством. Говорят, что случайная величина x имеет *гипергео-*

метрическое распределение с параметрами N, D, n ($x \sim HG(D, N, n)$), если справедливо соотношение

$$\mathsf{P}[x = k] = \binom{D}{k}\binom{N-D}{n-k}\Big/\binom{N}{n}.$$

2. Предварительные сведения

Метод из работы [10], как было отмечено, позволяет восстанавливать ключ фильтрующего генератора тем эффективнее, чем более «точная» плоскостная аппроксимация им используется. В настоящем разделе вводится понятие однородной плоскостной аппроксимации (п. 2.1), кратко описывается метод восстановления ключа на её основе (п. 2.2), приводятся соотношения между параметрами плоскостной аппроксимации и основными численными характеристиками метода (п. 2.3).

2.1. Плоскостные аппроксимации

Для дальнейшего изложения приведём необходимые определения основных конструкций, введённых в [10].

Всюду далее A — линейное отображение из V_n в V_n, а f — функция из $\mathcal{F}_n$.

Пусть $m \in \mathbb{N}$, $\mathbb{L} = (L_0, \ldots, L_m)$, где все L_i являются плоскостями в V_n, и $\mathbb{T} = (t_0, t_1, \ldots, t_m) \in \mathbb{Z}^{m+1}$, где $t_0 = 0$, $t_0 \leqslant t_1 < \ldots < t_m$. Тройка $\mathrm{Traj} = (m, \mathbb{L}, \mathbb{T})$ называется *траекторией* для A, если выполнены соотношения $L_i = A^{t_i - t_{i-1}}(L_{i-1})$, $i = 1, \ldots, m$. При этом m называется *длиной траектории*, а L_0 — *начальной плоскостью*.

Пусть $\mathrm{Traj} = (m, \mathbb{L}, \mathbb{T})$ — траектория для некоторого линейного преобразования. Пусть также $\mathbb{B} = (b_1, \ldots, b_m)$, где все $b_i \in \mathbb{F}_2$, и $\mathbb{P} = (p_1, \ldots, p_m)$, где все $p_i \in [0; 1]$. Пару $(\mathbb{B}, \mathbb{P})$ будем называть *характеристикой* траектории Traj относительно функции $f \in \mathcal{F}_n$, если p_i — вероятность того, что при случайном равновероятном выборе вектора v из L_i значение $f(v)$ совпадает с константой b_i.

Отметим, что каждой траектории длины m соответствует 2^m характеристик, среди которых существует не более одной характеристики, у которой $p_i > 1/2$ для всех i. Характеристику, обладающую таким свойством, будем называть *положительной* относительно f. Траектория, для которой существует положительная характеристика, будем называть *подходящей траекторией*.

В целях визуализации понятия подходящей траектории на рис. 1 для $n = 6$ схематично изображена траектория длины 3. Включение векторов в плоскость носит условный характер, так как на практике плоскость может состоять из нескольких несвязанных областей из векторов. Плоскости пространства V_6 условно заключены в рамку; плоскости, вошедшие в траекторию, выделены отдельно.

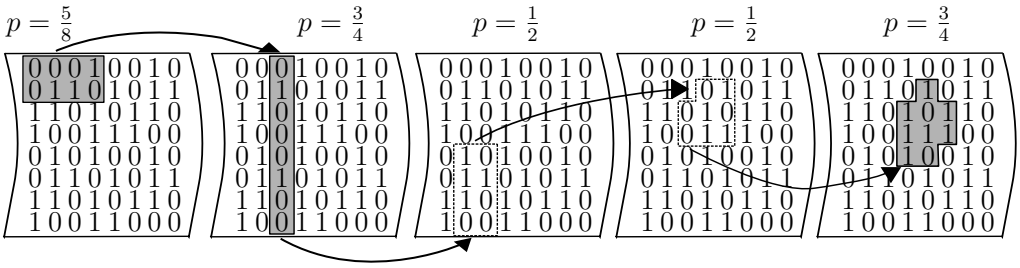

Рис. 1. Траектория $\mathrm{Traj} = (3, \mathbb{L}, \mathbb{T})$, где $\mathbb{T} = (0, 0, 1, 4)$, характеристика ($\mathbb{B} = (0, 0, 1)$, $\mathbb{P} = (5/8, 3/4, 3/4)$)

Конечный набор траекторий $\mathrm{Traj}^{(1)}, \ldots, \mathrm{Traj}^{(s)}$ для A, являющихся подходящими относительно функции f, с попарно различными начальными плоскостями будем называть *плоскостной аппроксимацией* функции f относительно отображения A.

Для плоскостной аппроксимации через $\mathbb{L}_{\mathrm{start}}$ будем обозначать множество, состоящее из начальных плоскостей входящих в аппроксимацию траекторий, т. е. $\mathbb{L}_{\mathrm{start}} = \{L_0^{(1)}, \ldots, L_0^{(s)}\}$.

Определение 2. *Однородной плоскостной аппроксимацией* будем называть плоскостную аппроксимацию, удовлетворяющую следующим условиям:

1) начальные плоскости аппроксимации являются смежными классами по одному и тому же подпространству и их объединение совпадает с ключевым пространством V_n, т. е. $\bigcup\limits_{L \in \mathbb{L}_{\mathrm{start}}} L = V_n$, и $\dim(L) = k$ для любого $L \in \mathbb{L}_{\mathrm{start}}$;
2) длины m_i всех траекторий равны между собой.

Всюду далее речь идёт именно об этом типе плоскостных аппроксимаций, которую для краткости будем называть однородной аппроксимацией.

2.2. Восстановление ключа с помощью однородной плоскостной аппроксимации

Под задачей восстановления ключа фильтрующего генератора по известному отрезку его выходной последовательности длины N будем понимать задачу поиска по известным $z_0, \ldots, z_{N-1}$, где $z_i \in \{0, 1\}$, такого значения $u^* \in V_n$, что

$$f(A^i(u^*)) = z_i, \;\; i = 0, \ldots, N-1.$$

Метод восстановления ключа состоит из двух этапов. На первом этапе последовательно просматриваются траектории плоскостной аппроксимации и выбираются те из них, в начальной плоскости которых вероятнее всего находится искомый ключ. Для проверки одной траектории $\mathrm{Traj} = (m, \mathbb{L}, \mathbb{T})$ необходимо сравнить биты характеристики траектории $b_1, \ldots, b_m$ и биты выходной последовательности генератора $z_{t_1}, \ldots, z_{t_m}$, где $b_i \in \mathbb{B}$ и $t_i \in \mathbb{T}$ для $i = 1, \ldots, m$. Всюду далее будем полагать, что для любой траектории из плоскостной аппроксимации выполняется неравенство $t_m < N$.

Пусть p — минимальное значение p_i среди вероятностей характеристик всех траекторий. При расчете характеристик метода будем заменять вероятности p_i на p. Отметим, что оценки, полученные в рамках такого допущения, хотя и будут более грубыми, но останутся справедливыми. Проверка принадлежности ключа начальной плоскости траектории сводится к различению двух распределений Бернулли по выборке объёма m. Благодаря тому, что все траектории имеют одинаковую длину m и одинаковые вероятности p появления преобладающих значений на плоскостях, на первом этапе достаточно построить единое для всех траекторий решающее правило, отвечающее за принятие начальной плоскости траектории в качестве возможного расположения искомого ключа генератора. В силу введённых в [10] статистических гипотез данному решающему правилу соответствуют ошибки первого и второго рода α, β, где α — вероятность принять ложную траекторию, начальной плоскости которой ключ не принадлежит; β — вероятность отклонить истинную траекторию.

Второй этап метода заключается в полном опробовании векторов из начальных плоскостей тех траекторий, что успешно прошли первый этап. Для этого необходимо для каждого вектора плоскости построить отрезок выходной последовательности генератора длины N. При точном совпадении построенного и известного отрезков выходной последовательности рассматриваемый ключ объявляется искомым. При отсутствии такого совпадения метод заканчивает свою работу, не найдя ключа.

Как и в работе [10], будем предполагать, что основные параметры метода восстановления ключа далеки от своих крайних значений (например, значение параметра k не приближается к 0 или к n), так как именно в этом предположении предлагаемый метод имеет существенное преимущество относительно полного опробования ключевого пространства.

2.3. Характеристики метода восстановления ключа

Трудоёмкость

Оценка средней трудоёмкости метода восстановления ключа для общего случая приведена в работе [10]. Для получения значения средней трудоёмкости для случая однородной аппроксимации воспользуемся следствием 1 из упомянутой работы.

Следствие 1 [10]. Пусть s — количество траекторий в плоскостной аппроксимации, $C_\alpha = \sum\limits_{j=1}^{s} |L_0^{(j)}|\alpha_j$. Обозначим через M список, состоящий из векторов, которые не принадлежат ни одной начальной плоскости: $M = V_n \setminus \bigcup\limits_{L\in\mathbb{L}_{\text{start}}} L$. Если плоскости из $\mathbb{L}_{\text{start}}$ не имеют пересечений, то справедливо следующее соотношение:

$$\mathrm{E}D(u,\gamma_u) = s + C_\alpha + \frac{|M|}{2^n}\left(|M| + \sum_{i=1}^{s}|L_0^{(i)}|\beta_i\right) + \frac{1}{2^n}\sum_{i=1}^{s}|L_0^{(i)}|^2(1-\alpha_i-\beta_i).$$

Для случая однородной аппроксимации формула имеет следующий вид:

$$\mathrm{E}D(u,\gamma_u) = 2^{n-k} + 2^{n-k}|L|\,\alpha + \frac{1}{2^n}|L|^2(1-\alpha-\beta) = 2^{n-k} + 2^n\alpha + 2^k(1-\alpha-\beta).$$

Будем рассматривать случаи, когда $\alpha \ll \beta$ (наиболее практически значимые). Таким образом, будем пренебрегать α в последнем слагаемом:

$$\mathrm{E}D(u,\gamma_u) = 2^{n-k} + 2^n\alpha + 2^k(1-\beta).$$

Надёжность

Надёжность метода в общем случае описывается формулой

$$\pi \geqslant 1 - \frac{1}{2^n}\sum_{i=1}^{s}|L_0^{(i)}|\,\beta_i,$$

где s — число траекторий в плоскостной аппроксимации.

В случае однородной аппроксимации $s = 2^{n-k}$, $\beta_i = \beta$ для всех i, поэтому данная оценка преобразуется следующим образом:

$$\pi \geqslant 1 - \frac{s}{2^n}|L|\,\beta = 1 - \beta.$$

3. Построение плоскостных аппроксимаций

В данном разделе описан переборный метод построения однородной аппроксимации фильтрующего генератора (п. 3.1) и предложена модель на основе случайных множеств (п. 3.2), в которой получены оценки его характеристик (п. 3.3). Приведено экспериментальное подтверждение релевантности предложенной модели (п. 3.4).

Пусть необходимо построить такую однородную аппроксимацию фильтрующего генератора, чтобы метод восстановления ключа на её основе имел трудоёмкость не больше Q, надёжность не меньше π_0 и требовал для своей работы минимально возможный объём выходной последовательности генератора.

3.1. Описание метода

Каждая траектория в однородной аппроксимации обладает следующими параметрами:

1) $k \in \{0, 1, \ldots, n\}$ — размерность плоскости в траектории, мощность плоскости будем обозначать через $\Omega = 2^k$;
2) m — длина траектории;
3) p — нижняя граница доли преобладающих значений функции на плоскости.

При построении траектории будем использовать параметр $\delta \in \{0, 1, \ldots, \Omega\}$, определяющий минимально приемлемое преобладание той или иной константы на плоскости. Пусть $T_0 = \Omega/2 - \delta/2$, $T_1 = \Omega/2 + \delta/2$, S — вес функции f на плоскости, входящей в однородную аппроксимацию. Тогда либо $S < T_0$ (плоскость содержит достаточное количество нулей функции), либо $S > T_1$ (плоскость содержит достаточное количество единиц функции). Таким образом, вероятность появления преобладающего значения функции на плоскости выражается через параметр δ следующим образом: $p = \dfrac{1}{2} + \dfrac{\delta}{2\Omega}$.

Метод построения однородной аппроксимации с параметрами k, m и p описывается следующим образом. Фиксируем произвольным образом подпространство L размерности k. Для каждого из 2^{n-k} смежных классов строится своя траектория. Для конкретного смежного класса, обозначаемого через L_0, выполняем следующие действия для каждого $i = 0, 1, 2, \ldots$:

1) $L_i = A^i(L_0)$;
2) обозначим через S_i вес функции f на плоскости L_i. Тогда если $|S_i - \Omega/2| > \delta/2$, то добавляем плоскость в строящуюся траекторию с соответствующей преобладающей константой; иначе переходим к п. 1;
3) если длина траектории равна m, то закончить построение траектории для текущего смежного класса.

Число проверок веса функции на плоскости при построении одной траектории определяет длину выходной последовательности генератора, необходимую для проверки принадлежности ключа начальной плоскости этой траектории на первом этапе работы алгоритма восстановления ключа. Таким образом, чем больше максимальное число проверок веса функции на плоскости среди траекторий однородной аппроксимации, тем больше и целевой параметр — объём выходной последовательности, требуемый для работы метода.

3.2. Математическая модель

При оценке характеристик метода будем считать, что все плоскости L_i, участвующие в проверке веса функции, являются множествами мощности 2^k, выбранными из множества всех подмножеств V_n случайно и равновероятно. Допустимость данного предположения обосновывается точностью получаемых с её помощью оценок параметров метода (см. п. 3.4).

Обозначим через $\widetilde{p}(\delta, k)$ вероятность принять случайное множество мощности 2^k в траекторию. Тогда $\widetilde{N}(\delta, k) = \dfrac{m}{\widetilde{p}(\delta, k)}$ — среднее количество шагов метода, которое необходимо выполнить для построения траектории длины m. Отметим, что величина $\widetilde{N}(\delta, k)$ равна среднему объёму выходной последовательности генератора, необходимому для восстановления ключа с заданными параметрами; именно её требуется минимизировать, согласно постановке задачи.

Перед началом работы метода построения однородной аппроксимации должны быть зафиксированы параметры k и δ.

Отметим, что стремление параметра k к нулю означает уменьшение мощности Ω плоскостей в траектории, а значит, приближение метода восстановления ключа к полному перебору векторов ключевого пространства на первом этапе работы алгоритма. При $k \to n$, то есть при увеличении мощности плоскости в траектории, трудоёмкость второго этапа алгоритма приближается к трудоёмкости полного перебора.

Рассмотрим подробнее влияние параметра δ на характеристики метода восстановления ключа при фиксированном значении k. Согласно математической модели, используемой для расчёта характеристик в [10], для принятия решения о возможном расположении ключа генератора в начальной плоскости траектории необходимо различать две простые статистические гипотезы по выборке объёма m. Для однородной аппроксимации эта задача сводится к построению оптимального критерия в схеме Бернулли с вероятностями $p_0 = \dfrac{1}{2}$ и $p_1 = \dfrac{1}{2} + \dfrac{\delta}{2\Omega}$. Чем больше δ, тем меньше объём выборки, необходимый для различения двух гипотез с заданными вероятностями ошибок первого и второго рода.

Таким образом, для фиксированного k при $\delta \to 0$ вероятность принятия случайной плоскости в траекторию увеличивается, но растёт и значение длины траектории m, требуемой для достижения заданной надёжности метода восстановления ключа. При $\delta \to \Omega$ длина траектории m уменьшается, но уменьшается и вероятность принять плоскость в траекторию, так как растёт требуемое преобладание на плоскости.

Возникает следующая задача: найти такие значения параметров δ и k, при которых величина $\widetilde{N}(\delta, k)$ минимальна для заданных ограничений трудоёмкости и надёжности метода восстановления ключа.

3.3. Характеристики метода

Математическая формулировка задачи минимизации

Как отмечено ранее, среднее количество проверок веса функции на плоскости, осуществляемых при построении траекторий, входящих в однородную аппроксимацию, равно $\widetilde{N}(\delta, k) = \dfrac{m}{\widetilde{p}(\delta, k)}$. Оценим числитель и знаменатель этой дроби.

Для оценки m найдём сначала вероятности ошибок первого и второго рода, возникающих при восстановлении ключа на основе однородной аппроксимации, построенной описанным методом с параметрами k и δ. Напомним, что для случая однородной аппроксимации эти вероятности одинаковы для всех траекторий:

- $\beta = 1 - \pi_0$ — вероятность ошибки второго рода;
- так как $Q = 2^{n-k} + \alpha\, 2^n + (1 - \beta)\, 2^k$, то вероятность ошибки первого рода $\alpha = 2^{-n}\left(Q - 2^{n-k} - \pi_0\, 2^k\right)$.

Получим выражение для длины траектории m, необходимой для работы метода восстановления ключа с заданными вероятностями ошибок первого и второго рода. На первом этапе метода для каждой траектории в однородной аппроксимации строится вектор $w = (c_1 \oplus \widetilde{z}_1, \ldots, c_m \oplus \widetilde{z}_m)$, $\widetilde{z}_i = z_{t_i}$, $i = 1, \ldots, m$. При этом если в начальном множестве данной траектории находится ключ, то $\mathsf{P}[w_i = 0] \geqslant q_1 = \dfrac{1}{2} + \dfrac{\delta}{2\Omega}$, в противном случае $\mathsf{P}[w_i = 0] = q_0 = \dfrac{1}{2}$. Для статистического различения двух распределений Бернулли с вероятностями успеха q_0 и q_1, вероятностями ошибок первого и второго

рода α и β соответственно потребуется объём материала [9]

$$m \approx \frac{\left(u_\alpha\sqrt{q_0(1-q_0)}+u_\beta\sqrt{q_1(1-q_1)}\right)^2}{(q_1-q_0)^2},$$

где u_α, u_β — квантили стандартного нормального распределения.

Оценим величину $\widetilde{p}(\delta,k)$ — вероятность принять случайное множество мощности Ω в траекторию. Для этого оценим распределение веса функции f на плоскостях L_i, выбираемых, согласно модели, случайно и равновероятно. Выбор таких множеств можно представить в виде случайной выборки без возвращения мощности Ω из конечной совокупности 2^n элементов, при этом ровно 2^{n-1} из них обладают тем свойством, что значение функции f на них равно единице. Рассмотрим случайную величину S, равную количеству точек выбранного множества, на которых функция f равна единице. Случайная величина S имеет гипергеометрическое распределение с параметрами $2^n, 2^{n-1}, \Omega$, то есть $S \sim \mathrm{HG}(2^{n-1}, 2^n, \Omega)$ [12].

Для удобства вычислений перейдём от гипергеометрического распредления к нормальному $\mathrm{N}(\Omega/2, \Omega/4)$ следующим образом. Так как $\mathrm{HG}(D,N,n) \approx \mathrm{Bin}(n, D/N)$ при $N \to \infty$ [12], то будем считать, что $S \sim \mathrm{Bin}(\Omega, 1/2)$. Поскольку $\mathrm{Bin}(n,p) \approx \mathrm{N}(np, npq)$ при больших n, где $\mathrm{N}(np,npq)$ — нормальное распределение с математическим ожиданием np и дисперсией npq, то $\mathrm{Bin}(\Omega,1/2) \approx \mathrm{N}(\Omega/2, \Omega/4)$, $p = q = 1/2$.

Пусть $\Phi(y) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y} e^{-x^2/2}dx$ — функция распределения стандартной нормальной случайной величины. Тогда

$$\begin{aligned}\widetilde{p}(\delta,k) &= 1 - \mathsf{P}\left[T_0 \leqslant S \leqslant T_1\right] = 1 - \mathsf{P}\left[\frac{T_0-\Omega p}{\sqrt{\Omega pq}} \leqslant \frac{S-\Omega p}{\sqrt{\Omega pq}} \leqslant \frac{T_1-\Omega p}{\sqrt{\Omega pq}}\right] = \\ &= 1 - Pr\left[-\frac{\delta}{\sqrt{\Omega}} \leqslant \frac{S-\Omega p}{\sqrt{\Omega pq}} \leqslant \frac{\delta}{\sqrt{\Omega}}\right] = 1 - \left(\Phi\left(\frac{\delta}{\sqrt{\Omega}}\right) - \Phi\left(-\frac{\delta}{\sqrt{\Omega}}\right)\right) = \\ &= 2\left(1 - \Phi\left(\frac{\delta}{\sqrt{\Omega}}\right)\right).\end{aligned}$$

Таким образом, объём выходной последовательности генератора, необходимый для построения траектории, равен

$$\widetilde{N}(\delta,k) = \frac{m}{\widetilde{p}(\delta,k)} \approx \frac{\left(u_\alpha + u_\beta\sqrt{1-(\delta/\Omega)^2}\right)^2 (\Omega/\delta)^2}{2\left(1-\Phi\left(\delta/\sqrt{\Omega}\right)\right)}.$$

Необходимо минимизировать величину $\widetilde{N}(\delta,k)$ при возможных значениях параметров $\delta \in \{0,1,\ldots,\Omega\}$ и $k \in \{0,1,\ldots,n\}$.

Решение задачи минимизации

Перейдём к непрерывной переменной $t = \delta/\sqrt{\Omega}$, $t \in (0;\sqrt{\Omega}]$. Тогда

$$\widetilde{N}(\delta,k) \approx \frac{\Omega}{2}\left(u_\alpha + u_\beta\sqrt{1-t^2/\Omega}\right)^2 \frac{1}{t^2(1-\Phi(t))}.$$

Рассмотрим наиболее интересный для решаемой задачи случай, когда $Q \ll 2^n$, а π_0 принимает значения из области, естественной для практических применений (например, $\pi_0 = 1/2$ или $\pi_0 = 1/10$). При таком условии $\alpha \ll \beta$ и, следовательно, $u_\alpha \gg u_\beta$. Поэтому далее будем полагать $\widetilde{N}(\delta,k) \approx \frac{\Omega}{2}\frac{u_\alpha^2}{t^2(1-\Phi(t))}$.

Найдём значение переменной t, при котором достигается максимум выражения $f(t) = t^2(1 - \Phi(t))$, $t \in (0; \sqrt{\Omega}]$. Приравнивая производную $f'(t)$ к нулю, получим уравнение

$$2(1 - \Phi(t)) = \frac{t}{\sqrt{2\pi}} e^{-t^2/2},$$

которое эквивалентно

$$\frac{t}{2} = \frac{1 - \Phi(t)}{\varphi(t)},$$

где $\varphi(t) = \dfrac{e^{-t^2/2}}{\sqrt{2\pi}}$.

Выражение $R(t) = \dfrac{1 - \Phi(t)}{\varphi(t)}$ называется отношением Миллса [13]. Функция $R(t)$ монотонно убывает [14], значит, уравнение имеет в точности одно решение. Получить аналитическое выражение для корня данного уравнения не представляется возможным, но можно его посчитать с любой наперёд заданной точностью. Приведём значение корня с точностью до 0,00001: $t_0 = 1{,}19061$.

Произведём обратную замену и тем самым вернёмся к дискретным величинам: $\delta_0 \approx \lceil t_0\sqrt{\Omega} \rceil$. Значение $\widetilde{N}(\delta_0, k)$ в этой точке равно

$$\widetilde{N}(\delta_0, k) \approx \frac{\Omega}{2} \cdot u_\alpha^2 \cdot C_\Phi,$$

где $C_\Phi = t_0^{-2}(1 - \Phi(t_0))^{-1} \approx 6{,}03442$.

Так как для малых α справедливо $u_\alpha \approx \sqrt{-\ln(2\pi\alpha^2)}$, то u_α изменяется не более чем на n при допустимых k. В то же время Ω растет по k экспоненциально. Таким образом, $\widetilde{N}(\delta_0, k)$ достигает минимума при минимально допустимом k. Допустимыми являются те $k \in \{1, 2, \ldots, n\}$, для которых $\alpha = 2^{-n}\left(Q - 2^{n-k} - \pi_0 2^k\right) > 0$. Минимально допустимое значение k определяется равенством

$$k = \left\lceil \log_2 \left(\frac{Q - \sqrt{Q^2 - \pi_0 2^{n+2}}}{2\pi_0} \right) \right\rceil.$$

В табл. 1 указаны параметры k, δ и величины m, $\widetilde{N}$ при $n = 128$, $\pi_0 = 1/2$ для различных значений трудоёмкости.

Таблица 1

Q	k	δ	m	$\widetilde{N}$
2^{70}	59	2^{30}	2^{73}	2^{73}
2^{80}	49	2^{25}	2^{63}	2^{63}
2^{90}	39	2^{20}	2^{53}	2^{53}

3.4. Экспериментальная проверка релевантности математической модели

Рассуждения из п. 3.2 основаны на замене плоскостей, участвующих в однородной аппроксимации, случайными множествами. Проверим точность полученных оценок.

Для этого рассмотрим следующий фильтрующий генератор. Пусть $n = 32$, а преобразование A задаётся регистром сдвига с линейной обратной связью, характеристический многочлен которого является примитивным и равен $p(x) = x^{32} + x^7 + x^6 + x^2 + 1$. Перейдём к описанию фильтрующей функции. Пусть $\pi_i : V_4 \to V_4$ — перестановки, заданные в [15] ($i = 0, 1, \ldots, 7$); $\Psi : V_{32} \to V_{32}$ — отображение $\Psi(x) = \Psi(x_0 || \ldots || x_7) = \pi_0(x_0) || \ldots || \pi_7(x_7)$, где $x = x_0 || \ldots || x_7 \in V_{32}$, $x_i \in V_4$, $i = 0, 1, \ldots, 7$; $\mathcal{S}(x) : V_{32} \to V_{32}$ — циклический сдвиг вправо на 11 бит; $\mathcal{X}(x) : V_{32} \to \mathbb{F}_2$ — сумма по модулю 2 бит вектора x. Эксперименты проводились для функции $f(x) = \mathcal{X}(\Psi(\mathcal{S}(\Psi(\mathcal{S}(\Psi(x))))))$.

Пусть необходимо построить такую однородную аппроксимацию, чтобы метод восстановления ключа на её основе имел трудоёмкость $Q = 2^{24}$ и надёжность не меньше $\pi_0 = 1/2$. Согласно выражениям, представленным в п. 3.3, оптимальные значения параметров для построения однородной аппроксимации следующие:

$$k = 9, \quad \delta = 27.$$

При данных параметрах для достижения целевых значений надёжности и трудоёмкости метода восстановления ключа потребуется отрезок выходной последовательности длины $\widetilde{N} = 12861$, длина траекторий составляет $m = 3007$.

Для проверки релевантности предложенной математической модели разработана программа, реализующая построение траектории плоскостной однородной аппроксимации для заданных параметров метода при случайном выборе начальной плоскости L_0. Для подсчёта среднего значения $\widetilde{N}$ для каждой пары параметров (k, δ) процедура построения траектории была запущена 100 раз. В табл. 2 представлены значения целевого параметра $\widetilde{N}$, полученные с помощью формулы из п. 3.3 ($\widetilde{N}_t$) и в результате экспериментов ($\widetilde{N}_e$). Жирным шрифтом выделены столбцы, в которых достигается минимальное значение длины отрезка выходной последовательности.

Таблица 2

$k = 9$							
δ	24	25	26	**27**	28	29	30
$\widetilde{N}_t$	13121	12974	12892	**12866**	12898	12981	13120
$\widetilde{N}_e$	14094	13911	13880	**13849**	13975	14018	14174
$k = 10$							
δ	36	37	38	**39**	40	41	42
$\widetilde{N}_t$	23573	23487	23456	**23477**	23549	23667	23833
$\widetilde{N}_e$	26201	24736	23451	**24635**	26256	25026	23842
$k = 11$							
δ	51	52	53	**54**	55	56	57
$\widetilde{N}_t$	45406	45282	45213	**45198**	45224	45302	45429
$\widetilde{N}_e$	47048	48714	46999	**45278**	46935	48848	47203
$k = 12$							
δ	74	75	76	**77**	78	79	80
$\widetilde{N}_t$	89005	88925	88890	**88905**	88966	89075	89233
$\widetilde{N}_e$	89085	91210	93756	**91362**	89037	91424	94197

Данные эксперименты подтверждают, что теоретически предсказанные результаты действительно являются оптимальными, а величина целевого параметра $\widetilde{N} \approx 12861$.

Заключение

В настоящей работе получены выражения для оптимальных характеристик переборного метода построения плоскостных аппроксимаций одного специального вида. Проведённый численный эксперимент показал, что данные выражения, будучи получены в рамках довольно сильного модельного предположения, позволили получить относительно точные значения характеристик метода. Уточнение оценок за счёт конкретизации модели остаётся открытым вопросом для будущих исследований. Другим открытым вопросом является исследование проблемы построения плоскостных аппроксимаций более общего вида.

ЛИТЕРАТУРА

1. `http://www.bluetooth.org` — Bluetooth™. Bluetooth Specification, version 1.2. 2003. P. 903–948.
2. *Briceno M., Goldberg I., and Wagner D.* A pedagogical implementation of A5/1. 1999. `https://cryptome.org/jya/a51-pi.htm`.
3. *Ekdahl P. and Johansson T.* A new version of the stream cipher SNOW // LNCS. 2003. V. 2595. P. 47–61.
4. *Hell M., Johansson T., and Meier W.* Grain: stream cipher for constrained environments // Int. J. Wireless Mobile Computing. 2007. No. 2(1). P. 86–93.
5. *Siegenthaler T.* Decrypting a class of stream cipher using ciphertext only // IEEE Trans. Comput. 1985. V. C-34(1). P. 81–85.
6. *Meier W. and Staffelbach O.* Fast correlation attacks on certain stream cipher // J. Cryptology. 1989. V. 1. No. 3. P. 159–176.
7. *Courtois N. and Meier W.* Algebraic attacks on stream ciphers with linear feedback // LNCS. 2003. V. 2656. P. 345–359.
8. *Courtois N.* Fast algebraic attacks on stream ciphers with linear feedback // LNCS. 2003. V. 2729. P. 176–194.
9. *Логачев О. А., Сальников А. А., Ященко В. В.* Корреляционная иммунность и реальная секретность // Математика и безопасность информационных технологий. Материалы конф. в МГУ 23–24 октября 2003 г. М.: МЦНМО, 2004. С. 165–171.
10. *Алексеев Е. К., Кущинская Л. А.* Обобщение одного метода восстановления ключа фильтрующего генератора // Дискретная математика. 2017. Т. 29. № 4. С. 3–27.
11. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра: Учебник. В 2-х т. Т. 1. М.: Гелиос АРВ, 2003. 416 с.
12. *Феллер В.* Введение в теорию вероятностей и ее приложения. Т. 1. М.: Мир, 1984.
13. *Mills J. P.* Table of the ratio: area to bounding ordinate, for any portion of normal curve // Biometrika. 1986. V. 18. No. 3/4. P. 395–400.
14. *Gasull A. and Utzet F.* Approximating Mills ratio // J. Math. Anal. Appl. 2014. V. 420. Iss. 2. P. 1832–1853.
15. ГОСТ Р 34.12–2015. Информационная технология. Криптографическая защита информации. Блочные шифры. М.: Стандартинформ, 2015.

REFERENCES

1. `http://www.bluetooth.org` — Bluetooth™. Bluetooth Specification, version 1.2, 2003, pp. 903–948.

2. *Briceno M., Goldberg I., and Wagner D.* A pedagogical implementation of A5/1. 1999. `https://cryptome.org/jya/a51-pi.htm`.
3. *Ekdahl P. and Johansson T.* A new version of the stream cipher SNOW. LNCS, 2003, vol. 2595, pp. 47–61.
4. *Hell M., Johansson T., and Meier W.* Grain: stream cipher for constrained environments. Int. J. Wireless Mobile Computing, 2007, no. 2(1), pp. 86–93.
5. *Siegenthaler T.* Decrypting a class of stream cipher using ciphertext only. IEEE Trans. Comput., 1985, vol. C-34(1), pp. 81–85.
6. *Meier W. and Staffelbach O.* Fast correlation attacks on certain stream cipher. J. Cryptology, 1989, vol. 1, no. 3, pp. 159–176.
7. *Courtois N. and Meier W.* Algebraic attacks on stream ciphers with linear feedback. LNCS, 2003, vol. 2656, pp. 345–359.
8. *Courtois N.* Fast algebraic attacks on stream ciphers with linear feedback. LNCS, 2003, vol. 2729, pp. 176–194.
9. *Logachev O. A., Sal'nikov A. A., and Yashchenko V. V.* Korrelyatsionnaya immunnost' i real'naya sekretnost' [Correlation immunity and real secrecy]. Matematika i Bezopasnost' Informatsionnykh Tekhnologiy, Moscow, MCCME Publ., 2004, pp. 165–171. (in Russian)
10. *Alekseev E. K. and Kushchinskaya L. A.* Generalization of one method of a filter generator key recovery. Discrete Math. Appl., 2019, vol. 29, no. 2, pp. 69–87.
11. *Glukhov M. M., Elizarov V. P., and Nechaev A. A.* Algebra [Algebra]. V. 1, Moscow, Gelios ARV Publ., 2003. 416 p.
12. *Feller V.* Vvedenie v teoriyu veroyatnostey i ee prilozheniya [An Introduction to Probability Theory and its Applications]. V. 1, Moscow, Mir Publ., 1984.
13. *Mills J. P.* Table of the ratio: area to bounding ordinate, for any portion of normal curve. Biometrika, 1986, vol. 18, no. 3/4, pp. 395–400.
14. *Gasull A. and Utzet F.* Approximating Mills ratio. J. Math. Anal. Appl., 2014, vol. 420, iss. 2, pp. 1832–1853.
15. GOST R 34.12–2015. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Blochnye shifry [Information Technology. Information Security. Block Ciphers]. Moscow, Standartinform Publ., 2015. (in Russian)

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.17 DOI 10.17223/20710410/58/7

ПРЯМОЙ МЕТОД ВЫЧИСЛЕНИЯ ЦИКЛОВ ЯЧЕЕК КАРТЫ БЛОКА ПРОСТОГО ПЛАНАРНОГО ГРАФА

Б. Н. Иванов

Дальневосточный федеральный университет, г. Владивосток, Россия

E-mail: ivanov.bn@dvfu.ru

Предлагаемый алгоритм вычисления циклов ячеек карты блока графа простого планарного является расширением классического алгоритма поиска в глубину циклов DFS-базиса. Ключевой идеей модификации алгоритма является стратегия правого обхода при прохождении графа в глубину. Начальной вершиной при правом обходе назначается вершина с минимальной координатой по оси OY. Выход из начальной вершины выполняется по ребру с минимальным полярным углом. Продолжение обхода из каждой следующей вершины осуществляется по ребру с минимальным полярным углом относительно ребра, по которому пришли в текущую вершину. Вводится двухуровневая структура вложенности циклов — основной и нулевой уровни вложенности. Все циклы базиса относятся к основному уровню. Каждый из циклов может иметь и нулевой уровень вложенности в другом основном для него цикле, если он вложен в него и не вложен ни в какой другой цикл из основного цикла. При правом обходе циклы нулевой вложенности являются смежными основному циклу и не имеют между собой общих вершин вне основного цикла. Указанные два свойства позволили в каждом цикле базиса последовательно выделить и исключить из него все циклы нулевой вложенности, применяя операцию симметрической разности. Показано, что оставшаяся часть базисного цикла является циклом ячейки карты блока. Сложность каждого шага алгоритма не превышает квадратичной относительно числа вершин планарного графа.

Ключевые слова: *циклы планарного графа, циклы блока графа, планарный граф.*

A DIRECT METHOD FOR CALCULATING CELL CYCLES OF A BLOCK MAP OF A SIMPLE PLANAR GRAPH

B. N. Ivanov

Far Eastern Federal University, Vladivostok, Russia

The proposed algorithm for calculating the cycles of the cells the simple planar graph block map is an extension of the classical depth-first search algorithm for cycles of the DFS-basis. The key idea of the modification of this algorithm is the strategy of right-hand traversal when passing the graph in depth. The vertex with the minimum coordinate on the OY axis is assigned as the starting vertex in the right-hand traversal. The exit from the initial vertex is performed along the edge with the minimum polar

angle. The continuation of the traversal from each next vertex is carried out along an edge with a minimum polar angle relative to the edge along which arrived at the current vertex. A two-level structure of nested cycles is introduced. This is the main level and the zero level of nesting. All cycles of the basis belong to the main level. Each of the cycles can additionally have a zero level of nesting in another main cycle for it, if it is nested in the main cycle and not nested in any other cycle from the main cycle. With the right-hand traversal, zero nesting cycles are adjacent to the main cycle and do not have common vertices outside the main cycle. These two properties allowed in each basis cycle sequentially select and exclude from it all its zero nesting cycles, using the symmetric difference operation. It is shown that the rest of the basic cycle is the cycle of the block map cell. The complexity of each step of the proposed algorithm does not exceed the quadratic complexity with respect to the number of vertices of the simple planar graph.

Keywords: *planar graph cycles, graph block cycles, planar graph.*

Введение

Рассмотрим задачу раскраски территорий государств на карте поверхности Земли. В равной степени вместо государств можно взять любые другие области на карте. Исходными данными в этой задаче выступают география поверхности Земли (береговая черта) и границы государств. Данной модели поставим в соответствие конечный *простой планарный* граф $G = (X, U)$, где U — рёбра графа (границы государств); X — вершины (точки пересечения границ). К такому графу всегда можно свести рассматриваемую карту поверхности Земли, включая фиктивные вершины на границах или береговой черте.

Будем пользоваться терминами и обозначениями, принятыми в теории графов [1, с. 11]. *Подграфом* графа $G = (X, U)$ называется такой граф $G' = (X', U')$, что $X' \subseteq X$, $U' \subseteq U$. *Остовным* подграфом графа $G = (X, U)$ называется такой граф $G = (X, U')$, что $U' \subseteq U$. *Простой* граф — это неориентированный граф без петель, в котором любая пара вершин соединена не более чем одним ребром. В таком графе каждое ребро однозначно представляется неупорядоченной парой вершин $u = (x, y)$, где $u \in U$, $x, y \in X$. В этом случае говорят, что ребро u *инцидентно* вершинам x и y или вершины x и y инцидентны ребру u. *Маршрутом* в простом графе называется конечная последовательность вершин

$$S = (x_0, x_1, \ldots, x_{n+1}), \tag{1}$$

где каждая пара соседних вершин x_i и x_{i+1} определяет ребро $u_i = (x_i, x_{i+1})$ в графе. Тогда маршрут (1) можно записать и как последовательность рёбер $S = (u_0, u_1, \ldots, u_n)$. Маршрут называется *замкнутым*, если $x_0 = x_{n+1}$. Маршрут называется *цепью*, если все его рёбра различные. Замкнутая цепь называется *циклом*. Цепь называется *простой*, если никакая вершина в ней не повторяется. Простая замкнутая цепь называется *простым циклом*.

Укладка на плоскости рёбер конечного *простого планарного графа* без самопересечений определяет разбиение такой плоскости, называемое *планарным подразбиением* или *картой* графа [2, с. 31]. Карта фиксирует на плоскости геометрию рёбер графа, что позволяет на этой плоскости ввести *прямоугольную систему координат* и определить в ней координаты каждой вершины графа и положение рёбер. Координаты вершин определяют их взаимное расположение. В частности, можно найти вершину

графа, координата которой вдоль оси OY минимальная. В этом случае имеет смысл говорить о вершинах и рёбрах, *локализованных внутри цикла*, *заключенных внутри цикла* или *вложенных в цикл*. Такие вершины и рёбра графа геометрически располагаются внутри цикла. Сама карта складывается из простых многоугольников, которые будем называть *ячейками* карты. Каждый цикл ячейки карты графа представляет собой границы отдельного государства.

Замечание 1. Представление рёбер на плоскости будем выполнять кусочно-линейными кривыми. Рассматриваемый алгоритм вычисления циклов предполагает наличие в графе вершины с минимальной координатой вдоль оси OY среди всех вершин и *узлов рёбер* графа. Если узел с минимальной координатой по оси OY среди всех узлов рёбер меньше соответствующей координаты любой вершины графа, то такой узел необходимо перевести в разряд вершин графа. Поэтому далее полагаем, что в графе существует *единственная* вершина с минимальной координатой по оси OY среди всех вершин и узлов рёбер графа. Если таких вершин несколько, то уменьшим незначительно координату по оси OY одной из них и получим граф с единственной такой вершиной.

На рис. 1 показаны четыре ребра графа $(s,x),(x,a),(x,b),(x,c)$ с общей вершиной x и их узлы s_i,a_i,b_i,c_i кусочно-линейной интерполяции. Взаимное расположение кусочно-линейных рёбер $(x,a),(x,b),(x,c)$ относительно ребра (s,x) будем устанавливать по их первым отрезкам $(s_1,x),(x,a_1),(x,b_1),(x,c_1)$, примыкающим к вершине x (рис. 2). Для каждой пары отрезков $\{(s_1,x),(x,a_1)\}$, $\{(s_1,x),(x,b_1)\}$, $\{(s_1,x),(x,c_1)\}$ вычисляем величину угла, заключённого между ними, в направлении против часовой стрелки от отрезка (s_1,x) к другому отрезку соответствующей пары. Такие углы будем называть *полярными*. Сам отрезок (s_1,x) имеет нулевой полярный угол. Остальные такие углы могут принимать значения в диапазоне $(0,2\pi)$. Под *сортировкой рёбер* $(x,a),(x,b),(x,c)$ относительно ребра (s,x) будем понимать их расположение в порядке увеличения их полярных углов относительно ребра (s,x).

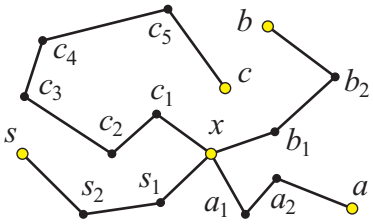

Рис. 1. Кусочно-линейные рёбра (s,x), (x,a), (x,b), (x,c) границ государств

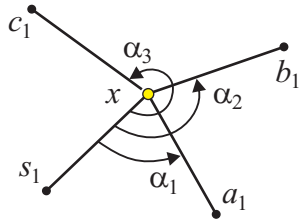

Рис. 2. Сортировка рёбер (x,a), (x,b), (x,c) относительно ребра (s,x) по полярным углам α_1, α_2, α_3

Рассмотрим другой случай вычисления полярных углов рёбер, инцидентных заданной вершине x, при условии, что координата вершины x вдоль оси OY меньше соответствующей координаты любой другой вершины карты графа. Перенесём начало системы координат в заданную вершину x. Тогда полярные углы всех рёбер, инцидентных вершине x, будем вычислять по рассмотренной ранее схеме, где роль ребра (x,s) выполняет ось OX. Углы могут принимать значения в диапазоне $(0,\pi)$.

Изучение циклической структуры графа является важной задачей общего анализа структуры графа. Первый практический алгоритм перечисления всех циклов графа опубликован в [3], его модифицированная версия — в [4]. Работы [3, 4] определили направление разработки алгоритмов генерации циклов графов. Показано, что в дан-

ной задаче метод поиска с возвращениями позволяет в среднем существенно понизить сложность полного перебора и перейти к практически значимым алгоритмам.

После этого был предложен ряд других алгоритмов [5, 6] порождения всех циклов графа, их обзор можно найти в работе [7]. Основу этих алгоритмов составляет *метод поиска в глубину* на графе [8]. Это адаптированный к структуре графа поиск с возвращениями.

Вычисление циклов графа с определёнными свойствами, как правило, выполняется на основе алгоритмов генерации всего множества циклов. Выделение циклов с заданными свойствами из общего множества может существенно увеличить сложность такого алгоритма.

Сегодня можно встретить и параллельные алгоритмы генерации циклов в графе [9, 10]. Авторы пытаются решать тестовые задачи большой размерности, используя параллельные вычисления. В этом случае для поиска циклов в работе [10] предлагается приближённый алгоритм как альтернатива полному перебору. Для информационных сетей актуальна задача сетевой надёжности передачи данных, и здесь первостепенное значение приобретает знание циклической структуры сети [11].

Вычисление циклов в простом планарном графе рассматривалось автором данной работы в [12], где в основу метода положены выделенные геометрические свойства циклов DFS-базиса. В настоящей работе предлагается прямой алгоритм вычисления циклов ячеек карты блока графа при обходе его в глубину.

1. Постановка задачи

Решается задача вычисления циклов ячеек карты блока конечного графа $G = (X, U)$ по данным множествам его вершин X и рёбер U. Вычисления проводятся в рамках классического алгоритма поиска в глубину циклов DFS-базиса. Новизна заключается в стратегии правого обхода при прохождении графа в глубину. Правый обход позволил для каждого цикла базиса последовательно выделить и исключить из него все циклы нулевой вложенности. Показано, что оставшаяся часть базисного цикла будет искомым циклом ячейки карты блока. Сложность предлагаемого алгоритма является квадратичной относительно числа вершин в графе.

2. Фундаментальное множество циклов карты графа

Полагаем, что $G = (X, U)$ — это конечный связный простой планарный граф.

Определение 1 (операции $\oplus$). Обозначим через $M = \{G_1, G_2, \ldots, G_{n_M}\}$ множество всех остовных подграфов $G_i = (X, U_i)$ графа $G = (X, U)$, где $U_i \subseteq U$. На множестве M определим бинарную операцию $\oplus$ так, что $G_i \oplus G_j = (X, U_i \oplus U_j)$ для всех $G_i, G_j \in M$, где символ $\oplus$ обозначает операцию симметрической разности: $U_i \oplus U_j = \{u : u \in U_i \cup U_j \wedge u \notin U_i \cap U_j\}$. Множество M с операцией $\oplus$ является абелевой группой. Единица группы — пустой подграф $O = (X, \varnothing)$. Обратным к G_k является тот же самый G_k, так как $G_k \oplus G_k = O$.

Определение 2 (смежных циклов). Простые циклы h_1 и h_2 будем называть *смежными*, если их пересечение $h_1 \cap h_2$ состоит из одного непустого непрерывного участка рёбер.

Из определения 1 имеем, что операция $\oplus$ применяется к остовным подграфам. Определим данную операцию $h_1 \oplus h_2$ для простых смежных циклов h_1 и h_2. Для этого рассмотрим определение цикла как остовного цикла.

Определение 3 (остовного цикла). Остовным циклом простого цикла h называется подграф $H = (X, [h])$, где $[h]$ — рёбра цикла h; X — вершины исходного графа

$G = (X, U)$. Подграф H тоже будем называть циклом, у него одна компонента связности, которая является простым циклом h, а вершины, не вошедшие в цикл h, являются изолированными. Каждый такой остовный цикл H будем представлять компонентой связности — простым циклом h.

Замечание 2. Далее под результатом операции $h_1 \oplus h_2$ простых смежных циклов h_1 и h_2 будем понимать простой цикл, представленный компонентой связности остовного цикла $H_1 \oplus H_2$, где $H_1 = (X, [h_1])$ и $H_2 = (X, [h_2])$.

Обозначим: $Z = \{Z_1, Z_2, \ldots, Z_{n_Z}\}$, где $Z_k = (X, [z_k])$ — остовные циклы; z_k — все простые циклы графа G. Пусть $L = \{\lambda_{i_1} Z_{k_1} \oplus \lambda_{i_2} Z_{k_2} \oplus \ldots \oplus \lambda_{i_Z} Z_{k_Z}\}$ — линейная оболочка циклов множества Z, где $\lambda_{i_r} \in \{0, 1\}$, $0 \cdot Z_{k_r} = O$ и $1 \cdot Z_{k_r} = Z_{k_r}$. Множество L является линейным векторным пространством циклов над полем $P = \langle \{0, 1\}, \oplus, \cdot \rangle$ [13, с. 61]. Уточним структуру, размерность и базис пространства L.

Пусть $T_0 = (X, U_0)$ — произвольное остовное дерево исходного графа $G = (X, U)$. Для дерева выполняется соотношение $|U_0| = |X| - 1$. Пусть $E = U \setminus U_0 = \{e_1, e_2, \ldots, e_{n_F}\}$ — рёбра, не вошедшие в T_0. Такие рёбра называются обратными. Любое обратное ребро $e_i \in E$ порождает в T_0 ровно один простой цикл C_i, где $e_i \in C_i$ и $e_i \notin C_j$, если $i \neq j$. Можно составить $n_F = |U \setminus U_0| = |U| - |X| + 1$ таких циклов C_i. Множество $F = \{C_1, C_2, \ldots, C_{n_F}\}$ называется *фундаментальным множеством циклов* и составляет базис пространства L [13, с. 63]. В простом планарном графе [13, с. 36] выполняется соотношение $|U| \leqslant 3|X| - 6$, что даёт основание пользоваться оценкой для числа циклов $n_F = |U| - |X| + 1 \leqslant 2|X| - 5$.

3. Блоковые свойства циклов DFS-базиса карты графа

Построение фундаментального множества циклов выполним *методом поиска в глубину со стеком*. Оригинальное описание метода представлено в [14]. Адаптированный вариант алгоритма для практического применения рассматривается в работе [15, с. 385]. При порождении фундаментального множества циклов методом поиска в глубину на простом связном графе строится дерево поиска, которое является остовным деревом графа и называется *DFS-деревом*. Каждое обратное ребро порождает один цикл относительно этого дерева. Всё множество таких циклов составляет *DFS-базис* циклов графа.

Предлагаемый алгоритм вычисления циклов ячеек карты графа ориентирован на свойства карты блока. Отметим несколько свойств циклов DFS-базиса карты блока, необходимых для обоснования алгоритма.

Теорема 1. Если пара циклов базиса C_1 и C_2 имеет хотя бы две общие вершины, то такие циклы являются смежными.

Доказательство. Пусть $e_1 \in C_1$ и $e_2 \in C_2$ — обратные рёбра циклов, $e_1 \notin C_2$ и $e_2 \notin C_1$. Пройдя по циклу C_1 в обе стороны от ребра e_1, найдём две вершины x_1 и x_2, принадлежащие также C_2 (рис. 3). Обозначим через $C_1(x_1, e_1, x_2)$, $C_2(x_1, e_2, x_2)$ участки циклов C_1 и C_2, содержащие соответственно рёбра e_1 и e_2. Их объединение $C_1(x_1, e_1, x_2) \cup C_2(x_1, e_2, x_2)$ есть простой цикл. Данная схема построения простого цикла из двух простых циклов при наличии у них не менее двух общих вершин рассматривается в [1, с. 110].

Обозначим через $R_1(x_1, x_2)$ и $R_2(x_1, x_2)$ участки циклов соответственно C_1 и C_2, не вошедших в простой цикл $C_1(x_1, e_1, x_2) \cup C_2(x_1, e_2, x_2)$. Если допустить, что $R_1(x_1, x_2) \neq$ $\neq R_2(x_1, x_2)$ (рис. 3), то вершины x_1 и x_2 будут связаны четырьмя маршрутами. Тогда при удалении всех обратных рёбер в графе получим остовное дерево T_0, в котором

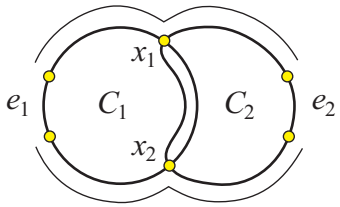

Рис. 3. Взаимное расположение циклов C_1 и C_2 базиса

вершины x_1 и x_2 останутся связанными двумя маршрутами $R_1(x_1, x_2) \neq R_2(x_1, x_2)$ (рис. 3), а значит, в T_0 будет цикл. Это противоречит тому, что T_0 есть дерево. Итак, имеем $R_1(x_1, x_2) = R_2(x_1, x_2)$, следовательно, C_1 и C_2 есть смежные циклы, а $C_1 \oplus C_2$ есть простой цикл, составленный из участков $C_1(x_1, e_1, x_2)$ и $C_2(x_1, e_2, x_2)$. ■

Определение 4 (нулевой вложенности). Пусть C_v и C_{v_j} — циклы базиса карты графа. Будем говорить, что цикл C_{v_j} имеет нулевую вложенность в цикле C_v, если цикл C_{v_j} вложен в C_v и не вложен ни в какой другой цикл из C_v.

Определение 5 (блока). В блоке графа все рёбра сильно циклически связаны [1, с. 109]. Это означает, что для любой пары рёбер u_1, u_2 блока существует такая последовательность простых циклов $H_1, H_2, \ldots, H_k$, что $u_1 \in H_1$, $u_2 \in H_k$ и любая пара соседних циклов H_i, H_{i+1} имеет по крайней мере одно общее ребро.

Замечание 3. Все простые циклы графа распределяются между блоками графа. Каждый простой цикл расположен целиком в своем блоке [1, с. 111]. Поэтому далее полагаем, что исходный граф $G = (X, U)$ есть невырожденный конечный блок карты графа (вырожденный блок состоит из одного ребра). В невырожденном блоке степень любой вершины не меньше двух.

Поиск в глубину на простом неориентированном графе осуществляется следующим образом. Когда посещаем вершину x, то далее идём по одному из рёбер (x, v), инцидентных вершине v. Если вершина v уже пройдена, то возвращаемся в x и выбираем другое ребро. Если вершина v не пройдена, то заходим в неё и применяем процесс прохождения рекурсивно уже с вершиной v.

Предлагаемая модификация алгоритма поиска в глубину [15, с. 385] касается выбора очередного ребра (x, v) при обходе карты блока. Выбор такого ребра выполняется в соответствии с двумя следующими правилами.

Правило 1 (начала обхода). Начальной вершиной поиска в глубину является вершина с минимальной координатой по оси OY (см. замечание 1). Инцидентные ей рёбра упорядочиваются по полярному углу против часовой стрелки. Выход из начальной вершины выполняется по ребру с минимальным значением полярного угла.

Правило 2 (правого обхода). Пусть в вершину x пришли по ребру (s, x). Упорядочим все рёбра, инцидентные вершине x, по полярному углу относительно ребра (s, x) против часовой стрелки (см. рис. 2). Ребру (s, x) назначается нулевой угол. Упорядочивание рёбер в каждой вершине выполняется один раз при первом её посещении. Для продолжения прохода по графу из вершины x будем выбирать ещё не пройденное ребро (x, v) с минимальным полярным углом.

Замечание 4. При поиске циклов DFS-базиса используется поиск в глубину со стеком, в котором хранятся пройденные вершины. Когда попадаем на обратное ребро (x, v), то найденный цикл будет состоять из этого ребра и рёбер, соединяющих вершины из верха стека до вершины v обратного ребра в стеке.

Определение 6 (правого обхода). Будем говорить, что цикл карты блока построен при «правом обходе», если он сформирован при обходе карты блока в соответствии с правилами 1 и 2.

Определение 7 (классификация рёбер). Обозначим через X_1 непустое подмножество множества вершин X простого графа $G(X)$, а через $X_2 = X \setminus X_1$ — его дополнение [1, с. 111]. Тогда

1) Ребро $u_1 = (x, y)$ и вершины $x, y \in X_1$ называются внутренними для X_1 и располагаются в подграфе $G(X_1)$.
2) Ребро $u_2 = (x, y)$, где $x \in X_1$, $y \in X_2$, называется соединяющим для X_1, а вершина x — соединяющей.
3) Ребро $u_3 = (x, y)$ и вершины $x, y \in X_2$ называются внешними для X_1.

Определение 8 (разделяющей вершины). В простом графе $G(X)$ вершина $a \in X$ называется разделяющей, если существует непустое подмножество $X_1 \subset X$, для которого $a \in X_1$ является единственной соединяющей вершиной. Тогда в любой вершине $b \neq a$ из X_1 все рёбра идут к вершинам из X_1.

Теорема 2 (о соединяющих вершинах цикла). Если множество вершин X_C, заключенных в простом цикле C карты графа, имеет соединяющие вершины, то эти вершины являются вершинами цикла C.

Доказательство. Исходный граф — это карта графа, а значит, пересечения рёбер цикла C с рёбрами графа являются вершинами цикла C. Следовательно, соединяющими вершинами множества X_C могут быть только вершины цикла C. ■

При «правом обходе» поиск в глубину начинается с вершины с минимальной координатой по оси OY (см. правило 1). Пройденные вершины сохраняются в стеке пройденных вершин и удаляются из стека при возврате по дереву поиска. Признаком начала возврата является обнаружение обратного ребра сформированного цикла. Пусть после обнаружения обратного ребра очередного сформированного цикла C_v вернулись по рёбрам этого цикла до первой его вершины a_0, которая имеет не пройденные инцидентные ей ребра. Данная вершина a_0 является началом формирования нового цикла C_0. Рассмотрим формирование данного цикла. Вершина a_0 имеет не пройденное инцидентное ей ребро, с которого можно начать строить простую цепь по карте блока. Степень любой вершины блока (невырожденного) не меньше 2. Поэтому при посещении какой-либо вершины первый раз (новой вершины v_i) из неё всегда можно продолжить простую цепь по ещё не пройденному ребру. Пусть такая последовательность из новых пройденных вершин $v_1 v_2 \dots v_n$ закончится обратным ребром в ранее пройденной вершине b_0. В стеке пройденных вершин будем иметь следующую последовательность вершин, составляющих простой цикл:

$$C_0 = (b_0 r_1 r_2 \dots r_m \, a_0 v_1 v_2 \dots v_n b_0). \tag{2}$$

Здесь a_0 — вершина начала формирования цикла; b_0 — вершина начала цикла; $r_1 r_2 \dots r_m$ — ранее пройденные вершины до вершины a_0; $v_1 v_2 \dots v_n$ — новые пройденные вершины после вершины a_0.

Определение 9 (нулевой вершины цикла). Вершину a_0 в цикле (2) будем называть *нулевой вершиной* цикла C_0 как вершину начала формирования данного цикла.

Теорема 3 (структура цикла при «правом обходе»). Вершины и рёбра цикла C_0, сформированного при «правом обходе» карты графа согласно формуле (2), удовлетворяют следующим свойствам:

1) все непройденные рёбра, инцидентные вершинам цикла C_0, кроме вершины b_0, локализованы внутри данного цикла;
2) вершина b_0 не совпадает ни с одной из новых вершин $v_1, v_2, \ldots, v_n$ цикла C_0;
3) если вершины a_0 и b_0 совпадают, то цикл C_0 является циклом, огибающим по периметру карту блока.

Доказательство.

1) При «правом обходе» формирования цикла C_0 в каждой его вершине, кроме вершины b_0, по построению (см. правила 1 и 2) непройденные инцидентные ей рёбра остаются внутри формируемого цикла.

2) Если допустить, что b_0 совпадает с одной из новых пройденных вершин $v_1, v_2, \ldots, v_n$, например $b_0 = v_k$, то вершина v_k будет единственной соединяющей для цикла $C_0 = (b_0 = v_k v_{k+1} \ldots v_n b_0 = v_k)$, так как при «правом обходе» в каждой новой вершине цикла, кроме b_0, инцидентные ей рёбра остаются внутри сформированного цикла. Следовательно, вершина $b_0 = v_k$ будет разделяющей для множества вершин, заключенных в цикле C_0. Это является противоречием, в блоке нет разделяющих вершин.

3) Пусть a_0 и b_0 совпадают, тогда цикл C_0 по формуле (2) запишется в виде $C_0 = (b_0 v_1 v_2 \ldots v_n b_0)$. В данном цикле только одна вершина b_0 не является новой. Как и в случае 2, такая вершина b_0 будет единственной соединяющей вершиной в цикле C_0. Если такой цикл C_0 охватывает не все вершины карты блока, то вершина b_0 является разделяющей. Это противоречие, в блоке нет разделяющих вершин. Следовательно, цикл C_0 является циклом, огибающим по периметру карту блока. ■

Теорема 4 (цикл по периметру блока). Первым циклом, сформированным при «правом обходе», будет цикл C_0, огибающий по периметру карту блока.

Доказательство. По правилу 1 начальной вершиной a_0 обхода карты блока является вершина с минимальной координатой по оси OY. Такой обход закончится в ранее пройденной вершине. Запишем данный маршрут обхода в виде $C_0 = (a_0, v_1, v_2, \ldots, v_n, b_0)$, где b_0 — ранее пройденная вершина, которая совпадает с одной из вершин $a_0, v_1, v_2, \ldots, v_n$. Покажем, что $b_0 = a_0$. Допустим, что $b_0 \neq a_0$, тогда $b_0 = v_k$, а маршрут обхода примет вид $C_0 = (a_0, v_1, v_2, \ldots, v_{k-1}, b_0, v_{k+1}, \ldots, v_n, b_0)$, где $B = (b_0, v_{k+1}, \ldots, v_n, b_0)$ — простой цикл. Вершина начала обхода a_0 не может быть локализована в цикле B, так как её координата по оси OY минимальная среди соответствующих координат всех вершин и узлов рёбер графа (см. правило 1 и замечание 1). Тогда вершина b_0 становится разделяющей вершиной блока, но в блоке нет разделяющих вершин. Следовательно, $b_0 = a_0$, а маршрут $C_0 = (a_0, v_1, v_2, \ldots, v_n, b_0)$ является простым циклом. Из теоремы 3 имеем, что такой цикл является циклом, огибающим по периметру карту блока. ■

Рассмотрим формирование вложенных циклов при «правом обходе». Пусть при возврате по дереву поиска после обнаружения обратного ребра цикла C_v вернулись по рёбрам этого цикла до первой его вершины a_1 (рис. 4), которая имеет непройденные инцидентные ей ребра. Вершина a_1 является нулевой вершиной формирования нового цикла C_{v_1}, который по формуле (2) запишется в следующем виде:

$$C_{v_1} = (b_1 r_1 r_2 \ldots r_m\, a_1, v_1, v_2, \ldots, v_n, b_1), \tag{3}$$

где a_1 — нулевая вершина цикла; b_1 — начальная вершина цикла; $r_1 r_2 \ldots r_m$ — ранее пройденные вершины до a_1; $v_1 v_2 \ldots v_n$ — новые пройденные вершины после a_1.

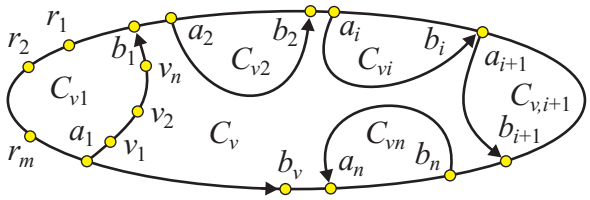

Рис. 4. Формирование цикла нулевой вложенности C_{v_1} в цикле C_v

Теорема 5 (первый вложенный цикл). Цикл C_{v_1}, представленный формулой (3), является циклом нулевой вложенности в цикле C_v и смежным ему.

Доказательство. Пусть обратное ребро цикла C_v заканчивается в вершине b_v. Из этой вершины начинается возврат (по часовой стрелке) по рёбрам данного цикла до первой вершины $a_1 \neq b_v$, у которой есть непройденные инцидентные ей рёбра (рис. 4). По теореме 3 при «правом обходе» все непройденные рёбра, инцидентные вершинам цикла C_v, кроме рёбер, инцидентных вершине b_v, остаются внутри этого цикла. Следовательно, все новые вершины v_i (см. формулу (3)) цикла C_{v_1} останутся внутри цикла C_v. Так как граф конечный, перечень новых вершин v_i цикла C_{v_1} завершится обратным ребром (v_n, b_1) в ранее пройденной вершине b_1, которая не совпадает ни с одной из новых вершин v_i (см. теорему 3). Так как вершина v_n локализована внутри цикла C_v, ранее пройденная вершина b_1 может быть только вершиной цикла C_v. Из вершин цикла C_v в стеке пройденных вершин остался участок вершин от a_1 до начальной вершины b_v . Поэтому b_1 может оказаться любой из них, исключая вершину a_1 (см. теорему 3).

Таким образом, циклы C_{v_1} и C_v имеют две общие вершины a_1 и b_1. По теореме 1 такие циклы являются смежными, а цикл C_{v_1} — вложенным в цикле C_v (рис. 4). Покажем, что цикл C_{v_1} является циклом нулевой вложенности. Если допустить, что существует вложенный цикл Q в цикле C_v, который охватывает цикл C_{v_1}, то нулевая вершина цикла Q должна размещаться на участке цикла C_v (против часовой стрелки) от вершины a_1 до конечной его вершины (рис. 4). На этом участке вершина a_1 была первой, которая имела непройденные инцидентные ей рёбра. Все такие рёбра остались вложенными в цикле C_{v_1} при «правом обходе» его формирования. Следовательно, цикл C_{v_1} не имеет охватывающих циклов и, значит, является циклом нулевой вложенности в цикле C_v. ■

Замечание 5. Из теоремы 5 следует, что цикл нулевой вложенности C_{v_1} в цикле C_v состоит из двух простых цепей (рис. 4). Первая цепь — это смежный участок указанных циклов от начальной вершины b_1 цикла C_{v_1} и до нулевой его вершины a_1. Вторая цепь — это участок цикла C_{v_1} из новых вершин.

При «правом обходе» после формирования любого вложенного цикла, например цикла C_{v_1}, мы переходим к формированию вложенных в него циклов, так как возвращаемся по рёбрам данного цикла, а все непройденные рёбра, инцидентные вершинам цикла C_{v_1}, остались локализованными внутри него (см. теорему 3).

Пусть мы вернулись по дереву поиска в цикл C_v, в котором был сформирован первый цикл C_{v_1} нулевой вложенности (рис. 4). На верху стека пройденных вершин имеем участок вершин цикла C_v от вершины b_1 до начальной вершины b_v, среди которых выполняется поиск нулевой вершины a_2 для построения следующего цикла C_{v_2}, который по теореме 5 будет циклом нулевой вложенности и смежным циклу C_v. По данной

схеме построим в цикле C_v все такие циклы нулевой вложенности

$$C_{v_1}, C_{v_2}, \ldots, C_{v_n}. \tag{4}$$

Каждый из таких циклов C_i определяется парой вершин a_i и b_i (см. рис. 4).

Теорема 6 (второй и следующие вложенные циклы). Сформированные при «правом обходе» циклы нулевой вложенности $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ в цикле C_v (см. формулу (4)) не имеют общих вершин вне цикла C_v.

Доказательство. Допустим, что циклы C_{v_i} и C_{v_j} имеют общую вершину w, которая не принадлежит циклу C_v. Тогда нулевые вершины этих циклов a_i и a_j связаны маршрутом M по рёбрам циклов C_{v_i} и C_{v_j} через вершину w. Исходная карта блока является связным графом. После удаления обратных рёбер получим остовное дерево T_0 карты [13, с. 63]. Вершины a_i и a_j в этом дереве останутся связанными маршрутом M, так как обратные рёбра примыкают к вершинам b_i и b_j. Но эти же вершины останутся связанными в остовном дереве T_0 и маршрутом N по рёбрам цикла C_v. Маршруты M и N образуют цикл, что невозможно в дереве. Значит, предположение неверное и никакая пара циклов C_{v_i} и C_{v_j} не имеет общих вершин вне цикла C_v. Отметим, что вершины цикла C_v могут быть общими вершинами соседних циклов нулевой вложенности. Например, начальная вершина b_i цикла C_{v_i} может быть нулевой вершиной a_{i+1} цикла $C_{v_{i+1}}$ (см. рис. 4). Перечень циклов, представленных формулой (4), получен полным последовательным перебором по алгоритму поиска в глубину. Поэтому любой другой цикл нулевой вложенности в цикле C_v должен совпадать с одним из них. ■

Теорема 7. Пусть $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — все циклы нулевой вложенности в цикле C_v, полученные в результате «правого обхода» карты блока. Тогда цикл

$$S_v = (((C_v \oplus C_{v_1}) \oplus C_{v_2}) \oplus \ldots \oplus C_{v_n}) \tag{5}$$

является циклом одной ячейки карты блока, локализованной в цикле C_v.

Доказательство. Из теорем 5 и 6 имеем, что каждый из циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ является смежным циклу C_v и никакая пара циклов из $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ не имеет общих вершин. Значит, результатом вычисления по формуле (5) будет цикл C_v, из которого последовательно удаляются все циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ нулевой вложенности. Цикл S_v не может быть пустым, так как в этом случае будем иметь линейно зависимые циклы C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$, что противоречит их принадлежности DFS-базису циклов. Предположение, что остаток S_v может оказаться объединением двух или более ячеек карты, нарушит условие полноты циклов DFS-базиса. Значит, S_v есть цикл ячейки карты блока. Всё множество циклов ячеек карты блока $S = \{S_1, S_2, \ldots, S_{n_F}\}$ составляет базис циклов блока. Каждый цикл DFS-базиса C_j карты блока можно представить как линейную комбинацию $C_j = S_{j_1} \oplus S_{j_2} \oplus \ldots \oplus S_{j_k}$ циклов ячеек карты блока, которые охватывает данный цикл C_j. Размерность линейного пространства является инвариантом, следовательно, число циклов DFS-базиса и циклов ячеек карты совпадает. Отсюда заключаем, что все циклы S_v ячеек карты представляются по формуле (5), где каждый цикл C_v определяет свой цикл S_v ячейки карты. ■

4. Вычисление циклов ячеек карты графа

Предлагаемый алгоритм вычисления циклов ячеек карты блока реализует их расчёт по формуле (5). Справедливость данной формулы ограничена рамками карты блока. При разложении исходного графа на блоки можно воспользоваться алгоритмом,

представленным в [8]. Формирование циклов DFS-базиса и вычисление циклов ячеек карты блока по формуле (5) представлены в алгоритмах 1 и 2.

Алгоритм 1. Поиск в глубину циклов ячеек блока карты графа

1: $i = 0$; // *Счетчик циклов C_i DFS-базиса*
2: $j = 0$; // *Индекс верха стека циклов S_j и стеков вершин ev_j, ex_j обратных ребер*
3: $m = 0$; // *Число циклов S_j ячеек карты в архиве циклов T_m*
4: **Для всех** $v \in X$
5: $Mark[v] = 0$; // *Начальные метки вершин*
6: $v =$ `MinOY`(); // *Поиск вершины v с минимальной координатой по оси OY*
7: `Sort1`(v); // *Сортировка по полярному углу ребер, инцидентных вершине v*
8: $count = 0$; // *Счётчик меток пройденных вершин*
9: $nC = 1$; // *Число вершин в стеке C пройденных вершин*
10: $C[nC] = v$; // *Добавить вершину v в стек C пройденных вершин*
11: `Cycle`($v, 0$); // *Рекурсивный проход в глубину, 0 — фиктивная вершина*
12: $nC = nC - 1$; // *Удалить вершину из стека C пройденных вершин*
13: **ПРОЦЕДУРА** `Cycle`(x, y) // *Рекурсивный проход в глубину*
14: // *x — текущая вершина, y — вершина, из которой пришли в x*
15: $count = count + 1$; $Mark[x] = count$. // *Вершина x пройдена*
16: **Для всех** $v \in Adj[x]$
17: $nC = nC + 1$; $C[nC] = v$. // *Добавить вершину v в стек вершин C*
18: **Если** $Mark[v] = 0$, **то**
19: // *Вершина v не пройдена*
20: `Sort2`(v, x); // *Сортировка по полярному углу относительно ребра (x, v)*
21: `Cycle`(v, x); // *Рекурсивный проход в глубину по ребру (x, v)*
22: **иначе если** $Mark[v] < Mark[x] \wedge v \neq y$, **то**
23: // *(x, v) — обратное ребро*
24: $i = i + 1$; // *Номер нового цикла C_i DFS-базиса*
25: `CopyCycle`(C_i); // *Копировать C_i из стека вершин C*
26: **Если** $j > 0$, **то**
27: // *Удалить из цикла S_j, вложенный в него цикл C_i*
28: $S_j = S_j \oplus C_i$;
29: // *(x, v) — вершины обратного ребра цикла сохранить в стеках ev_j и ex_j*
30: $j = j + 1$; // *Индекс верха стеков S_j, ev_j и ex_j*
31: $ev_j = v$; // *v — первая вершина цикла*
32: $ex_j = x$; // *x — последняя вершина цикла*
33: $S_j = C_i$; // *Начать формирование нового цикла S_j ячейки карты*
34: **иначе если** $(x = ev_j) \wedge (v = ex_j)$, **то**
35: // *Вернуться по дереву поиска в начало цикла S_j ячейки карты*
36: // *Вершина x — начало цикла S_j, вершина v — конец цикла*
37: // *Цикл S_j ячейки карты сформирован, сохранить цикл в архиве T*
38: $m = m + 1$; // *Увеличить размер архива T циклов S_j ячеек карты*
39: $T_m = S_j$; // *Сохранить цикл S_j ячейки карты в архиве T*
40: $j = j - 1$; // *Сформированный цикл S_j удаляем из стека*
41: $nC = nC - 1$. // *Удалить вершину из стека C при возврате по дереву поиска*

Алгоритм 2. Копировать из стека C вершины цикла C_i

1: **ПРОЦЕДУРА** `CopyCycle`(C_i)
2: $k = nC$; // *Верх стека вершин* C
3: $w = C[k]$; // *Вершина* w *— начало цикла по обратному ребру*
4: $n = 1$; // *Индекс упаковки вершин цикла* C_i
5: **Повторять**
6: $k = k - 1$; $C_i[n] = C[k]$; $n = n + 1$;
7: **Пока** $C[k] = w$.

Карта блока исходного графа $G(X, U)$ задаётся структурой смежности $Adj[x]$, где $Adj[x]$ обозначает множество вершин, смежных с вершиной $x \in X$. Чтобы отличить пройденные вершины от непройденных, вводится вектор $Mark[x]$ меток вершин, которые постепенно нумеруются от 1 до $|X|$ по мере того, как попадаем в них.

Пройденные по рёбрам карты вершины сохраняются в стеке C пройденных вершин, например, $C[nC] = v$, где v — пройденная вершина, nC — индекс верха стека. Когда попадаем на обратное ребро (x, v), обнаруженный цикл C_i базиса будет состоять из этого ребра и рёбер, соединяющих вершины из верха стека до вершины v обратного ребра в стеке, где x — последняя и v — первая вершины цикла. Процедура `CopyCycle`(C_i) (алгоритм 2) копирует вершины цикла из стека C в цикл C_i базиса.

Циклы S_j ячеек карты блока во время их вычисления размещаются в стеке циклов ячеек карты, где j — индекс верха стека. Вычислению по формуле (5) подлежит текущий цикл S_j, после вычисления он копируется в архив T_m и удаляется из стека.

Каждый новый сформированный цикл C_i базиса будет циклом нулевой вложенности (если это не огибающий цикл блока) в текущем вычисляемом цикле S_j ячейки карты, из которого его исключаем ($S_j = S_j \oplus C_i$) по формуле (5).

С другой стороны, новый цикл C_i базиса определяет и новый цикл S_{j+1} ячейки карты блока (см. теорему 7). Начальное значение устанавливается равным $S_{j+1} = C_i$. Данный цикл размещается на верху стека ячеек карты и далее уже в этом цикле S_{j+1} выполняется поиск циклов базиса нулевой вложенности.

Определим условие, которое фиксирует, что все циклы базиса нулевой вложенности в цикл S_j (начальное значение $S_j = C_i$) исключены из S_j по формуле (5). Формирование цикла C_i базиса заканчивается обратным ребром (x, v), где v — первая и x — последняя вершины цикла. Вершины v и x обратных рёбер циклов C_i сохраняются в своих стеках $ev_j = v$ и $ex_j = x$, где ev_j — стек первых вершин циклов; ex_j — стек последних вершин циклов; j — индекс верха стеков. Индекс j является индексом верха и для стека циклов S_j ячеек.

При поиске циклов нулевой вложенности в цикле C_i мы возвращаемся от последней вершины x по ребрам этого цикла к его начальной вершине v, найденной по обратному ребру (x, v) цикла. Пусть мы вернулись в начальную вершину цикла. Текущая рассматриваемая вершина в алгоритме 1 обозначается через x, а исследуемое ребро — через (x, v). Пусть далее вершина x является начальной вершиной цикла C_i. Все рёбра (x, v), инцидентные вершине x и локализованные в границах цикла C_i, будут пройдены как обратные для вложенных циклов. Для продолжения обхода из вершины x выполняется перебор инцидентных ей рёбер (x, v). Инцидентные рёбра вершин сортированы по полярному углу (см. правила 1 и 2). Поэтому перебор начнётся с рёбер (x, v), инцидентных вершине x и локализованных в границах цикла C_i. Для выделения об-

ратного ребра выполняется проверка условия $(x = ev_j) \wedge (v = ex_j)$, которое является признаком окончания вычисления текущего цикла ячейки S_j карты блока, так как все циклы нулевой вложенности в цикле C_i выделены.

Описание алгоритма 1. Начальная вершина $v = \texttt{MinOY}()$ прохода по графу — это вершина с минимальной координатой по оси OY, что гарантирует принадлежность вершины v огибающему циклу карты блока (см. теорему 4). В соответствии с правилом 1 выполняется сортировка $\texttt{Sort1}(v)$ рёбер, инцидентных вершине v, в порядке возрастания их полярных углов. В этом порядке располагаются соответствующие рёбрам вершины в структуре смежности $Adj[v]$ для вершины v. Поэтому перебор рёбер, инцидентных вершине v, выполняется против часовой стрелки в порядке роста их полярных углов. Выбранная вершина v первой помещается в стек пройденных вершин ($nC = 1$; $C[nC] = v$). Далее выполняется первый вызов рекурсивной процедуры $\texttt{Cycle}(v, 0)$ обхода графа в глубину с двумя параметрами: v — текущая выбранная вершина для просмотра; 0 — фиктивная вершина при первом вызове.

ПРОЦЕДУРА $\texttt{Cycle}(x, y)$ — заголовок описания рекурсивной процедуры обхода карты блока в глубину, где x — текущая обрабатываемая вершина; y — вершина, из которой пришли в вершину x по ребру (y, x). Далее выполним разбор ключевых операторов данной процедуры.

Разбор цикла: Для всех $v \in Adj[x]$. В данном цикле из текущей вершины x пытаемся выполнить проход в глубину по ребру (x, v). Для вершины v возможны три случая.

С л у ч а й 1: «**Если** $Mark[v] = 0$**, то**». Имеем вершину v, которая не пройдена. Рекурсивно выполняем проход в глубину $\texttt{Cycle}(v, x)$ с новой текущей вершиной v, где (x, v) — ребро, по которому пришли в v. Предварительно, в соответствии с правилом 2, выполняется сортировка $\texttt{Sort2}(v, x)$ рёбер, инцидентных вершине v, в порядке возрастания полярных углов относительно ребра (x, v) против часовой стрелки. В соответствующем порядке располагаются вершины в структуре смежности $Adj[v]$.

С л у ч а й 2: «**иначе Если** $Mark[v] < Mark[x] \wedge v \neq y$**, то**». Условие фиксирует, что вершина v была пройдена и мы приходим в неё по обратному ребру (x, v) цикла, так как $v \neq y$. Вершины v и x обратного ребра найденного цикла сохраняем в соответствующих стеках $ev_j = v$ и $ex_j = x$, где v — вершина начала цикла; x — последняя вершина цикла. Сформированный цикл из пройденных вершин в стеке C копируем в очередной цикл C_i базиса.

Новый цикл базиса C_i является циклом нулевой вложенности в текущем вычисляемом цикле S_j ячейки карты блока (см. теорему 5), из которого его исключаем ($S_j = S_j \oplus C_i$) по формуле (5).

Новый цикл C_i базиса определяет и новый цикл S_{j+1} ячейки карты блока (см. теорему 7). Начальное значение устанавливается равным $S_{j+1} = C_i$. Данный цикл размещается на верху стека циклов ячеек карты и далее уже в этом цикле S_{j+1} выполняется поиск циклов базиса нулевой вложенности.

С л у ч а й 3: «**иначе Если** $(x = ev_j) \wedge (v = ex_j)$**, то**». Здесь (x, v) — ребро прохода по карте блока из вершины x в вершину v. Ребро (ex_j, ev_j) — обратное ребро текущего вычисленного цикла S_j ячейки карты, где ev_j — вершина начала цикла; ex_j — вершина конца цикла. Выражение $(x = ev_j) \wedge (v = ex_j)$ означает, что в результате прохода по графу мы вернулись в вершину x и, продолжив перебор смежных ей вершин, выбрали ребро (x, v), которое является обратным (ex_j, ev_j) для цикла S_j. Отсюда заключаем, что цикл S_j ячейки карты сформирован. Цикл копируется в архив ($T_m = S_j$) и удаляется из стека циклов ячеек карты посредством операции $j = j - 1$. Это значит, что

возвращаемся к формированию оставшегося на верху стека цикла S_j ячейки карты блока.

Сложность алгоритма 1 оценим относительно числа вершин в графе. Пусть $G = (X, U)$ — исходная карта блока. В простом планарном графе выполняется соотношение $|U| \leqslant 3|X| - 6$ (см. п. 2). Пусть $n = |X|$, тогда $|U| \leqslant 3|X| - 6 = 3n - 6 = O(n)$. Оценим сложность в худшем случае, когда $|U| = O(n)$.

Сложность начальной сортировки $\texttt{Sort1}(v)$ вершин, смежных вершине v, можно считать равной $O(n \log_2 n)$. Сложность поиска $v = \texttt{MinOY}()$ является линейной — $O(n)$.

Сложность выполнения процедуры $\texttt{Cycle}(x, y)$ определяется сложностью выполнения цикла «**Для всех** $v \in Adj[x]$». Каждое ребро (x, v) просматривается дважды. Пусть n_i — число вершин, смежных вершине x_i, тогда $\sum\limits_{i=1}^{n} n_i = 2|U| = O(n)$. Таким образом, цикл выполняется $O(n)$ раз.

Для каждой вершины один раз в цикле выполняется сортировка $\texttt{Sort2}(v, x)$ смежных ей вершин. Полагаем, что сортировка n_i вершин имеет сложность $C_i n_i \log_2 n_i$, где C_i — положительная константа. Суммарная сложность сортировки составит $\sum\limits_{i=1}^{n} C_i n_i \log_2 n_i \leqslant C \sum\limits_{i=1}^{n} n_i \log_2 n_i \leqslant C \log_2 n \sum\limits_{i=1}^{n} n_i \leqslant C(\log_2 n) O(n) = O(n \log_2 n)$, где C — константа, максимальная из всех C_i.

Сложности операций копирования $\texttt{CopyCycle}(C_i)$ и слияния $S_j = S_j \oplus C_i$ являются линейными относительно длин циклов. Длины циклов не превышают числа рёбер $|U| = O(n)$. Суммарная сложность операций копирования не превышает величины $\sum\limits_{i=1}^{n} n_i |U| = |U| \sum\limits_{i=1}^{n} n_i \leqslant 2|U||U| = 2O(n)O(n) = O(n^2)$. Отсюда заключаем, что сложность алгоритма 1 составляет $O(n^2)$.

Заключение

Рассмотренный алгоритм вычисления циклов ячеек карты блока графа является расширением классического алгоритма поиска в глубину циклов DFS-базиса. Ключевой идеей модификации классического алгоритма является предложенная стратегия «правого обхода» при прохождении графа в глубину. Правый обход при поиске в глубину позволил для каждого цикла DFS-базиса последовательно выделить и исключить из него все циклы нулевой вложенности. Показано, что оставшаяся часть базисного цикла является циклом ячейки карты блока. Сложность каждого шага алгоритма не превышает квадратичной относительно числа вершин в графе.

ЛИТЕРАТУРА

1. *Оре О.* Теория графов. М.: Наука, 1980. 336 с.
2. *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение. М.: Мир, 1989. 478 с.
3. *Welch J.* A mechanical analysis of the cyclic structure of undirected linear graphs // J. Assoc. Comput. Mech. 1966. V. 13. P. 205–210.
4. *Gibbs N. W.* A cycle generation algorithm for finite undirected linear graphs // J. Assoc. Comput. Mech. 1969. V. 16. P. 564–568.
5. *Tarjan R.* Enumeration of the elementary circuits of a directed graph // SIAM J. Comput. 1973. V. 2. P. 211–216.
6. *Jonson D. B.* Finding all the elementary circuits of a directed graph // SIAM J. Comput. 1975. V. 4. P. 77–84.
7. *Mateti P. and Deo N.* On algorithms for enumerating all circuits of a graph // SIAM J. Comput. 1976. V. 5. No. 1. P. 90–99.

8. *Tarjan R.* Depth-first search linear graph algorithms // SIAM J. Comput. 1972. V. 1. P. 146–160.
9. *Mahdi F., Safar M., and Mahdi K.* Detecting cycles in graphs using parallel capabilities of GPU // Digital Inform. Commun. Techn. Appl. (DISTAP). 2011. V. 167. P. 193–205.
10. *Kavitha T., Mehlhorn K., and Michail D.* New approximation algorithms for minimum cycle bases of graphs // J. Algorithmica. 2011. V. 59. No. 4. P. 471–488.
11. *Pfaltz J. L.* Chordless cycles in networks // IEEE 29th Intern. Conf. ICDEW. 2013. P. 223–228.
12. *Иванов Б. Н.* Генерация циклов ячеек карты простого планарного графа // Вычислительные методы и программирование. 2014. № 15. С. 304–316.
13. *Алексеев В. Е., Таланов В. А.* Графы. Модели вычислений. Структуры. Н. Новгород: Изд-во ННГУ, 2005. 307 с.
14. *Paton K.* An algorithm for finding a fundamental set of cycles of a graph // Comm. ACM. 1969. V. 12. No. 9. P. 514–518.
15. *Рейнголд Э., Нивергельд Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980. 476 с.

REFERENCES

1. *Ore O.* Theory of Graphs. AMS Press, Providence, 1962. 336 p.
2. *Preparatat F. P. and Shamos M. I.* Computational Geometry: An Introduction. N.Y., Springer Verlag, 1989. 478 p.
3. *Welch J.* A mechanical analysis of the cyclic structure of undirected linear graphs. J. Assoc. Comput. Mech., 1966, vol. 13, pp. 205–210.
4. *Gibbs N. W.* A cycle generation algorithm for finite undirected linear graphs. J. Assoc. Comput. Mech., 1969, vol. 16, pp. 564–568.
5. *Tarjan R.* Enumeration of the elementary circuits of a directed graph. SIAM J. Comput., 1973, vol. 2, pp. 211–216.
6. *Jonson D. B.* Finding all the elementary circuits of a directed graph. SIAM J. Comput., 1975, vol. 4. pp. 77–84.
7. *Mateti P. and Deo N.* On Algorithms for enumerating all circuits of a graph. SIAM J. Comput., 1976, vol. 5. no. 1, pp. 90–99.
8. *Tarjan R.* Depth-first search linear graph algorithms. SIAM J. Comput., 1972, vol. 1, pp. 146–160.
9. *Mahdi F., Safar M., and Mahdi K.* Detecting cycles in graphs using parallel capabilities of GPU. Digital Inform. Commun. Techn. Appl. (DISTAP), 2011, vol. 167, pp. 193–205.
10. *Kavitha T., Mehlhorn K., and Michail D.* New approximation algorithms for minimum cycle bases of graphs. Algorithmica, 2011, vol. 59, no. 4, pp. 471–488.
11. *Pfaltz J. L.* Chordless cycles in networks. IEEE 29th Intern. Conf. ICDEW, 2013, pp. 223–228.
12. *Ivanov B. N.* Generatsiya tsiklov yacheek karty prostogo planarnogo grafa [Generation of cycles of map cells for a simple planar graph]. Vychislitel'nye Metody i Programmirovanie, 2014, no. 15, pp. 304–316. (in Russian)
13. *Alekseev B. E. and Talanov V. A.* Grafy. Modeli vychisleniy. Struktury [Graphs. Computational Models. Structures]. N. Novgorod, Lobachevsky State University, 2005. 307 p. (in Russian)
14. *Paton K.* An algorithm for finding a fundamental set of cycles of a graph. Comm. ACM, 1969, vol. 12, no. 9, pp. 514–518.
15. *Reingold E. M., Nievergelt J., and Deo N.* Combinatorial Algorithms. Theory and Practice. Prentice-Hall, Englewood Cliff, 1977. 476 p.

УДК 519.17 DOI 10.17223/20710410/58/8

О ЕДИНСТВЕННОСТИ МИНИМАЛЬНОГО РЁБЕРНОГО 1-РАСШИРЕНИЯ ГИПЕРКУБА Q_4[1]

А. А. Лобов, М. Б. Абросимов

Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, г. Саратов, Россия

E-mail: aisanekai@mail.ru, mic@rambler.ru

Одним из важных свойств надёжных вычислительных систем является их отказоустойчивость. Для исследования отказоустойчивости можно использовать аппарат теории графов. Рассматриваются минимальные рёберные расширения графа, которые являются моделью для исследования отказа связей вычислительной системы. Граф $G^* = (V^*, \alpha^*)$ с n вершинами называется минимальным рёберным k-расширением n-вершинного графа $G = (V, \alpha)$, если граф G вкладывается в каждый граф, получающийся из G^* удалением любых его k рёбер и имеет при этом минимально возможное число рёбер. Гиперкуб Q_n — это регулярный 2^n-вершинный граф порядка n, представляющий собой декартово произведение n полных 2-вершинных графов K_2. Гиперкуб является распространённой топологией для построения вычислительных систем. Ранее было описано семейство графов Q_n^*, представители которого при $n > 1$ являются минимальными рёберными 1-расширениями соответствующих гиперкубов. В данной работе получено аналитическое доказательство единственности минимальных рёберных 1-расширений гиперкубов при $n \leqslant 4$ и установлено общее свойство произвольного минимального рёберного 1-расширения гиперкуба Q_n при $n > 2$: оно не содержит рёбер, соединяющих вершины, расстояние между которыми в гиперкубе равно 2.

Ключевые слова: *граф, гиперкуб, рёберная отказоустойчивость, минимальное рёберное 1-расширение.*

ABOUT UNIQUENESS OF THE MINIMAL 1-EDGE EXTENSION OF HYPERCUBE Q_4

A. A. Lobov, M. B. Abrosimov

Saratov State University, Saratov, Russia

One of the important properties of reliable computing systems is their fault tolerance. To study fault tolerance, we can use the graph theory. In this paper, minimal edge extensions of graph are considered as a model for studying the failure of links in a computing system. A graph G^* is a k-edge extension of a graph G if every graph obtained by removing any k edges from G^* contains G. A k-edge extention G^* of graph G is said to be minimal if it contains n vertices, where n is the number of vertices of G, and G^* has the minimum number of edges among all k-edge extensions of graph G with n vertices. The hypercube Q_n is a regular 2^n-vertex graph of order n, which is the Cartesian product of n complete 2-vertex graphs K_2. We propose a

[1]Работа выполнена при поддержке Минобрнауки России в рамках выполнения госзадания (проект № FSRR-2020-0006).

family of graphs Q_n^* whose representatives for $n > 1$ are minimal 1-edge extensions of the corresponding hypercubes. In this paper, we provide an analytical proof of the uniqueness of minimal 1-edge extensions of hypercubes for $n \leqslant 4$, as well as establishing one general property of an arbitrary minimal 1-edge extension of a hypercube Q_n for $n > 2$: it does not contain edges connecting vertices, the distance between which in the hypercube is equal to 2.

Keywords: *graph, hypercube, edge fault tolerance, minimal 1-edge extension.*

Введение

Топология гиперкуба является популярной схемой соединения параллельных процессоров [1], в том числе и в отказоустойчивых системах типа IBM Blue Gene/Q [2]. Особый интерес с точки зрения отказоустойчивости представляет 4-куб Q_4. Для исследования полной отказоустойчивости элементов Дж. Хейз предложил модель, основанную на графах [3, 4], которая позднее была перенесена и на случай отказов связей [5]. Оптимальные вершинные k-отказоустойчивые реализации гиперкубов Q_n при $n > 3$ неизвестны. На практике используются тривиальные отказоустойчивые реализации с одним избыточным элементом, соединённым со всеми остальными. В работе [6] описывается вершинное 1-расширение гиперкуба Q_4, которое имеет на одно ребро меньше, чем тривиальное. Однако не доказано, что оно является минимальным. Ранее была предложена оптимальная рёберная 1-отказоустойчивая реализация гиперкуба Q_n при $n \geqslant 2$ [5]. В данной работе удалось доказать, что при $n \leqslant 4$ гиперкуб Q_n имеет единственную с точностью до изоморфизма оптимальную рёберную 1-отказоустойчивую реализацию или, в другой терминологии, минимальное рёберное 1-расширение. Отметим, что существуют и другие подходы для исследования отказоустойчивости [7, 8]. Мы будем использовать основные определения из теории графов в соответствии с работами [9, 10].

Определение 1. *Декартовым произведением* $G_1 \times G_2$ двух графов $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$ называется граф $G = (V, \alpha)$, такой, что $V = V_1 \times V_2$ и вершины (u_1, u_2) и (v_1, v_2) смежны в G тогда и только тогда, когда либо $u_1 = v_1$, а u_2 и v_2 смежны в G_2, либо $u_2 = v_2$, а u_1 и v_1 смежны в G_1.

Определение 2. *Гиперкубом* Q_n (*n-кубом*) называется граф, являющийся декартовым произведением n полных 2-вершинных графов K_2.

Гиперкуб Q_n — это регулярный 2^n-вершинный граф порядка n. Семейство гиперкубов достаточно хорошо изучено [11]. Важным свойством гиперкуба является его симметричность: все вершины и рёбра в гиперкубе являются подобными.

Введём традиционное обозначение меток вершин гиперкуба Q_n двоичными векторами длины n так, чтобы расстояние между двумя вершинами равнялось дистанции Хэмминга между их метками. На рис. 1 показаны гиперкубы Q_1, Q_2, Q_3 и Q_4.

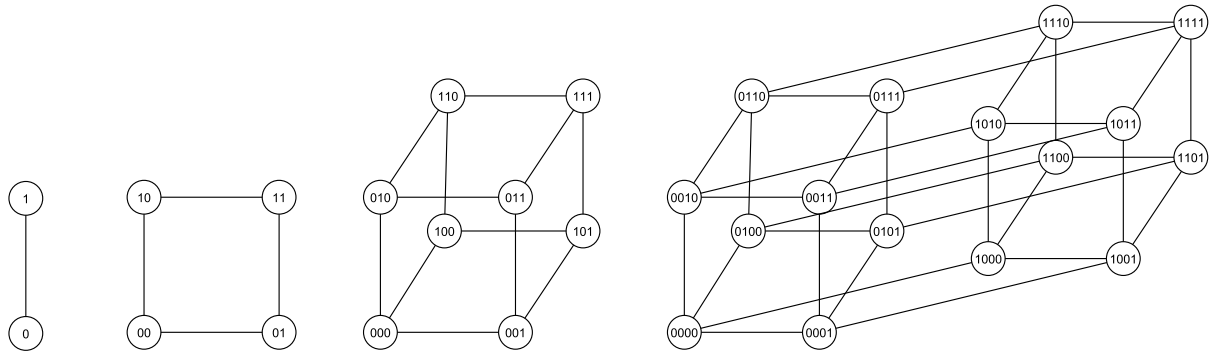

Рис. 1. Гиперкубы Q_1, Q_2, Q_3 и Q_4

1. Минимальные рёберные 1-расширения гиперкуба

Определение 3. Граф $G^* = (V^*, \alpha^*)$ называется *минимальным рёберным k-расширением* n-вершинного графа $G = (V, \alpha)$, если выполняются следующие условия:

1) граф G^* является рёберным k-расширением графа G, то есть граф G вкладывается в каждый граф, получающийся из G^* удалением любых его k рёбер;
2) граф G^* содержит n вершин, то есть $|V^*| = |V|$;
3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Если рассматривать простые графы, то минимальное рёберное k-расширение существует не у всех графов. Например, полные графы K_n не имеют минимальных ребёрных k-расширений ни при каких натуральных k. Задача поиска минимальных рёберных k-расширений является вычислительно сложной [12], только для некоторых классов графов удалось найти частичное или полное описание схем построения их минимальных рёберных k-расширений. В работах [13, 14] предложены алгоритмы построения всех неизоморфных минимальных вершинных и рёберных расширений заданного графа без непосредственной проверки на изоморфизм, что хорошо подходит для параллельной реализации. С помощью этих алгоритмов удалось экспериментально найти все минимальные рёберные 1-расширения 16-вершинного гиперкуба, однако минимальные вершинные 1-расширения пока найти не удалось, так как для их поиска требуется на несколько порядков больше вычислительных ресурсов. Предположительно минимальное вершинное 1-расширение найдено аналитическим путём, однако его минимальность пока не подтверждена ни аналитически, ни с помощью вычислительного эксперимента [6].

В работе [2] доказаны несколько лемм, которые позволяют охарактеризовать вид минимального рёберного k-расширения и оценить минимально возможное количество дополнительных рёбер в нём. Далее нам понадобится одна из них.

Лемма 1 [2]. Если минимальная степень вершины графа G равна $d > 0$, то его минимальное рёберное k-расширение не содержит вершин степени меньше $d + k$.

Будем рассматривать минимальные рёберные 1-расширения гиперкуба Q_n. При $n = 1$ гиперкуб Q_1 изоморфен полному графу K_2 и не имеет минимальных рёберных 1-расширений. При $n = 2$ гиперкуб Q_2 изоморфен циклу C_4 и по лемме 1 имеет единственное с точностью до изоморфизма минимальное рёберное 1-расширение — полный граф K_4. Далее рассматриваем случай $n > 2$.

В [5] предложена схема построения графов Q_n^*. Это однородные графы порядка $n + 1$, которые при $n > 1$ получаются из гиперкуба Q_n добавлением рёбер, соединяющих противоположные вершины, то есть вершины, расположенные друг от друга на

максимальном расстоянии n. Если вершина имеет код k, то она соединяется с вершиной, код которой получается из k поразрядной инверсией.

Теорема 1. Для n-мерного гиперкуба Q_n при $n > 1$ граф Q_n^* является минимальным рёберным 1-расширением.

Легко убедиться, что минимальное рёберное 1-расширение для гиперкуба Q_2 единственно с точностью до изоморфизма: граф Q_2^* изоморфен графу K_4, и это единственный с точностью до изоморфизма регулярный 4-вершинный граф порядка 3. В работе [5] вопрос единственности минимального рёберного 1-расширения не обсуждается. В гиперкубе Q_n каждая вершина имеет степень n, а в его минимальном рёберном 1-расширении Q_n^* каждая вершина имеет степень $n+1$. Следовательно, если у гиперкуба Q_n существуют другие минимальные рёберные 1-расширения, то все вершины в них также имеют степень $n+1$. Докажем, что гиперкубы Q_3 и Q_4 имеют единственные с точностью до изоморфизма минимальные рёберные 1-расширения, что является основным результатом данной работы.

2. Единственность минимального рёберного 1-расширения гиперкуба Q_3

Для произвольного минимального рёберного 1-расширения гиперкуба Q_n при $n > 2$ установлено следующее общее свойство:

Лемма 2. Минимальное рёберное 1-расширение гиперкуба Q_n при $n > 2$ не содержит рёбер, соединяющих вершины, расстояние между которыми в гиперкубе Q_n равно 2.

Доказательство. Предположим, что G^* является минимальным рёберным 1-расширением гиперкуба Q_n. Граф G^* строится из графа Q_n путём добавления только одного ребра в каждой вершине, то есть разбиения множества несмежных вершин графа Q_n на пары и соединения их рёбрами. Предположим, что в графе G^* есть ребро, соединяющее вершины, коды которых отличаются в двух позициях. Не ограничивая общности, пусть это будут, например, вершины $0\ldots000$ и $0\ldots011$.

В графе Q_n все вершины и рёбра являются подобными. Очевидно, что удаление добавленного ребра в графе G^* даст граф, в который будет вкладываться исходный гиперкуб Q_n.

Рассмотрим удаление любого ребра, инцидентного вершине $0\ldots00$ и вершине, несмежной с $0\ldots011$, например $10\ldots0$. Обозначим получившийся граф через H. Попробуем вложить в граф H гиперкуб Q_n. Вершина $0\ldots00$ в графе H имеет степень n, следовательно, на расстоянии 1 от неё в гиперкубе будут все оставшиеся вершины, кроме $10\ldots0$: $010\ldots0$, $0010\ldots0$, …, $0\ldots010$, $0\ldots001$ и $0\ldots011$. При вложении гиперкуба между вершинами на расстоянии 1 нет рёбер, то есть рёбра между вершиной $0\ldots011$ и вершинами $0\ldots010$ и $0\ldots001$ не могут быть задействованы при вложении. Но без учёта этих рёбер у вершины $0\ldots011$ степень станет равна $n-1$ и вложение гиперкуба будет невозможно. ■

Из леммы 2 следует

Теорема 2. Граф Q_3^* является единственным с точностью до изоморфизма минимальным рёберным 1-расширением гиперкуба Q_3.

Единственные с точностью до изоморфизма минимальные рёберные 1-расширения гиперкубов Q_2 и Q_3 представлены на рис. 2.

Аналогичный результат удалось доказать и для гиперкуба Q_4, что является основным результатом данной работы.

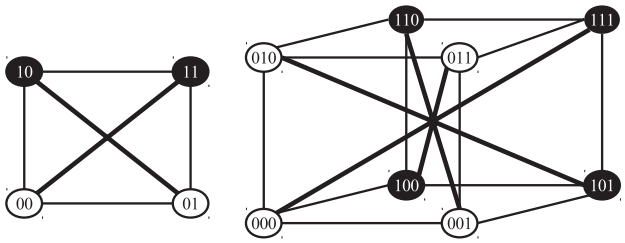

Рис. 2. Минимальные рёберные 1-расширения для Q_2 и Q_3

3. Единственность минимального рёберного 1-расширения гиперкуба Q_4

Теорема 3. Граф Q_4^* является единственным с точностью до изоморфизма минимальным рёберным 1-расширением гиперкуба Q_4.

Доказательство. Предположим, что G^* является минимальным рёберным 1-расширением гиперкуба Q_4, причём G^* не изоморфен Q_4^*. Как отмечено в доказательстве леммы 2, граф G^* строится из графа Q_4 путём добавления только одного ребра в каждой вершине, то есть разбиения множества несмежных вершин графа Q_4 на пары и соединения их рёбрами.

С учётом леммы 2 достаточно показать, что минимальное рёберное 1-расширение гиперкуба Q_4 не содержит рёбер, соединяющих вершины, расстояние между которыми в гиперкубе Q_4 равно 3.

Предположим, что это неверно и в графе G^* есть ребро, соединяющее вершины, код которых отличается в трёх позициях. Не ограничивая общности, пусть это будут, например, вершины 0000 и 0111.

Далее будем последовательно рассматривать удаление рёбер, инцидентных вершине 0000, и пробовать строить вложение гиперкуба, доопределяя по необходимости нужные рёбра в графе G^*. Будем строить вложение по мере удаления вершин от вершины 0000. Заметим, что в гиперкубе должно быть четыре вершины на расстоянии 1 от вершины 0000, шесть вершин на расстоянии 2, четыре вершины на расстоянии 3 и одна вершина на расстоянии 4. Расстояние до вершины 0000 равно сумме единиц в коде вершины. Каждая вершина на расстоянии 1 смежна с вершиной 0000 и тремя вершинами на расстоянии 2, каждая вершина на расстоянии 2 смежна с двумя вершинами на расстоянии 1 и с двумя вершинами на расстоянии 3, каждая вершина на расстоянии 3 смежна с вершиной на расстоянии 4 и тремя вершинами на расстоянии 2. Так как в графе G^* все вершины имеют степень 5, то при вложении только одно ребро в каждой вершине не будет задействовано.

Будем называть вершину на расстоянии t от вершины 0000 вершиной типа t. Тогда в гиперкубе Q_4 есть одна вершина типа 0 (это сама вершина 0000), четыре вершины типа 1 (0001, 0010, 0100 и 1000), шесть вершин типа 2 (0011, 0101, 0110, 1001, 1010 и 1100), четыре вершины типа 3 (0111, 1011, 1101 и 1110) и одна вершина типа 4 (1111).

На рис. 3–8 белым цветом помечены вершина 0000 и вершины, тип которых неизвестен. Вершины типа 1 помечены белым цветом с жирной границей, рёбра от них до вершины 0000 также выделены жирным. Удалённое ребро обозначено пунктиром, а добавленные рёбра — жирным пунктиром. Светло-серым цветом обозначены вершины типа 2, а тёмно-серым — вершины типа 3. Черным цветом обозначена вершина типа 4.

С л у ч а й 1. Рассмотрим удаление ребра между вершинами 0000 и 0100 в графе G^*. Предположим, что возможно вложение гиперкуба Q_4 в получившийся граф.

Так как у вершины 0000 после удаления ребра степень будет равна 4, то все смежные с ней вершины при вложении гиперкуба окажутся от неё на расстоянии 1: 0001, 0010, 1000 и 0111. Так как вершины 0110, 0011, 0101, 1010 и 1001 смежны, по крайней мере, с двумя из этих вершин, то при вложении гиперкуба они будут находиться на расстоянии 2 от вершины 0000. Рассуждаем далее. Вершины 0100, 1110, 1011 и 1101 смежны, по крайней мере, с двумя вершинами на расстоянии 2, следовательно, это будут вершины на расстоянии 3. Таким образом, осталось только две вершины, роли которых пока не определены: 1100 и 1111. Одна из них должна быть вершиной на расстоянии 4, а одна — на расстоянии 2. Каждая из них смежна с тремя вершинами на расстоянии 3 и одной вершиной на расстоянии 1.

Заметим, что в гиперкубе для каждой пары вершин на расстоянии 1 есть единственная смежная с ними обеими вершина на расстоянии 2. Однако вершина 0111 имеет две общие смежные вершины типа 2 с вершиной 0010 (0110 и 0011) и две общие смежные вершины типа 2 с вершиной 0001 (0011 и 0101). Так как все рёбра вершины 0111 уже определены и при вложении гиперкуба только одно из них не будет использоваться, то это ребро 0111–0011. Следовательно, вершина 1111 будет также вершиной типа 2, а вершина 1100 — вершиной типа 4. Так как вершина типа 4 смежна со всеми вершинами типа 3, то должно быть ребро между вершинами 1100 и 1011. Далее заметим, что у вершин 0111 и 1000 должна быть общая смежная вершина типа 2. У вершины 0111 уже все рёбра определены, и это не может быть ни одна из вершин 0110, 0011 и 0101. Следовательно, это должна быть вершина 1111, поэтому должно быть ребро 1000–1111. Получили ситуацию, представленную на рис. 3.

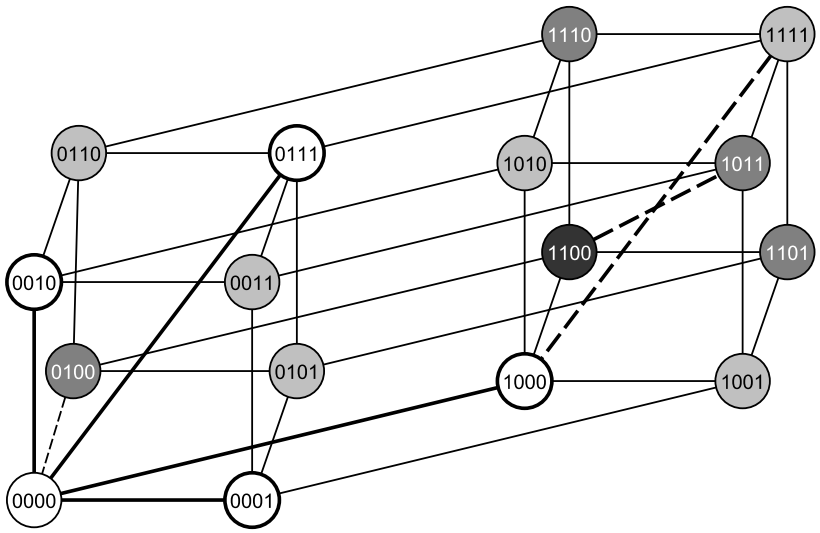

Рис. 3. Случай 1. Удаление ребра между вершинами 0000 и 0100

С л у ч а й 2. Рассмотрим теперь удаление ребра между вершинами 0000 и 0001. Повторяя те же рассуждения, получим, что вершины 0010, 0100, 0111 и 1000 будут вершинами типа 1, а 0110, 0011, 0101, 1100, 1010 — вершинами типа 2, так как смежны, по крайней мере, с двумя вершинами типа 1. С учётом уже определённого ребра между вершинами 1000 и 1111, получаем, что вершина 1111 также будет вершиной типа 2, так как смежна с двумя вершинами типа 1. Тогда вершины 0001, 1110, 1011 и 1101 будут вершинами типа 3, так как смежны, по крайней мере, с двумя вершинами типа 2. Оставшаяся вершина 1001 будет вершиной типа 4. Так как вершина типа 4 смежна

со всеми вершинами типа 3, то должно быть ребро между вершинами 1001 и 1110. Получили ситуацию, представленную на рис. 4.

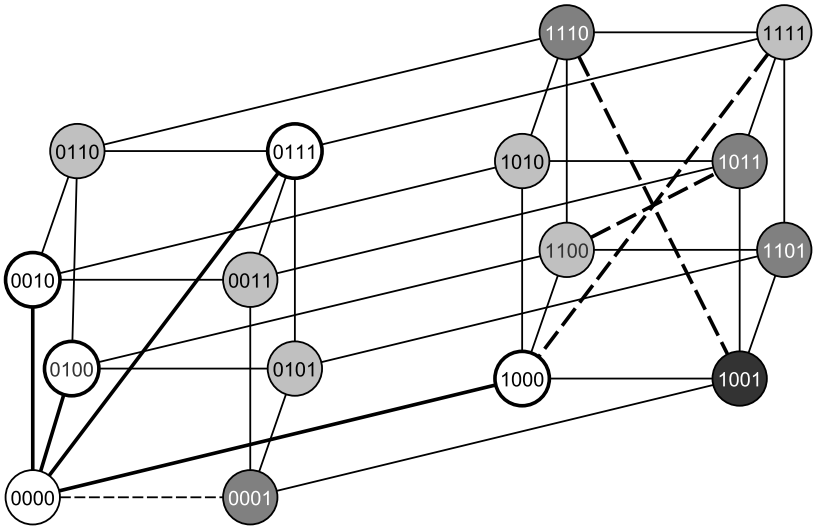

Рис. 4. Случай 2. Удаление ребра между вершинами 0000 и 0001

С л у ч а й 3. Рассмотрим удаление ребра между вершинами 0000 и 0010. Повторяя предыдущие рассуждения, получим, что вершины 0001, 0100, 0111 и 1000 будут вершинами типа 1, а 0110, 0011, 0101, 1001, 1100 и 1111 — вершинами типа 2, так как смежны, по крайней мере, с двумя вершинами типа 1. Тогда вершины 0010, 1110, 1011 и 1101 будут вершинами типа 3, так как смежны, по крайней мере, с двумя вершинами типа 2. Оставшаяся вершина 1010 будет вершиной типа 4. Так как вершина типа 4 смежна со всеми вершинами типа 3, то должно быть ребро между вершинами 1010 и 1101. Получили ситуацию, представленную на рис. 5.

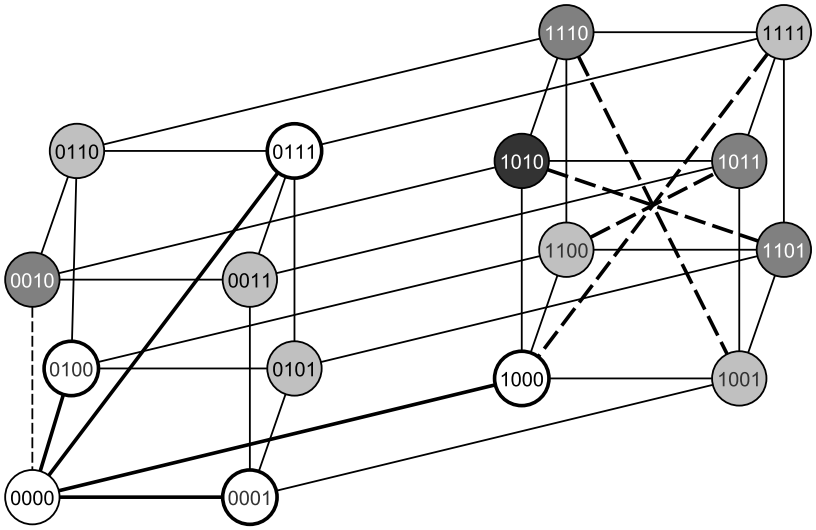

Рис. 5. Случай 3. Удаление ребра между вершинами 0000 и 0010

Заметим, что мы полностью определили дополнительные рёбра у десяти вершин графа G^*, и осталось только шесть вершин, относительно которых дополнительные

рёбра не определены: 0001, 0010, 0011, 0100, 0101 и 0110. Так как с учётом леммы 2 мы не можем соединять вершины на расстоянии 2, то остаётся единственным возможным образом соединить вершины на расстоянии 3 (все вершины на расстоянии 4 уже «заняты»). Получили, что граф G^* должен иметь вид, представленный на рис. 6.

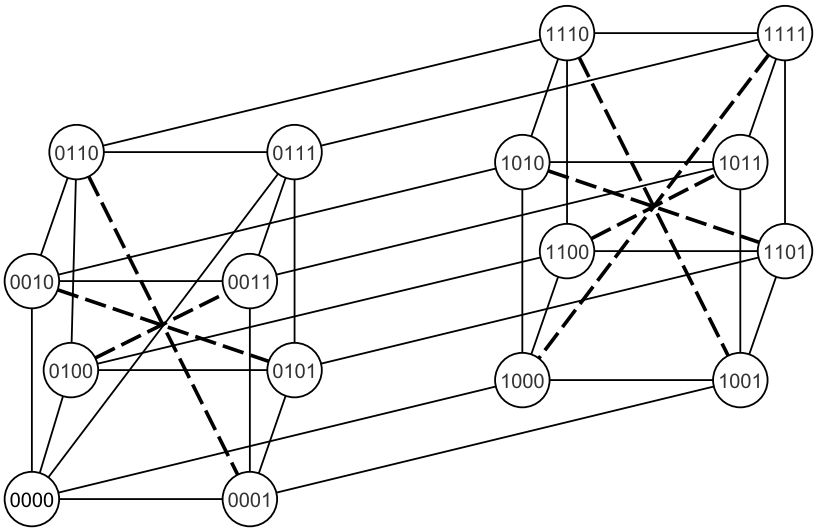

Рис. 6. Предполагаемое минимальное реберное 1-расширение гиперкуба Q_4

Убедимся, что построенный граф G^* не является расширением гиперкуба Q_4.

С л у ч а й 4. Рассмотрим удаление ребра между вершинами 0000 и 1000. Повторяем предыдущие рассуждения. Вершины 0001, 0010, 0100 и 0111 будут вершинами типа 1, а 0110, 0011, 0101 — вершинами типа 2, так как смежны, по крайней мере, с двумя вершинами типа 1 (рис. 7).

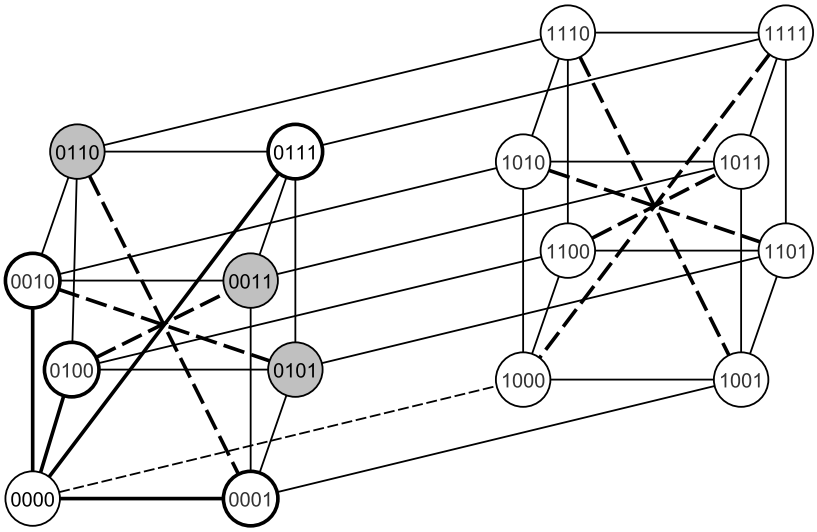

Рис. 7. Случай 4. Удаление ребра между вершинами 0000 и 1000

Видно, что вершина 0110 смежна с четырьмя вершинами типа 1 и ещё с вершиной 1110. Напомним, что вершина типа 2 должна быть смежна с двумя вершинами типа 1 и двумя вершинами типа 3. Следовательно, вложение гиперкуба невозможно. ■

Таким образом, гиперкуб Q_4 имеет единственное с точностью до изоморфизма минимальное рёберное 1-расширение, которое представлено на рис. 8.

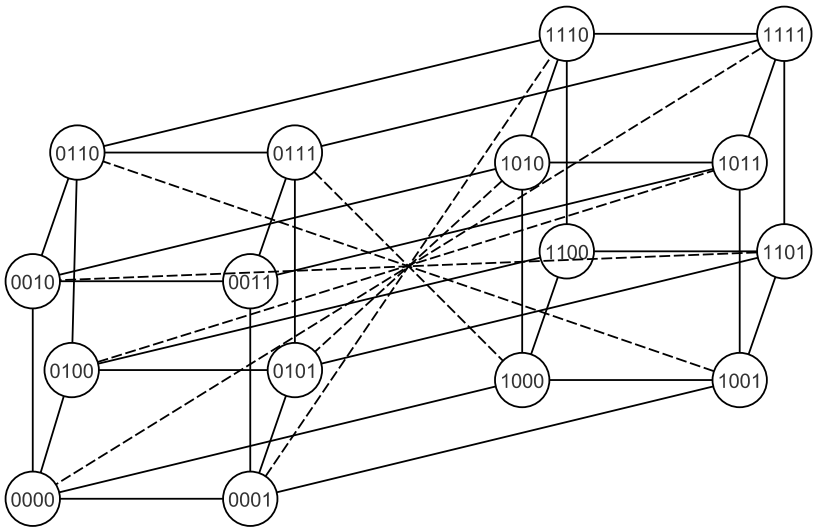

Рис. 8. Минимальное рёберное 1-расширение гиперкуба Q_4

Заключение

Рассмотрен вопрос о единственности минимальных рёберных 1-расширений гиперкубов. Доказано общее свойство, что в минимальном рёберном 1-расширении любого гиперкуба Q_n при $n > 2$ не может быть рёбер между вершинами на расстоянии 2. Из этого следует единственность минимального рёберного 1-расширения гиперкуба Q_3. Продолжая рассуждения, было бы логично доказать, что в минимальном рёберном 1-расширении любого гиперкуба Q_n при $n > 3$ не может быть рёбер между вершинами на расстоянии 3, и далее, что в минимальном рёберном 1-расширении любого гиперкуба Q_n при $n > 2$ не может быть рёбер между вершинами на расстоянии $n - 1$. Из этого следовала бы единственность минимального рёберного 1-расширения любого гиперкуба. Удалось доказать только частный случай: в минимальном рёберном 1-расширении гиперкуба Q_4 не может быть рёбер между вершинами на расстоянии 3 и, следовательно, гиперкуб Q_4 имеет единственное с точностью до изоморфизма минимальное рёберное 1-расширение.

ЛИТЕРАТУРА

1. *Padua D. A.* Encyclopedia of Parallel Computing. N.Y.: Springer, 2011.
2. *Абросимов М. Б.* Графовые модели отказоустойчивости. Саратов: Изд-во Сарат. ун-та, 2012. 192 с.
3. *Hayes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C25. No. 9. P. 875–884.
4. *Harary F. and Hayes J. P.* Node fault tolerance in graphs // Networks. 1996. V. 27. P. 19–23.
5. *Harary F. and Hayes J. P.* Edge fault tolerance in graphs // Networks. 1993. V. 23. P. 135–142.
6. *Лобов А. А., Абросимов М. Б.* О вершинном 1-расширении гиперкуба // Компьютерные науки и информационные технологии: Материалы Междунар. науч. конф. Саратов: Издат. центр «Наука», 2018. С. 249–251.
7. *Zhang Y., Zhao S., and Meng J.* Edge fault tolerance of graphs with respect to $\lambda 2$-optimal property // Theor. Comput. Sci. 2019. V. 783. P. 95–104.

8. *Liu H. and Cheng D.* Structure fault tolerance of balanced hypercubes // Theor. Comput. Sci. 2020. V. 845. P. 198–207.
9. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997. 368 с.
10. *Харари Ф.* Теория графов. М.: Мир, 1973. 296 с.
11. *Harary F., Hayes J. P., and Wu H.-J.* A survey of the theory of hypercube graphs // Computers & Math. with Appl. 1988. V. 15. Ed. 4. P. 277–289.
12. *Абросимов М. Б.* О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. № 5(88). С. 643–650.
13. *Абросимов М. Б., Камил И. А. К., Лобов А. А.* Построение всех неизоморфных минимальных вершинных расширений графа методом канонических представителей // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2019. Т. 19. Вып. 4. С. 479–486.
14. *Абросимов М. Б., Судани Х. Х. К., Лобов А. А.* Построение минимальных рёберных расширений графа без проверки на изоморфизм // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2020. Т. 20. Вып. 1. С. 105–115.

REFERENCES

1. *Padua D. A.* Encyclopedia of Parallel Computing. N.Y., Springer, 2011.
2. *Abrosimov M. B.* Grafovye modeli otkazoustoychivosti [Graph Models of Fault Tolerance]. Saratov, SSU Publ., 2012. 192 p. (in Russian).
3. *Hayes J. P.* A graph model for fault-tolerant computing system. IEEE Trans. Comput., 1976, vol. C25, no. 9, pp. 875–884.
4. *Harary F. and Hayes J. P.* Node fault tolerance in graphs. Networks, 1996, vol. 27, pp. 19–23.
5. *Harary F. and Hayes J. P.* Edge fault tolerance in graphs. Networks, 1993, vol. 23, pp. 135–142.
6. *Lobov A. A. and Abrosimov M. B.* O vershinnom 1-rasshirenii giperkuba [On the vertex 1-extension of a hypercube]. Komp'yuternye nauki i informatsionnye tekhnologii, Saratov, "Nauka" Publ., 2018, pp. 249–251. (in Russian)
7. *Zhang Y., Zhao S., and Meng J.* Edge fault tolerance of graphs with respect to $\lambda 2$-optimal property. Theor. Comput. Sci., 2019, vol. 783, pp. 95–104.
8. *Liu H. and Cheng D.* Structure fault tolerance of balanced hypercubes. Theor. Comput. Sci., 2020, vol. 845, pp. 198–207.
9. *Bogomolov A. M. and Saliy V. N.* Algebraicheskie osnovy teorii diskretnykh sistem [Algebraic Foundations of the Discrete Systems Theory]. Moscow, Nauka Publ., 1997. 368 p. (in Russian).
10. *Harary F.* Graph Theory. Addison-Wesley, 1969. 274 p.
11. *Harary F., Hayes J. P., and Wu H.-J.* A survey of the theory of hypercube graphs. Computers & Math. with Appl., 1988, vol. 15, ed. 4, pp. 277–289.
12. *Abrosimov M. B.* On the complexity of some problems related to graph extensions. Math. Notes, 2010, vol. 88:5, pp. 619–625.
13. *Abrosimov M. B., Kamil I. A. K., and Lobov A. A.* Postroenie vsekh neizomorfnykh minimal'nykh vershinnykh rasshireniy grafa metodom kanonicheskikh predstaviteley [Construction of all nonisomorphic minimal vertex extensions of the graph by the method of canonical representatives]. Izv. Saratov Univ. (N. S.), Ser. Matematika. Mekhanika. Informatika, 2019, vol. 19, iss. 4, pp. 479–486. (in Russian).
14. *Abrosimov M. B., Sudani Kh. Kh. K., and Lobov A. A.* Postroenie minimal'nykh rebernykh rasshireniy grafa bez proverki na izomorfizm [Construction of all minimal edge extensions of the graph with isomorphism rejection]. Izv. Saratov Univ. (N. S.), Ser. Matematika. Mekhanika. Informatika, 2020, vol. 20, iss. 1, pp. 105–115. (in Russian).

УДК 519.87 DOI 10.17223/20710410/58/9

ЭФФЕКТИВНЫЙ АЛГОРИТМ ПОИСКА КРАТЧАЙШИХ ПУТЕЙ В ПЛОТНЫХ ГАУССИАНСКИХ СЕТЯХ[1]

Э. А. Монахова, О. Г. Монахов

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

E-mail: emilia@rav.sscc.ru, monakhov@rav.sscc.ru

Для семейства плотных Гауссианских сетей вида $C(D^2 + (D + 1)^2; D, D + 1)$ как перспективной топологии сетей на кристалле предложен алгоритм поиска кратчайших путей между вершинами графа, использующий относительную адресацию вершин и позволяющий в отличие от ряда известных алгоритмов рассчитать кратчайшие пути без использования координат соседних нулей решетки в плотной укладке графов на плоскости $\mathbb{Z}^2$. Это сокращает затраты памяти и времени выполнения по сравнению с другими алгоритмами при реализации данного алгоритма в сетях на кристалле с топологией плотной Гауссианской сети.

Ключевые слова: *плотные Гауссианские сети, циркулянтные графы, кратчайшие пути, сети на кристалле.*

EFFECTIVE ALGORITHM FOR FINDING SHORTEST PATHS IN DENSE GAUSSIAN NETWORKS

E. A. Monakhova, O. G. Monakhov

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

As a promising topology of networks on a chip, we consider a family of Dense Gaussian Networks, which are optimal circulant degree four graphs of the form $C(D^2+(D+1)^2; D, D+1)$. For this family, an algorithm for finding the shortest paths between graph vertices is proposed, which uses relative addressing of vertices and, unlike a number of the known algorithms, allows to calculate the shortest paths without using the coordinates of neighboring lattice zeros in a dense tessellation of graphs on the $\mathbb{Z}^2$ plane. This reduces the memory and execution time costs compared to other algorithms when the new algorithm is implemented on a network-on-chip with a Dense Gaussian Network topology.

Keywords: *Dense Gaussian Networks, circulant graphs, shortest paths, networks on a chip.*

Введение

Плотные Гауссианские сети [1 – 7], далее — DGNs (Dense Gaussian Networks), относятся к классу циркулянтных сетей [8 – 10] и представляют собой семейство оптимальных циркулянтных графов вида $C(N_D; D, D + 1)$ степени четыре и диаметра $D \geqslant 1$

[1]Работа поддержана проектом ИВМиМГ СО РАН №0251-2021-0005.

с числом вершин $N_D = D^2 + (D+1)^2$ и образующими $s_1 = D$ и $s_2 = D+1$. Графы данного семейства достигают максимально возможного числа вершин для заданного диаметра (расстояния между наиболее удалёнными друг от друга вершинами графа) и являются лучшими по диаметру и среднему расстоянию среди всех решётчатых графов степени четыре с таким же числом вершин, включая двумерные торы (2D torus) и двумерные решётки (2D mesh) [11]. Это вместе с вершинной симметрией и регулярностью связей позволяет рассматривать их как перспективную топологию для сетей на кристалле (Networks-on-Chip, NoC). В частности, DGNs могут вмещать больше узлов в одной и той же области чипа по сравнению с другими сетчатыми структурами. В [1, 2] плотные Гауссианские сети рассмотрены в качестве топологии для NoC, получена схема их оптимальной вложимости в чип без пересечения связей и разработаны иерархические двухуровневые структуры для NoC, в основе которых лежат плотные Гауссианские сети.

При проектировании сетей на кристалле важным этапом является разработка подсистемы связей, в том числе эффективных алгоритмов маршрутизации. В работах [1, 2] предложен алгоритм маршрутизации для DGNs, основанный на использовании координат всех соседних нулей решётки в плотной укладке (plane tessellation) графов семейства на плоскости $\mathbb{Z}^2$ для определения кратчайших путей между вершинами. При этом применяются операции суммирования относительных координат пары вершин «источник — приёмник» с координатами всех соседних нулей и последующее сравнение полученных сумм на минимум l_1-нормы, где l_1-норма определена как расстояние в двумерной целочисленной решётке $\mathbb{Z}^2$ [12]. Алгоритмы такого типа предложены как альтернатива методам маршрутизации из [13, 14], использующим деление Евклида с высокой вычислительной стоимостью.

В алгоритме маршрутизации из [1, 2] при определении кратчайших путей используются все девять соседних нулей; в алгоритме маршрутизации для Гауссианских сетей из [4] необходимое количество используемых соседних нулей сокращено до четырёх, но при этом также требуется найти минимум расстояний до четырёх нулей. В [15, 16] метод использования координат соседних нулей решётки при получении кратчайших путей между вершинами расширен для применения в оптимальных циркулянтных сетях степени четыре с любым числом вершин $N \geqslant 5$. При этом при поиске кратчайших путей в зависимости от значения N используется девять или семь соседних нулей. Отметим также работу [17], в которой рассмотрен аналогичный алгоритм маршрутизации для семейства циркулянтных графов с числом вершин $N = 2D^2$, где диаметр D — нечётное число, а количество используемых соседних нулей равно семи, и работы [5 – 7, 11, 18], в которых рассмотрены другие типы обменов либо другие подходы к построению алгоритмов маршрутизации в плотных Гауссианских сетях.

Из анализа алгоритмов маршрутизации [11, 15, 19, 20], реализованных в сетях на кристалле с двумерной циркулянтной топологией, следует, что критичными показателями для NoC наряду с задержкой передачи пакетов являются затраты ресурсов памяти и логических регистров. Как показано в [15], алгоритмы маршрутизации из [1, 2] и аналогичные им в [4, 15 – 17] , основанные на суммировании с координатами всех соседних нулей решётки и сравнении полученных l_1-норм (длин путей), требуют больших ресурсов памяти NoC для хранения промежуточных вычислений с большим количеством затратных по времени операций присваивания и взятия абсолютной величины, и, таким образом, отсутствие операций деления не является решающим фактором для NoC.

В настоящей работе получен алгоритм поиска кратчайших путей в плотных Гауссианских сетях константной сложности $O(1)$, модифицированный по сравнению с [1, 2, 4, 15 – 17, 21], который для любой пары вершин «источник — приёмник» определяет кратчайший путь (пути) между ними непосредственно по их координатам относительно нулевой вершины: 1) без использования суммирования координат со всеми соседними нулями решётки и последующего сравнения на минимум длин полученных путей; 2) с меньшим количеством операций присваивания и взятия абсолютной величины при отсутствии операций деления.

1. Алгоритм поиска кратчайших путей в плотных Гауссианских сетях

Пусть в циркулянтном графе $C(N_D; D, D+1)$, $D \geqslant 1$, (x, y) — координаты вершины u (далее — «источник») относительно вершины 0, а именно: x определяет число шагов вдоль образующей s_1 $(-s_1)$, а y — число шагов вдоль образующей s_2 $(-s_2)$ в кратчайшем пути из 0 в u. Аналогично определяются (x', y') — координаты вершины v (далее — «приёмник») относительно вершины 0. Требуется определить координаты (X, Y) — вектора кратчайшего пути между ними. Отметим, что вектор кратчайшего пути задаёт множество кратчайших путей между вершинами, отличающихся порядком прохождения образующих графа.

На рис. 1 показан фрагмент плотной укладки циркулянтного графа $C(N_D; D, D+1)$ на плоскости $\mathbb{Z}^2$. Пример дан для $D = 3$. Область ромба $R_D = \{(x, y) \in \mathbb{Z}^2 : |x| + |y| \leqslant \leqslant D\}$ с нулевой вершиной $Z_0 = (0, 0)$ соответствует рассматриваемому циркулянтному графу (графу DGN). Каждой клетке решётки (x, y) может быть сопоставлен номер вершины $m = xD + y(D+1) \bmod N_D$. Номера вершин графа $0 \leqslant m < N_D$ повторяются на плоскости бесконечное число раз, формируя плотную укладку. Восемь соседних с R_D ромбов получены трансляцией ромба R_D на плоскости. Центральные нули для соседних ромбов обозначены через $\pm Z_1, \pm Z_2$ и их комбинации. Для графов DGNs известны координаты относительно Z_0 соседних нулей:

$$Z_1 = (-D-1, D),\ Z_2 = (D, D+1),$$

из которых могут быть получены координаты других соседних нулей, например

$$Z_1 + Z_2 = (-1, 2D+1),\ -Z_1 + Z_2 = (2D+1, 1).$$

В [1, 2] доказано следующее утверждение: чтобы вычислить кратчайший путь из вершины (x, y) в вершину (x', y'), достаточно найти суммы относительного вектора $(x' - x, y' - y)$ с координатами девяти нулей Z_0, $\pm Z_1$, $\pm Z_2$, $\pm(Z_1 + Z_2)$, $\pm(Z_1 - Z_2)$ и сравнить полученные векторы на минимум длины пути.

Новый подход к разработке алгоритма поиска кратчайших путей для семейства графов DGNs заключается в следующем. Мы не используем суммирования с координатами всех соседних нулей и сравнения полученных результатов. Для этого всё множество $N = N_D$ вершин циркулянта DGN относительно вершины $Z_0 = (0, 0)$ разбиваем на четыре непересекающихся множества (квадранта), основанных на адресе источника, с числом вершин, равным $D(D+1)/2$. Обозначим эти множества через Q_i, $i = 1, \ldots, 4$ (рис. 2). Условия принадлежности источника (x, y), где $|x|, |y| \leqslant D$, к одному из этих квадрантов следующие:

$$Q_1 = \{(x, y) \in \mathbb{Z}^2 : -y \leqslant x < y\},\ Q_2 = \{(x, y) \in \mathbb{Z}^2 : -x < y \leqslant x\},$$
$$Q_3 = \{(x, y) \in \mathbb{Z}^2 : y < x \leqslant -y\},\ Q_4 = \{(x, y) \in \mathbb{Z}^2 : x \leqslant y < -x\}.$$

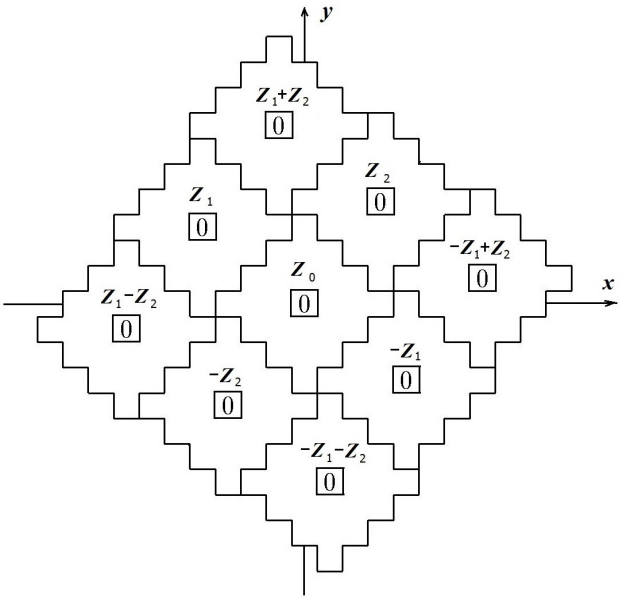

Рис. 1. Фрагмент плотной укладки графа $C(N_D; D, D+1)$ на плоскости $\mathbb{Z}^2$

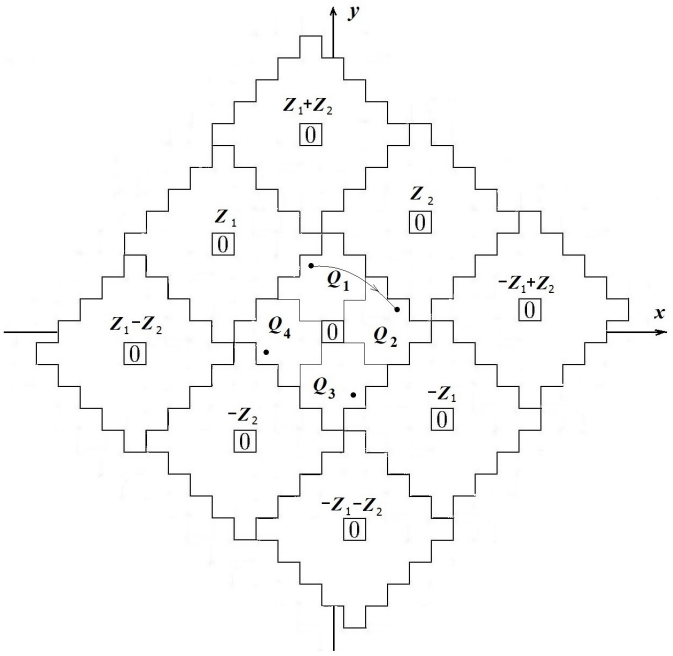

Рис. 2. Разбиение циркулянта $C(N_D; D, D+1)$ на квадранты Q_i, $i = 1, \ldots, 4$, по адресу источника

При попадании источника $u = (x, y)$ в один из квадрантов Q_i, $i = 1, \ldots, 4$, в зависимости от того, где находится приёмник $v = (x', y')$, $|x'|, |y'| \leqslant D$, ромб R_D с центром в Z_0 в общем случае также разбивается на четыре множества вершин соответственно тому, до какого из соседних нулей решётки оказывается ближе приемник v.

Сначала рассмотрим случай, когда $(x, y) \in Q_1$. На рис. 3 показан фрагмент укладки циркулянта $C(41; 4, 5)$ диаметра $D = 4$ на плоскости $\mathbb{Z}^2$ и соседние нули Z_0, Z_1, Z_2, $Z_1 + Z_2$, ближайшие для любого источника $(x, y) \in Q_1$. Пример дан для источника u

(обозначен знаком «·») с координатами $(x, y) = (-1, 3)$ относительно Z_0. В ромбе R_D с центром в Z_0 разными цветами и римскими цифрами обозначены: I — приёмники, находящиеся на расстоянии не более D от u и относящиеся соответственно к Z_0; II — приёмники, для которых ближайшим нулём из источника u является Z_1; III — приёмники, для которых ближайший нуль Z_1+Z_2; IV — приёмники, для которых ближайший нуль Z_2.

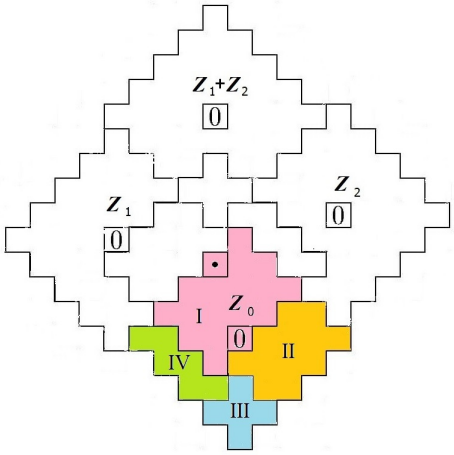

Рис. 3. Источник $(x, y) \in Q_1$

Условия принадлежности приёмников $v = (x', y')$ к соответствующим ближайшим соседним нулям находим из соотношений между $Y = y' - y$ и $X = x' - x$, задающих границы областей. Для всех приёмников, относящихся к Z_0, имеем $y' - y + D \geqslant x' - x$, $y' - y + D \geqslant x - x'$ (равенства достигаются на левой или правой границах области I), что равносильно условию $|y' - y| + |x' - x| \leqslant D$. Для приёмников, ближайших к Z_1 из u, имеем $x - x' < y' - y + D < x' - x$; для приёмников, ближайших к Z_2 из u, — $x' - x - 1 \leqslant y' - y + D < x - x'$; для оставшихся приёмников, принадлежащих области III, $Z_1 + Z_2$ — ближайший нуль из u. Отсюда следует

Утверждение 1. Пусть в графе $C(N_D; D, D + 1)$ с диаметром $D > 1$ даны вершины $u = (x, y)$ и $v = (x', y') \neq u$, такие, что $|x| + |y| \leqslant D$, $|x'| + |y'| \leqslant D$ и $y > x$, $y \geqslant -x$. Тогда вектор кратчайшего пути из u в v имеет следующий вид:

$$(X, Y) = \begin{cases} (x' - x, y' - y), & \text{если } |x' - x| + |y' - y| \leqslant D, \\ (x' - x, y' - y) + Z_1, & \text{если } x - x' < y' - y + D < x' - x, \\ (x' - x, y' - y) + Z_2, & \text{если } x' - x - 1 \leqslant y' - y + D < x - x', \\ (x' - x, y' - y) + Z_1 + Z_2 & \text{в остальных случаях.} \end{cases}$$

Здесь и далее операции типа $(X, Y) \pm Z_i$, $i = 1, 2$, означают сложение соответствующих координат векторов.

Для вершин, находящихся на границе Q_1, из утверждения 1 следует: если для источника (x, y) выполняется $y = -x$, $-\lfloor D/2 \rfloor \leqslant x \leqslant -1$, то Z_2 не является ближайшим нулём ни для какой вершины (x', y'); если для источника (x, y) выполняется $y = x + 1$, $0 \leqslant x < \lceil D/2 \rceil$, то $Z_1 + Z_2$ не является ближайшим нулём ни для какой вершины (x', y').

Теперь, зная, как определяются векторы кратчайших путей в любой приёмник из источника, принадлежащего Q_1, можем получить для источников, принадлежащих Q_i,

$i = 2, 3, 4$, векторы кратчайших путей в любой приёмник с помощью соответствующих ротаций источника (направление ротаций показано стрелкой на рис. 2), приёмника и соседних нулей. На рис. 2 отмечены знаком «·» вершины $(y, -x) = (3, 1) \in Q_2$, $(-x, -y) = (1, -3) \in Q_3$ и $(-y, x) = (-3, -1) \in Q_4$, полученные ротацией источника $(x, y) = (-1, 3) \in Q_1$. Рис. 4 даёт пример распределения вершин приёмников по областям, относящимся к ближайшим соседним нулям для источника u с координатами $(3, 1)$, принадлежащего Q_2. В ромбе R_D с центром в Z_0 разными цветами и римскими цифрами обозначены: I — приёмники, находящиеся на расстоянии не более D от u и относящиеся соответственно к Z_0; II — приёмники, для которых ближайшим нулем из u является Z_2; III — приёмники, для которых ближайший нуль — это $Z_2 - Z_1$; IV — приёмники, для которых ближайший нуль $-Z_1$.

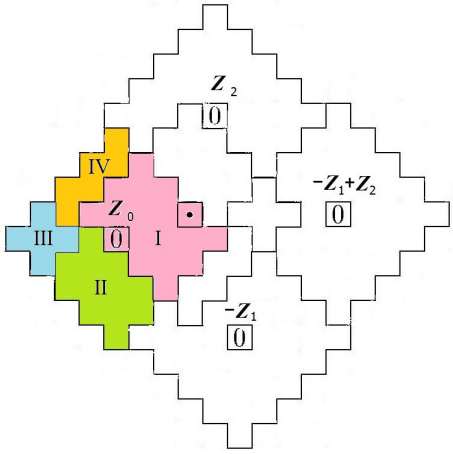

Рис. 4. Источник $(x, y) \in Q_2$

Таким образом, по аналогии с утверждением 1 для источников $(x, y) \in Q_2$ получаем

Утверждение 2. Пусть в графе $C(N_D; D, D+1)$ диаметра $D > 1$ даны вершины $u = (x, y)$ и $v = (x', y') \neq u$, такие, что $|x| + |y| \leqslant D$, $|x'| + |y'| \leqslant D$ и $-x < y \leqslant x$. Тогда вектор кратчайшего пути из u в v имеет следующий вид:

$$(X, Y) = \begin{cases} (x' - x, y' - y), & \text{если } |x' - x| + |y' - y| \leqslant D, \\ (x' - x, y' - y) - Z_1, & \text{если } y' - y - D > x' - x \text{ и } y' - y + D + 1 \geqslant x - x', \\ (x' - x, y' - y) + Z_2, & \text{если } y' - y - D < x' - x \text{ и } y' - y + D < x - x', \\ (x' - x, y' - y) - Z_1 + Z_2 & \text{в остальных случаях.} \end{cases}$$

Аналогично определяются векторы кратчайших путей между вершинами, когда источник находится в Q_3 или Q_4. Согласно полученным результатам, разработан алгоритм RDGN (Routing in Dense Gaussian Networks) поиска кратчайших путей в плотных Гауссианских сетях (алгоритм 1).

Алгоритм 1. RDGN — поиск кратчайших путей в плотных Гауссианских сетях

Вход: (x, y) — источник; (x', y') — приёмник; D — диаметр графа.
Выход: (X, Y) — вектор кратчайших путей из (x, y) в (x', y').

$X := x' - x$; $Y := y' - y$.
Если $|X| + |Y| > D$, **то**
  // пп. 3–7 — источник в квадранте Q_1
  Если $(y > x)\&(y \geqslant -x)$, **то**
    Если $(Y + X < -D)$ & $(Y - X \geqslant -D - 1)$, **то**
      $(X, Y) := (X, Y) + Z_2$,
    иначе
    Если $(Y + X > -D)$ & $(Y - X < -D)$, **то**
      $(X, Y) := (X, Y) + Z_1$,
      иначе $(X, Y) := (X, Y) + Z_1 + Z_2$,
  иначе *// пп. 9–13 — источник в квадранте Q_2*
    Если $(y \leqslant x)\&(y > -x)$, **то**
      Если $(Y + X < -D)\&(Y - X < D)$, **то**
        $(X, Y) := (X, Y) + Z_2$,
      иначе
        Если $(Y + X \geqslant -D - 1)\&(Y - X > D)$, **то**
          $(X, Y) := (X, Y) - Z_1$,
          иначе $(X, Y) := (X, Y) + Z_2 - Z_1$,
      иначе *// пп. 15–19 — источник в квадранте Q_3*
        Если $(y < x)\&(y \leqslant -x)$, **то**
          Если $(Y + X > D)\&(Y - X \leqslant D + 1)$, **то**
            $(X, Y) := (X, Y) - Z_2$,
          иначе
            Если $(Y + X < D)\&(Y - X > D)$, **то**
              $(X, Y) := (X, Y) - Z_1$,
              иначе $(X, Y) := (X, Y) - Z_1 - Z_2$,
        иначе *// пп. 21–24 — источник в квадранте Q_4*
          Если $(Y + X > D)\&(Y - X > -D)$, **то**
            $(X, Y) := (X, Y) - Z_2$,
          иначе
            Если $(Y + X \leqslant D + 1)\&(Y - X < -D)$, **то**
              $(X, Y) := (X, Y) + Z_1$
              иначе $(X, Y) := (X, Y) - Z_2 + Z_1$.

Корректность работы алгоритма RDGN проверена для плотных Гауссианских сетей с числом вершин от 13 до 2521 на множестве всех пар вершин «источник — приёмник» в системе Wolfram Mathematica 10 сравнением полученных кратчайших путей с результатами работы алгоритма Дейкстры [22].

2. Оценки алгоритма поиска кратчайших путей

Приведём некоторые теоретические оценки алгоритма RDGN. Используемое количество переменных: четыре входных x, y, x', y', две выходных X, Y и глобальная переменная D. Для плотных Гауссианских сетей диаметр $D = \lfloor\sqrt{N/2}\rfloor$. Кроме того, $|x|, |y|, |x'|, |y'|, |X|, |Y| \leqslant D$. Для хранения D в маршрутизаторе требует-

ся $\lceil \log_2 \lfloor \sqrt{N/2} \rfloor \rceil$ битов. Для хранения входных и выходных переменных добавляем по одному биту для знака и получаем: общее количество необходимой памяти в битах для хранения переменных в алгоритме RDGN равно $M = 7\lceil \log_2 \lfloor \sqrt{N/2} \rfloor \rceil + 6$. Таким образом, по сравнению с алгоритмом поиска кратчайших путей из [1, 2] происходит более чем трёхкратное уменьшение требуемой памяти. Алгоритм RDGN также требует меньше ресурсов памяти в NoC по сравнению с [15 – 17] за счёт меньшего количества переменных, используемых для его работы.

Число операций разных типов, применяемых в алгоритме RDGN, приведено в таблице. Показаны минимальное (min) и максимальное (max) значения числа операций при исполнении алгоритма и среднее время выполнения одной операции в наносекундах, найденное в [15] (при подсчёте использовался язык программирования C#, процессор Intel Core i5-6500).

Операция	$\lvert a \rvert$	$=$	$\&$	$+$	$-$	$\times$	$<$, $>$, $\leqslant$, $\geqslant$	$\div$
Время, нс	6,436	2,978	0,655	0,770	0,672	0,722	0,692	2,530
min/max	2/2	2/4	0/5	2/8	1/12	0/0	1/11	0/0

По сравнению с алгоритмами поиска кратчайших путей в плотных Гауссианских сетях из [1, 2, 4, 15 – 17, 21] алгоритм RDGN выполняет больше операций сравнения, но намного меньше затратных по времени операций взятия абсолютной величины ($|a|$) и присваивания ($=$) (например, в 8 раз меньше, чем в алгоритме 1 из [1]) и, в отличие от алгоритмов из [4, 21], не использует операций деления ($\div$).

Кроме того, моделирование парных обменов в сети типа «каждый — каждому», проведённое для плотных Гауссианских сетей с числом вершин $13 \leqslant N_D \leqslant 313$, показало следующий результат: условие $|X| + |Y| \leqslant D$ алгоритма выполняется в 52–56 % случаев пар «источник — приёмник», а это означает минимальное число исполняемых операций алгоритма RDGN и соответственно минимальное время его работы. На рис. 5 приведены результаты моделирования. Параметр λ означает отношение числа пар «источник — приёмник» с l_1-нормой, меньшей или равной диаметру, к общему числу пар вершин плотных Гауссианских сетей с числов вершин N.

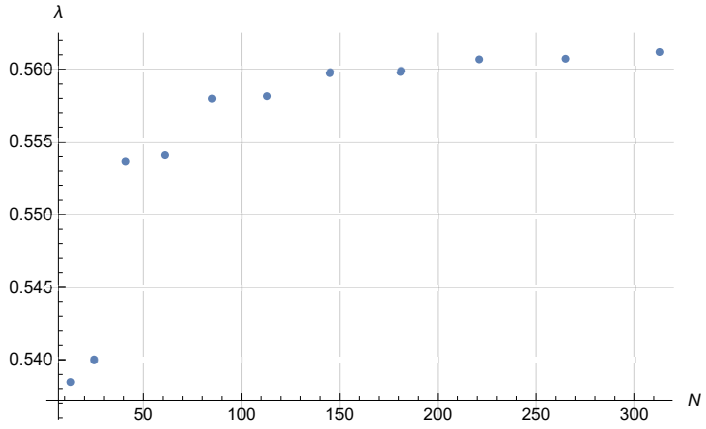

Рис. 5. Относительное сокращение числа операций алгоритма RDGN при выполнении условия $|X| + |Y| \leqslant D$

Заключение

Для семейства плотных Гауссианских сетей, как перспективной топологии сетей на кристалле по сравнению с 2D torus и 2D mesh, предложен новый алгоритм поиска

кратчайших путей между вершинами графа, использующий относительную адресацию вершин и позволяющий, в отличие от ряда известных алгоритмов маршрутизации, рассчитать кратчайшие пути без суммирования с координатами всех соседних нулей в плотной укладке графов на плоскости. Теоретические оценки алгоритма подтверждают уменьшение затрат памяти и времени выполнения при реализации в сети на кристалле по сравнению с известными алгоритмами маршрутизации, реализованными в NoC. В будущем планируется апробация разработанного алгоритма в сети на кристалле и экспериментальное сравнение его оценок памяти и эффективности с другими алгоритмами маршрутизации, а также исследование возможности расширения разработки аналогичных алгоритмов маршрутизации для любого числа вершин графов семейства оптимальных двумерных циркулянтов [10], к которому плотные Гауссианские сети принадлежат по определению.

ЛИТЕРАТУРА

1. *Martinez C., Vallejo E., Beivide R., et al.* Dense Gaussian networks: Suitable topologies for on-chip multiprocessors // Intern. J. Parallel Programming. 2006. V. 34. P. 193–211.
2. *Martinez C., Vallejo E., Moreto M., et al.* Hierarchical topologies for large-scale two-level networks // XVI Jornadas de Paralelismo, Granada, Spain, September 2005. `https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.7372`.
3. *Martinez C., Beivide R., Stafford E., et al.* Modeling toroidal networks with the Gaussian integers // IEEE Trans. Computers. 2008. V. 57. No. 8. P. 1046–1056.
4. *Flahive M. and Bose B.* The topology of Gaussian and Eisenstein — Jacobi interconnection networks // IEEE Trans. Parall. Distrib. Syst. 2010. V. 21. No. 8. P. 1132–1142.
5. *Pay K.-J., Yang J.-S., Chen G.-Y., and Chang J.-M.* Configuring protection routing via completely independent spanning trees in Dense Gaussian On-Chip Networks // IEEE Trans. Netw. Sci. Eng. 2022. V. 9. No. 2. P. 932–946.
6. *Touzene A.* On all-to-all broadcast in dense Gaussian network on-chip // IEEE Trans. Parall. Distrib. Syst. 2015. V. 26. No. 4. P. 1085–1095.
7. *Alsaleh O., Bose B., and Hamdaoui B.* On-to-many node-disjoint paths routing in dense Gaussian networks // Computer J. 2015. V. 58. No. 2. P. 173–187.
8. *Bermond J.-C., Comellas F., and Hsu D. F.* Distributed loop computer networks: a survey // J. Parallel Distrib. Comput. 1995. No. 24 (1). P. 2–10.
9. *Hwang F. K.* A survey on multi-loop networks // Theor. Comput. Sci. 2003. V. 299. P. 107–121.
10. *Монахова Э. А.* Структурные и коммуникативные свойства циркулянтных сетей // Прикладная дискретная математика. 2011. № 3 (13). С. 92–115.
11. *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.
12. *Zerovnik J. and Pisanski T.* Computing the diameter in multiple-loop networks // J. Algorithms. 1993. V. 14. P. 226–243.
13. *Gomez D., Gutierrez J., Ibeas A., et al.* On finding a shortest path in circulant graphs with two jumps // LNCS. 2005. V. 3595. P. 777–786.
14. *Cai J. Y., Havas G., Mans B., et al.* On routing in circulant graphs // LNCS. 1999. V. 1627. P. 360–369.
15. *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 Science and Engineering. 2022. https://ieeexplore.ieee.org/document/9910381.

16. *Monakhova E. and Monakhov O.* A generalized routing algorithm for a family of optimal 2D circulant networks based on relative addressing // Proc. 17th Inter. Asian School-Seminar (OPCS-21), Novosibirsk, Russia. 2021. P. 55–59.
17. *Jha P. K.* Dimension-order routing algorithms for a family of minimal-diameter circulants // J. Inter. Networks. 2013. V. 14. No. 1. P. 1350002 (24 pages).
18. *Монахова Э. А.* Поиск кратчайших путей в оптимальных двумерных циркулянтах // Прикладная дискретная математика. 2020. № 47. С. 87–100.
19. *Romanov A. Y.* Development of routing algorithms in networks-on-chip based on ring circulant topologies // Heliyon. 2019. V. 5. Iss. 4. Article e01516.
20. *Romanov A. Y., Lezhnev E. V., Glukhikh A. Y., and Amerikanov A. A.* Development of routing algorithms in networks-on-chip based on two-dimensional optimal circulant topologies // Heliyon. Jan. 2020. V. 6. No. 1. Art. No. e03183.
21. *Camarero C., Martinez C., and Beivide R.* L-networks: A topological model for regular two-dimensional interconnection networks // IEEE Trans. Computers. 2013. V. 62. No. 7. P. 1362–1375.
22. *Dijkstra E. W.* A note on two problems in connexion with graphs // Numer. Math. 1959. V. 1. No. 1. P. 269–271.

REFERENCES

1. *Martinez C., Vallejo E., Beivide R., et al.* Dense Gaussian networks: Suitable topologies for on-chip multiprocessors. Inter. J. Parallel Programming, 2006, vol. 34, pp. 193–211.
2. *Martinez C., Vallejo E., Moreto M., et al.* Hierarchical topologies for large-scale two-level networks. XVI Jornadas de Paralelismo, Granada, Spain, September 2005. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.7372.
3. *Martinez C., Beivide R., Stafford E., et al.* Modeling toroidal networks with the Gaussian integers. IEEE Trans. Comput., 2008, vol. 57, no. 8, pp. 1046–1056.
4. *Flahive M. and Bose B.* The topology of Gaussian and Eisenstein — Jacobi interconnection networks. IEEE Trans. Parall. Distrib. Syst., 2010, vol. 21, no. 8, pp. 1132–1142.
5. *Pay K.-J., Yang J.-S., Chen G.-Y., and Chang J.-M.* Configuring protection routing via completely independent spanning trees in Dense Gaussian On-Chip Networks. IEEE Trans. Netw. Sci. Eng., 2022, vol. 9, no. 2, pp. 932–946.
6. *Touzene A.* On all-to-all broadcast in dense Gaussian network on-chip. IEEE Trans. Parall. Distrib. Syst., 2015, vol. 26, no. 4, pp. 1085–1095.
7. *Alsaleh O., Bose B., and Hamdaoui B.* On-to-many node-disjoint paths routing in dense Gaussian networks. Computer J., 2015, vol. 58, no. 2, pp. 173–187.
8. *Bermond J.-C., Comellas F., and Hsu D. F.* Distributed loop computer networks: a survey. J. Parallel Distrib. Comput., 1995, no. 24 (1), pp. 2–10.
9. *Hwang F. K.* A survey on multi-loop networks. Theor. Comput. Sci., 2003, vol. 299, pp. 107–121.
10. *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)
11. *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, pp. 215010–215019.

12. *Zerovnik J. and Pisanski T.* Computing the diameter in multiple-loop networks. J. Algorithms, 1993, vol. 14, pp. 226–243.
13. *Gomez D., Gutierrez J., Ibeas A., et al.* On finding a shortest path in circulant graphs with two jumps. LNCS, 2005, vol. 3595, pp. 777–786.
14. *Cai J. Y., Havas G., Mans B., et al.* On routing in circulant graphs. LNCS, 1999, vol. 1627, pp. 360–369.
15. *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 Science and Engineering. 2022. `https://ieeexplore.ieee.org/document/9910381`.
16. *Monakhova E. and Monakhov O.* A generalized routing algorithm for a family of optimal 2D circulant networks based on relative addressing. Proc. 17th Intern. Asian School-Seminar (OPCS-21), Novosibirsk, Russia, 2021, pp. 55–59.
17. *Jha P. K.* Dimension-order routing algorithms for a family of minimal-diameter circulants. J. Inter. Networks, 2013, vol. 14, no. 1, pp. 1350002 (24 pages).
18. *Монахова Э. А.* Poisk kratchajshih putej v optimal'nyh dvumernyh cirkulyantah [Search for the shortest paths in optimal two-dimensional circulants]. Prikladnaya Diskretnaya Matematika, 2020, no. 47, pp. 87–100. (in Russian)
19. *Romanov A. Yu.* Development of routing algorithms in networks-on-chip based on ring circulant topologies. Heliyon, 2019, vol. 5, iss. 4, article e01516.
20. *Romanov A. Y., Lezhnev E. V., Glukhikh A. Y., and Amerikanov A. A.* Development of routing algorithms in networks-on-chip based on two-dimensional optimal circulant topologies. Heliyon, 2020, vol. 6, no. 1, art. no. e03183.
21. *Camarero C., Martinez C., and Beivide R.* L-networks: A topological model for regular two-dimensional interconnection networks. IEEE Trans. Computers, 2013, vol. 62, no. 7, pp. 1362–1375.
22. *Dijkstra E. W.* A note on two problems in connexion with graphs. Numer. Math., 1959, vol. 1, no. 1, pp. 269–271.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52 DOI 10.17223/20710410/58/10

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ РАЗБИЕНИЯ ГРАФА НА ТРЕУГОЛЬНИКИ[1]

А. Н. Рыбалов

Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия

E-mail: alexander.rybalov@gmail.com

NP-полнота проблемы разбиения графа на треугольники доказана Шейфером в 1974 г. и содержится в классической монографии М. Гэри и Д. Джонсона. В данной работе изучается генерическая сложность этой проблемы. Доказывается, что при условии $P \neq NP$ и $P = BPP$ для её решения не существует полиномиального сильно генерического алгоритма.

Ключевые слова: *генерическая сложность, разбиение графа на треугольники.*

THE GENERIC COMPLEXITY OF THE GRAPH TRIANGULATION PROBLEM

A. N. Rybalov

Sobolev Institute of Mathematics, Omsk, Russia

Generic-case approach to algorithmic problems was suggested by A. Miasnikov, V. Kapovich, P. Schupp, and V. Shpilrain in 2003. This approach studies behavior of an algorithm on typical (almost all) inputs and ignores the rest of inputs. In this paper, we study the generic complexity of the graph triangulation problem. This problem is as follows. Given a finite simple graph with $3n$ vertices, determine whether the vertices of the graph can be divided into n three-element sets, each of which contains vertices connected by edges of the original graph (that is, they are triangles). NP-completeness of this problem was proved by Shaffer in 1974 and is mentioned in the classic monograph by M. Garey and D. Johnson. We prove that under the conditions $P \neq NP$ and $P = BPP$ there is no polynomial strongly generic algorithm for this problem. A strongly generic algorithm solves a problem not on the whole set of inputs, but on a subset whose frequency sequence converges exponentially to 1 with increasing size. To prove the theorem, we use the method of generic amplification, which allows one to construct generically hard problems from the problems that are hard in the classical sense. The main component of this method is the cloning technique, which combines the inputs of a problem together into sufficiently large sets of equivalent inputs. Equivalence is understood in the sense that the problem for them is solved in a similar way.

Keywords: *generic complexity, graph triangulation problem.*

[1]Работа поддержана грантом РНФ № 22-11-20019.

Введение

Проблема разбиения графа на треугольники состоит в следующем. Задан конечный простой граф с $3n$ вершинами. Необходимо определить, можно ли разбить вершины графа на n трёхэлементных множеств так, что в каждом из них все вершины соединены рёбрами исходного графа (то есть являются треугольниками). NP-полнота этой проблемы доказана Шейфером в 1974 г. и содержится в классической монографии [1]. При условии $\mathrm{P} \neq \mathrm{NP}$ не существует полиномиального алгоритма для решения этой задачи. Здесь P — это класс алгоритмических проблем распознавания, решаемых за полиномиально ограниченное время на детерминированных машинах Тьюринга, а класс NP состоит из проблем, решаемых за полиномиальное время на недетерминированных машинах Тьюринга.

Генерический подход [2] — это один из подходов к изучению алгоритмических проблем для «почти всех» входов. В рамках этого подхода алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. Такие входы образуют генерическое множество. Понятие «почти все» формализуется введением естественной меры на множестве входных данных. С точки зрения практики алгоритмы, решающие быстро проблему на генерическом множестве, так же хороши, как и быстрые алгоритмы для всех входов. Отметим, что похожий подход для изучения проблем оптимизации предложен ранее в [3].

Большой интерес как с теоретической точки зрения, так и с точки зрения практических приложений представляют алгоритмические проблемы, которые остаются неразрешимыми или трудноразрешимыми и в генерическом случае. Например, в современной криптографии интересны такие проблемы, которые, являясь (гипотетически) трудными в классическом смысле, остаются трудными и в генерическом смысле, т. е. для почти всех входов. Это объясняется тем, что при случайной генерации ключей в криптографическом алгоритме происходит генерация входа некоторой трудной алгоритмической проблемы, лежащей в основе алгоритма. Если проблема генерически легкоразрешима, то для почти всех таких входов её можно быстро решить и ключи почти всегда будут нестойкими. Поэтому проблема должна быть трудной для почти всех входов. Например, таким поведением обладают классические алгоритмические проблемы криптографии: распознавания квадратичных вычетов, дискретного логарифма, извлечения корня в группах вычетов (обращения функции RSA).

М. Блюм для доказательства того, что проблема дискретного логарифма является сложной для почти всех входов, при условии её трудноразрешимости в худшем случае, предложил метод амплификации [4]. Идея метода состоит в следующем. По заданному входу x алгоритмической проблемы A случайно генерируется «большое» множество входов $C(x)$, эквивалентных x в том смысле, что по решению проблемы для любого входа из $C(x)$ можно эффективно найти решение и для x. Если теперь допустить существование эффективного (полиномиального) генерического алгоритма для проблемы A, то из него можно получить вероятностный полиномиальный алгоритм, решающий проблему A для всех входов. Этот алгоритм по входу x путём генерации «большого» множества $C(x)$ как бы переносит вход x из области «плохих» входов для генерического алгоритма в область, где этот алгоритм может решить проблему. Таким образом, если проблема A трудноразрешима в худшем случае, то и генерически она тоже трудноразрешима. Оказалось, что метод амплификации применим и для перечисленных выше проблем криптографии.

Метод генерической амплификации был развит в [5]. С его помощью доказана генерическая неразрешимость и трудноразрешимость многих алгоритмических проблем:

проблема остановки для машин Тьюринга [6], проблема равенства в некоторых конечно определённых полугруппах [5], проблема разрешимости элементарных теорий, неразрешимых в классическом случае [5], арифметика Пресбургера [7], десятая проблема Гильберта [8], проблема выполнимости булевых формул [9], проблема кластеризации графов [10], проблема распознавания гамильтоновых графов [11].

Данная работа посвящена изучению генерической сложности разбиения графа на треугольники. Доказывается, что при условии $\mathrm{P} \neq \mathrm{NP}$ и $\mathrm{P} = \mathrm{BPP}$ для этой проблемы не существует полиномиального сильно генерического алгоритма. Сильно генерический алгоритм решает проблему не на всём множестве входов, а на подмножестве, последовательность частот которого при увеличении размера экспоненциально быстро сходится к 1. Класс BPP состоит из проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. Одной из важных гипотез в теории сложности вычислений является гипотеза о совпадении классов P и BPP. Из нее следует, что любой полиномиальный вероятностный алгоритм $\mathcal{A}$ можно эффективно дерандомизировать, то есть построить полиномиальный алгоритм $\mathcal{B}$, не использующий генератор случайных чисел и решающий ту же проблему, что и алгоритм $\mathcal{A}$. В работе [12] доказано, что равенство $\mathrm{P} = \mathrm{BPP}$ следует из весьма правдоподобных гипотез о вычислительной сложности некоторых трудных проблем.

1. Предварительные сведения

Пусть I — некоторое множество входов, I_n — подмножество входов размера n. Для подмножества $S \subseteq I$ определим последовательность

$$\rho_n(S) = \frac{|S_n|}{|I_n|},\ n = 1, 2, 3, \ldots,$$

где $S_n = S \cap I_n$ — множество входов из S размера n. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. *Асимптотической плотностью* S назовём предел

$$\rho(S) = \overline{\lim_{n\to\infty}} \rho_n(S).$$

Верхний предел здесь нужен потому, что часто при кодировании входных данных не для каждого n существуют коды размера n. Множество S называется *пренебрежимым*, если $\rho(S) = 0$, и *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что для любого n имеет место $\rho_n(S) < C\sigma^n$.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется (*сильно*) *генерическим*, если

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) = ?\}$ является (сильно) пренебрежимым.

Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если для всех $x \in I$ выполнено

$$(\mathcal{A}(x) = y \in J) \Rightarrow (f(x) = y).$$

Ситуация $\mathcal{A}(x) = ?$ означает, что $\mathcal{A}$ не может вычислить функцию f на аргументе x. Но условие 2 гарантирует, что $\mathcal{A}$ корректно вычисляет f на почти всех входах (входах из генерического множества). Различие между генерически разрешимыми проблемами и сильно генерически разрешимыми проблемами поясняется в работе [13].

Напомним некоторые понятия классической теории сложности вычислений. *Время работы* $t_M(x)$ машины Тьюринга M на входе $x \in I$ — это число шагов машины от начала работы до остановки. Если M на x не останавливается, полагаем $t_M(x) = \infty$. Машина Тьюринга M *полиномиальна*, если существует полином $p(n)$, такой, что для любого $x \in I$ имеет место $t_M(x) < p(|x|)$. Здесь через $|x|$ обозначен размер входа x. Класс P состоит из подмножеств I, распознаваемых полиномиальными машинами Тьюринга.

Вероятностная машина Тьюринга — это машина Тьюринга, в программе которой допускаются пары правил вида

$$(q_i, a) \to (q_j, b, S_1),$$
$$(q_i, a) \to (q_k, c, S_2).$$

В процессе работы такой машины с вероятностью $1/2$ выбирается первое или второе правило. Обозначим через $\mathsf{P}[M(x) = y]$ вероятность того, что машина M на входе x выдаёт ответ y. Время работы $t_M(x, \tau)$ вероятностной машины Тьюринга на входе x зависит от вычислительного пути (последовательности выполненных команд) τ. Проблема $S \subseteq I$ принадлежит *классу* BPP, если существует вероятностная машина Тьюринга M и полином $p(n)$, такие, что

1) для любого x и для любого вычислительного пути τ машины M на x имеет место $t_M(x, \tau) < p(|x|)$;
2) если $x \in S$, то $\mathsf{P}[M(x) = 1] > 2/3$;
3) если $x \notin S$, то $\mathsf{P}[M(x) = 0] > 2/3$.

Вероятностные машины Тьюринга формализуют понятие алгоритма, использующего генератор случайных чисел. Класс BPP — это класс проблем, эффективно решаемых такими вероятностными алгоритмами. Большинство специалистов по теоретической информатике считают, что имеет место равенство P = BPP. Это означает, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, т. е. построить полиномиальный детерминированный алгоритм, решающий ту же задачу. Хотя равенство пока не доказано, имеются серьёзные результаты в его пользу [12].

2. Разбиение графа на треугольники

Рассмотрим неориентированые графы без петель и кратных рёбер. *Проблема разбиения графа на треугольники* состоит в следующем. Пусть задан граф G с множеством вершин $V = \{v_1, \ldots, v_{3n}\}$. Необходимо определить, можно ли разбить вершины графа на n трёхэлементных множеств так, что в каждом из них все вершины соединены рёбрами исходного графа (то есть являются треугольниками). Набор из этих n трёхэлементных множеств будем называть *3-разбиением*.

Лемма 1. Пусть G_1 и G_2 — два графа с непересекающимися множествами вершин $V = \{v_1, \ldots, v_{3n}\}$ и $W = \{w_1, \ldots, w_{3m}\}$. Пусть граф G_2 можно разбить на треугольники. Тогда граф G_1 можно разбить на треугольники тогда и только тогда, когда граф $G_1 \cup G_2$ можно разбить на треугольники.

Доказательство. Пусть графы G_1 и G_2 можно разбить на треугольники и $P(G_1)$, $P(G_2)$ — соответствующие 3-разбиения. Тогда легко видеть, что 3-разбиение графа $G_1 \cup G_2$ есть $P(G_1) \cup P(G_2)$.

Обратно, пусть графы G_2 и $G_1 \cup G_2$ можно разбить на треугольники и $P(G_2)$, $P(G_1 \cup G_2)$ — соответствующие 3-разбиения. Так как подграфы G_1 и G_2 в графе $G_1 \cup G_2$ не имеют общих вершин и рёбер, 3-разбиением графа G_1 является $P(G_1 \cup G_2) \setminus P(G_2)$. ■

Будем использовать представление графов с помощью матриц смежности. Напомним, что матрица смежности $M(G)$ графа G с множеством вершин $v_1, \ldots, v_n$ — это матрица порядка n, в которой на месте (i, j) стоит 1, если в графе G есть ребро (v_i, v_j), и 0, если такого ребра нет. Так как графы неориентированные, для кодирования графа с n вершинами достаточно верхней части матрицы, состоящей из $n(n-1)/2$ бит. Таким образом, будем считать, что размер графа с n вершинами равен $n(n-1)/2$.

Теорема 1. Если P $\neq$ NP и P=BPP, то не существует сильно генерического полиномиального алгоритма для решения проблемы разбиения графа на треугольники.

Доказательство. Допустим, что существует сильно генерический полиномиальный алгоритм $\mathcal{A}$, решающий проблему разбиения графа на треугольники. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всём множестве входов. Это означает, что проблема разбиения графа на треугольники лежит в классе BPP. Так как P = BPP, она лежит и в классе P, откуда из NP-полноты данной проблемы следует P $\neq$ NP, что противоречит посылке теоремы.

Пусть имеется граф G с $3n$ вершинами $v_1, \ldots, v_{3n}$. Он имеет размер $3n(3n-1)/2$. Алгоритм $\mathcal{B}$ работает на графе G следующим образом:

1) Генерируем случайный граф H с $(3n)^2-3n$ вершинами $v_{3n+1}, \ldots, v_{(3n)^2}$, имеющий следующее 3-разбиение:

$$P(H) = \{(v_{3n+1}, v_{3n+2}, v_{3n+3}), \ldots, (v_{(3n)^2-2}, v_{(3n)^2-1}, v_{(3n)^2})\}.$$

   Это делаем таким образом. В верхней половине матрицы смежности графа H приравниваем 1 элементы, соответствующие рёбрам треугольников $(v_{3n+1}, v_{3n+2}, v_{3n+3})$, …, $(v_{(3n)^2-2}, v_{(3n)^2-1}, v_{(3n)^2})$. Этих фиксированных элементов будет $(3n)^2 - 3n$. Остальные $((3n)^2 - 3n)((3n)^2 - 3n - 1)/2 - (3n)^2 + 3n$ элементов выбираем из 0 и 1 случайно и равновероятно.
2) Запускаем алгоритм $\mathcal{A}$ на графе $G' = G \cup H$.
3) Если $\mathcal{A}(G') = 1$, то по лемме 1 в графе G есть 3-разбиение. Выдаём ответ «ДА».
4) Если $\mathcal{A}(G') = 0$, то по лемме 1 в графе G нет 3-разбиения. Выдаём ответ «НЕТ».
5) Если $\mathcal{A}(G') = ?$, то выдаём ответ «НЕТ».

Заметим, что полиномиальный вероятностный алгоритм $\mathcal{B}$ выдаёт правильный ответ на шагах 3 и 4, а на шаге 5 может выдать неправильный ответ. Нужно доказать, что вероятность того, что ответ выдаётся на шаге 5, меньше 1/3.

Граф $G \cup H$ имеет $(3n)^2$ вершин, то есть его размер равен $m = ((3n)^4 - (3n)^2)/2$. Вероятность того, что для случайного графа $G \cup H$ имеет место $\mathcal{A}(G \cup H) = ?$, не больше

$$\frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|} = \frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\mathcal{G}_m|} \cdot \frac{|\mathcal{G}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|}.$$

Так как множество $\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}$ сильно пренебрежимое, то существует константа $\alpha > 0$, такая, что

$$\frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\mathcal{G}_m|} < \frac{1}{2^{\alpha m}} = \frac{1}{2^{\alpha((3n)^4-(3n)^2)/2}} = \frac{1}{2^{\alpha(81n^4-9n^2)/2}}$$

для любого n.

С другой стороны, так как матрица смежности графа H имеет $((3n)^2 - 3n)((3n)^2 - - 3n - 1)/2 - (3n)^2 + 3n$ элементов, которые можно выбирать произвольно, то

$$|\{G \cup H : H \in \mathcal{G}\}_m| = 2^{((3n)^2-3n)((3n)^2-3n-1)/2-(3n)^2+3n} = 2^{(81n^4-54n^3-18n^2+9n)/2}.$$

Отсюда

$$\frac{|\mathcal{G}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|} = \frac{2^{(81n^4-9n^2)/2}}{2^{(81n^4-54n^3-18n^2+9n)/2}} = 2^{(54n^3+9n^2-9n)/2}.$$

Поэтому искомая вероятность не больше

$$\frac{2^{(54n^3+9n^2-9n)/2}}{2^{\alpha(81n^4-9n^2)/2}} < \frac{1}{3}$$

при достаточно больших n. ■

Автор выражает благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

ЛИТЕРАТУРА

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 420 с.
2. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
3. *Гимади Э. Х., Глебов Н. И., Перепелица В. А.* Алгоритмы с оценками для задач дискретной оптимизации // Проблемы кибернетики. 1975. Т. 31. С. 35–42.
4. *Blum M.* How to prove a theorem so no one else can claim it // Proc. Intern. Congress Math., Berkeley, CA, 1986. P. 1444–1451.
5. *Myasnikov A. G. and Rybalov A. N.* Generic complexity of undecidable problems // J. Symbolic Logic. 2008. V. 73. No. 2. P. 656–673.
6. *Rybalov A. N.* On the strongly generic undecidability of the Halting Problem // Theor. Comput. Sci. 2007. V. 377. P. 268–270.
7. *Rybalov A. N.* Generic complexity of Presburger arithmetic // Theory Comput. Systems. 2010. V. 46. No. 1. P. 2–8.
8. *Rybalov A. N.* Generic complexity of the Diophantine problem // Groups Complexity Cryptology. 2013. V. 5. No. 1. P. 25–30.
9. *Rybalov A. N.* Generic hardness of the Boolean satisfiability problem // Groups Complexity Cryptology. 2017. V. 9. No. 2. P. 151–154.
10. *Рыбалов А. Н.* О генерической сложности проблемы кластеризации графов // Прикладная дискретная математика. 2019. № 46. С. 72–77.
11. *Рыбалов А. Н.* О генерической сложности проблемы распознавания гамильтоновых путей // Прикладная дискретная математика. 2021. № 53. С. 120–126.
12. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma // Proc. 29th STOC. El Paso: ACM, 1997. P. 220–229.
13. *Рыбалов А.* О генерической сложности проблемы общезначимости булевых формул // Прикладная дискретная математика. 2016. № 2(32). С. 119–126.

REFERENCES

1. *Garey M. and Johnson D.* Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, 1979. 340 p.
2. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 2003, vol. 264, no. 2, pp. 665–694.
3. *Gimadi E. H., Glebov N. I., and Perepelitsa V. A.* Algoritmy s otsenkami dlya zadach diskretnoy optimizatsii [Algorithms with bounds for problems of discrete optimization]. Problemy Kibernetiki, 1975, vol. 31, pp. 35–42. (in Russian)

4. *Blum M.* How to prove a theorem so no one else can claim it. Proc. Intern. Congress Math., Berkeley, CA, 1986, pp. 1444–1451.
5. *Myasnikov A. G. and Rybalov A. N.* Generic complexity of undecidable problems. J. Symbolic Logic, 2008, vol. 73, no. 2, pp. 656–673.
6. *Rybalov A. N.* On the strongly generic undecidability of the Halting Problem. Theor. Comput. Sci., 2007, vol. 377, pp. 268–270.
7. *Rybalov A. N.* Generic complexity of Presburger arithmetic. Theory Comput. Systems, 2010, vol. 46, no. 1, pp. 2–8.
8. *Rybalov A. N.* Generic complexity of the Diophantine problem. Groups Complexity Cryptology, 2013, vol. 5, no. 1, pp. 25–30.
9. *Rybalov A. N.* Generic hardness of the Boolean satisfiability problem. Groups Complexity Cryptology, 2017, vol. 9, no. 2, pp. 151–154.
10. *Rybalov A. N.* O genericheskoy slozhnosti problemy klasterizatsii grafov [On generic complexity of the graph clustering problem]. Prikladnaya Diskretnaya Matematika, 2019, no. 46, pp. 72–77. (in Russian)
11. *Rybalov A. N.* O genericheskoy slozhnosti problemy raspoznavaniya gamil'tonovykh putey [The general complexity of the problem to recognize Hamiltonian paths]. Prikladnaya Diskretnaya Matematika, 2021, no. 53, pp. 120–126. (in Russian)
12. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC, El Paso, ACM, 1997, pp. 220–229.
13. *Rybalov A. N.* O genericheskoy slozhnosti problemy obshcheznachimosti bulevykh formul [On generic complexity of the validity problem for Boolean formulas]. Prikladnaya Diskretnaya Matematika, 2016, no. 2(32), pp. 119–126. (in Russian)

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.8 DOI 10.17223/20710410/58/11

ПРИМЕНЕНИЕ МЕТОДОВ ИДЕМПОТЕНТНОЙ АЛГЕБРЫ В ГЕНЕТИЧЕСКОМ АЛГОРИТМЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ[1]

А. М. Булавчук, Д. В. Семенова

Сибирский федеральный университет, г. Красноярск, Россия

E-mail: abulavchuk@sfu-kras.ru, DVSemenova@sfu-kras.ru

Рассматривается задача календарного планирования инвестиционных проектов с ограниченными ресурсами в денежной форме. Критерием оптимального расписания начала для каждой из работ проекта выступает максимум чистой приведённой стоимости, при котором выполняются ограничения на достаточность средств и учитываются технологические взаимосвязи между работами. Данная задача является NP-трудной в сильном смысле. Доказано, что расписание проекта представимо в виде решения линейного уравнения над идемпотентным полукольцом. Установлено достаточное условие допустимости расписания с точки зрения частичного порядка работ и срока реализации проекта. Доказано, что каждое из расписаний проекта может быть представлено в виде произведения матрицы специального вида, рассчитанной на основе матрицы частичного порядка проекта, и вектора из идемпотентого полумодуля. Для координат вектора определены ограничения сверху и снизу, учитывающие сроки реализации работ. Приводится описание генетического алгоритма решения задачи. В его основе лежит эволюция популяции, особи которой представляют собой решения идемпотентного уравнения для матрицы частичного порядка проекта. Проведённые вычислительные эксперименты демонстрируют эффективность предложенного алгоритма.

Ключевые слова: *задача календарного планирования, инвестиционный проект, NPV, идемпотентная алгебра, генетический алгоритм.*

APPLICATION OF IDEMPOTENT ALGEBRA METHODS IN GENETIC ALGORITHM FOR SOLVING THE SCHEDULING PROBLEM

A. M. Bulavchuk, D. V. Semenova

Siberian Federal University, Krasnoyarsk, Russia

The resource-constrained project scheduling problem in monetary form is considered. The criterion for the optimal start schedule for each project activity is the maximum net present value, which fulfills the constraints on sufficiency of funds and takes into

[1]Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ (Соглашение №075-02-2022-876).

account the technological relationship between the activities. This problem is NP-hard in a strong sense. It is proved that the project schedule can be represented as a solution of a linear equation over an idempotent semiring. A sufficient condition has been established for the admissibility of the schedule in terms of the partial order of work and the duration of the project. It is proved that each of the project schedules can be represented as a product of a matrix of a special form, calculated on the basis of the partial order matrix of the project, and a vector from an idempotent semimodule. For the coordinates of the vector, upper and lower limits have been determined, taking into account the timing of the activity. A description of the genetic algorithm for solving the problem is given. The algorithm is based on the evolution of a population whose individuals represent solutions of an idempotent equation for a partial order matrix of the project. The computational experiments demonstrate the effectiveness of the algorithm.

Keywords: *scheduling problem, investment project, NPV, idempotent mathematics, genetic algorithm.*

Введение

Одной из важных задач при разработке и реализации инвестиционных проектов является выбор оптимального расписания взаимосвязанных работ, которое удовлетворяет всем технологическим ограничениям, а также обеспечивает рациональное использование производственных ресурсов. Критерием оптимальности в этом случае могут выступать временные (длительность реализации проекта), денежные (совокупные затраты, прибыль) или иные характеристики проекта (эффективность использования ресурсов, экологичность производства) [1–4].

Практически важной разновидностью задачи календарного планирования являются постановки, в которых ресурсы проекта описываются денежными потоками. Обзор таких постановок приводится в [5, 6]. Одно из отличий в подходах заключается в выборе метода дисконтирования. Однако используемые в разных работах формулы непрерывных и сложных процентов легко конвертируются друг в друга. Важным отличием подходов являются особенности учёта компонентов потоков платежей. Например, эти потоки могут рассматриваться как регулярные или нерегулярные. Кроме того, существуют отличия в выборе целевой функции. Традиционно используемая в этом качестве прибыль может быть скорректирована, например, на величину штрафов за нарушение сроков выполнения работ.

В работе рассматривается задача календарного планирования инвестиционных проектов с ограниченными ресурсами в постановке, предложенной в [7]. В качестве критерия оптимальности выступает стоимость прибыли, приведённая к началу реализации проекта. Предполагается, что на множестве работ задан частичный порядок. Единственным ресурсом проекта выступает финансовый. Известны запасы средств, потребности в финансировании работ и доход, генерируемый ими, в каждый из целочисленных моментов времени. Деньги являются складируемым ресурсом. Кроме того, существует возможность получения дополнительного дохода от владения свободными средствами по заданной безрисковой ставке. Ограничения на запасы средств могут быть сняты введением возможности кредитования. Это означает также дополнительные затраты, связанные с выплатой процентов. Данная задача является NP-трудной в сильном смысле [7]. В [8] приведены алгоритмы решения этой задачи методом ветвей и границ, методами динамического программирования, а также гибридный алгоритм, объединяющий эти алгоритмы на разных этапах решения.

Данная работа является продолжением исследований по разработке эффективного генетического алгоритма для решения задачи календарного планирования инвестиционных проектов с ограниченными ресурсами в денежной форме [9]. Для упрощения применения генетических операторов предлагается использовать методы идемпотентной алгебры. Анализ свойств матрицы частичного порядка инвестиционного проекта позволил вывести соотношения, упрощающие генерацию популяции. Показано, что в терминах идемпотентной алгебры расписание проекта представимо в виде решения линейного уравнения над идемпотентным полукольцом. Установлено достаточное условие допустимости расписания с точки зрения частичного порядка работ и срока реализации проекта. Для формирования популяции предлагается использовать векторы из идемпотентного полумодуля, удовлетворяющие достаточному условию допустимости расписания. Приводятся результаты вычислительных экспериментов, подтверждающие результативность предложенного генетического алгоритма GASPIA (Genetic Algorithm for resource-constrained project Scheduling Problem using Idempotent Algebra methods).

1. Постановка задачи календарного планирования

Рассмотрим инвестиционный проект, предполагающий выполнение заданного количества работ за определённый промежуток времени. Каждая из работ характеризуется длительностью её выполнения, затратами ресурсов и доходами, которые она генерирует. Кроме того, известны технологические и организационные взаимосвязи между работами. Это, в частности, может выражаться в необходимости завершения одной работы до начала выполнения другой.

Источниками финансирования проекта являются собственные средства организации, а также доходы, возникающие в процессе его реализации. Для простоты будем считать, что все ресурсы проекта могут быть сведены к денежным. Бюджет проекта известен заранее. Будем считать, что в отсутствие достаточного объёма средств на выполнение любой из работ проект должен быть остановлен. Для оценки результатов реализации проекта воспользуемся показателем чистой приведённой стоимости (NPV). Данный показатель представляет собой стоимость прибыли от всех запланированных работ, дисконтированную к моменту начала инвестирования. Таким образом, рассматриваемая задача календарного планирования инвестиционного проекта заключается в нахождении оптимального времени начала каждой из работ, при котором выполняются ограничения на достаточность средств, взаимосвязь работ и обеспечивается максимум NPV.

Приведём математическую постановку задачи. Пусть $V = \{1, 2, \ldots, N\}$ — множество запланированных работ, связанных друг с другом отношением частичного порядка E. Каждая работа $j = 1, 2, \ldots, N$ характеризуется длительностью $q_j \in \mathbb{Z}^+$. Обозначим через $c_j(\tau)$ величину поступлений от j-й работы в момент времени $\tau \in \in \{0, 1, \ldots, q_j\}$. Для отрицательных значений $c_j(\tau)$ речь будет идти о расходах на выполнение работы.

Введём следующие обозначения:

— T — максимальная длительность проекта в целых периодах;
— $K(t)$ — бюджет проекта в момент $t \in \{0, 1, \ldots, T-1\}$;
— r_0 — ставка дисконтирования для одного периода;
— N_t — множество работ, выполняемых в интервале $[t; t+1)$.

Будем искать оптимальные сроки начала каждой из работ, которые образуют расписание выполнения проекта $S = (s_1, s_2, \ldots, s_N)$, где s_j — момент начала j-й работы. Очевидно, что $s_j \in \{0, 1, \ldots, T - q_j\}$.

При применении формулы сложных процентов чистый приведённый доход от j-й работы рассчитывается по формуле

$$NPV_j = \sum_{\tau=0}^{q_j} \frac{c_j(\tau)}{(1 + r_0)^\tau}. \tag{1}$$

Представленная ниже математическая постановка оптимизационной задачи сформулирована В. В. Сервахом и Е. А. Казаковцевой [7]:

$$NPV(S) = \sum_{j \in V} \frac{c_j(\tau)}{(1 + r_0)^{\tau + s_j}} \to \max_S; \tag{2}$$

$$s_i + q_i \leqslant s_j, \quad (i, j) \in E; \tag{3}$$

$$\sum_{t=0}^{t^*} \frac{K(t)}{(1 + r_0)^t} + \sum_{t=0}^{t^*} \sum_{j \in N_t} \frac{c_j(t - s_j)}{(1 + r_0)^t} \geqslant 0, \quad t^* = 0, 1, \ldots, T - 1. \tag{4}$$

Ограничениями в модели выступают условия частичного порядка (3) и требования достаточности средств (4). В последнем случае для каждого момента времени t^* рассчитывается баланс расходов и поступлений с учётом возможности реинвестирования доходов предшествующих периодов. Допустимыми по бюджету считаются только те расписания, для которых названный баланс неотрицателен. Будем называть расписание допустимым по срокам, если для него выполняются органичения (3), а проект завершается до конца директивного срока. Расписание может быть допустимым по сроку, но недопустимым по бюджету.

Величина $NPV(S)$ характеризует приведённую стоимость прибыли за весь срок реализации проекта при выбранном расписании S. Отметим также, что при заданном расписании S целевая функция $NPV(S)$ есть линейная комбинация величин поступлений от каждой из работ:

$$NPV(S) = \sum_{j=1}^{N} \sum_{\tau=0}^{q_j} \alpha_j(\tau) c_j(\tau).$$

Здесь $\alpha_j(\tau) = (1 + r_0)^{-(\tau + s_j)}$.

2. Достаточное условие допустимости расписания проекта по срокам

2.1. Основные определения и обозначения идемпотентной алгебры

Одним из популярных подходов к решению оптимизационных задач является применение методов идемпотентной математики. Основной алгебраической структурой идемпотентного анализа является общее определение идемпотентного полукольца [10]. Рассмотрим множество $\mathbb{X}$, на котором определены операции сложения $\oplus$ и умножения $\odot$. Тройка $\{\mathbb{X}, \oplus, \odot\}$ называется идемпотентным полукольцом, если для любых элементов множества выполняются следующие свойства:

— сложение и умножение ассоциативны;
— сложение коммутативно;
— сложение идемпотентно;
— умножение дистрибутивно относительно сложения.

Единичным будем называть такой элемент $\mathbb{1} \in \mathbb{X}$, что

$$x \odot \mathbb{1} = \mathbb{1} \odot x = x$$

для всех $x \in \mathbb{X}$.

Нулевым будем называть такой элемент $\mathbb{0} \in \mathbb{X}$, что

$$x \oplus \mathbb{0} = \mathbb{0} \oplus x = x$$

для всех $x \in \mathbb{X}$.

Идемпотентным полумодулем над полукольцом $\mathbb{X}$ будем называть декартово произведение $\mathbb{X}^l$, для которого определены операции идемпотентного ассоциативного коммутативного сложения и умножения слева на скаляр, удовлетворяющие свойствам ассоциативности и дистрибутивности.

Пусть $A \in \mathbb{X}^{n\times n}$ — квадратная матрица. Определим матрицы A^+ и $A^\times$ как

$$A^+ = I \oplus A \oplus \cdots \oplus A^{n-1}; \tag{5}$$

$$A^\times = A \oplus A^2 \oplus \cdots \oplus A^n = A \cdot A^+, \tag{6}$$

где I — единичная матрица. Определим также матрицу A^*, которая строится заменой на нулевые тех столбцов матрицы A^+, которые не совпадают со столбцами матрицы $A^\times$.

Будем называть псевдообратной для матрицы A матрицу A^-, элементы которой равны

$$a_{ij}^- = \begin{cases} a_{ji}^{-1}, & \text{если } a_{ji} \neq \mathbb{0}, \\ \mathbb{0}, & \text{если } a_{ji} = \mathbb{0}. \end{cases}$$

Будем называть матрицу A разложимой, если перестановкой столбцов вместе с такой же перестановкой строк она может быть приведена к блочно-диагональному виду

$$A = \begin{pmatrix} A_{11} & \mathbf{0} & \dots & \mathbf{0} \\ A_{21} & A_{22} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mm} \end{pmatrix}, \tag{7}$$

где $\mathbf{0}$ — блоки, состоящие из $\mathbb{0}$. Такой вид будем называть нормальной формой разложимой матрицы.

Утверждение 1 [11]. Нормальная форма разложимой матрицы A может быть представлена в виде суммы $A = F \oplus D$, где

$$F = \begin{pmatrix} \mathbf{0} & \dots & \dots & \mathbf{0} \\ A_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A_{m1} & \dots & A_{m,m-1} & \mathbf{0} \end{pmatrix}, \qquad D = \begin{pmatrix} A_{11} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & A_{mm} \end{pmatrix}.$$

Вспомогательные матрицы D^+, D^* рассчитываются по формулам (5) и (6); введём матрицу $G = D^+F$.

Обозначим через $\operatorname{Tr} A$ сумму всех циклических произведений элементов матрицы A. Согласно [11],

$$\operatorname{Tr} A = \bigoplus_{m=1}^{N} \operatorname{tr} A^m, \tag{8}$$

где $\operatorname{tr} A$ — след матрицы A.

Приведём без доказательств несколько вспомогательных теорем.

Теорема 1 [11]. Пусть $x \in \mathbb{X}^n$ — общее решение уравнения $A \cdot x = x$ с матрицей A, представленной в форме (7). Тогда справедливы следующие утверждения:

1) если $\operatorname{Tr} A < \mathbb{1}$, то уравнение имеет только тривиальное решение $x = (\mathbb{0}, \ldots, \mathbb{0})$;
2) если $\operatorname{Tr} A = \mathbb{1}$, то $x = G^+ D^*$ для всех $v \in \mathbb{X}^n$;
3) если $\operatorname{Tr} A > \mathbb{1}$, то $x = G^+ D^*$ для всех $v \in \mathbb{X}^n$, причём имеется только тривиальное решение $x = (\mathbb{0}, \ldots, \mathbb{0})$, когда $\mathcal{I}_0 = \varnothing$, где $\mathcal{I}_0$ — множество индексов i, для которых $\operatorname{Tr} A_{ii} = \mathbb{0}$.

Теорема 2 [11]. Для любой матрицы A и вектора $b \neq (\mathbb{0}, \ldots, \mathbb{0})$ решение неравенства $Ax \leqslant b$ имеет вид $x \leqslant (b^- A)^-$.

2.2. Поиск допустимых решений

Докажем, что в терминах идемпотентной алгебры поиск допустимых расписаний можно свести к решению линейных уравнений.

Отношение частичного порядка работ проекта может быть представлено в виде взвешенного ориентированного графа с матрицей смежности $A = (a_{ij})_{i,j=1}^N$, где

$$a_{ij} = \begin{cases} q_j, & \text{если } (i,j) \in E, \\ 0, & \text{если } (i,j) \notin E \text{ и } i = j, \\ -\infty, & \text{если } (i,j) \notin E \text{ и } i \neq j. \end{cases}$$

Рассмотрим полукольцо $\mathbb{R}_{\max,+}$ с операциями $\oplus = \max$ и $\odot = +$. В этом полукольце нулевой элемент $\mathbb{0} = -\infty$, единичный элемент $\mathbb{1} = 0$, а элементы матрицы A описываются соотношениями

$$a_{ij} = \begin{cases} q_j, & \text{если } (i,j) \in E, \\ \mathbb{1}, & \text{если } (i,j) \notin E \text{ и } i = j, \\ \mathbb{0}, & \text{если } (i,j) \notin E \text{ и } i \neq j. \end{cases}$$

Лемма 1. Матрица, описывающая частичный порядок работ инвестиционного проекта, является разложимой, а её нормальная форма имеет нижний треугольный вид.

Доказательство. Ориентированный граф называется сильносвязным, если любые две его вершины являются связанными [12]. Матрица является разложимой, если соответствующий ей граф не является сильносвязным [11, с. 26]. Ориентированный граф, соответствующий матрице A, не является сильносвязным, поскольку наличие прямого пути между вершинами графа проекта означает отсутствие обратного пути. Следовательно, матрица A является разложимой.

Выполним преобразования, необходимые для приведения к нормальной форме, для всех i и j, для которых $j > i$ и $a_{ij} > \mathbb{0}$. Каждому ненулевому a_{ij} соответствует нулевое значение a_{ji}. Таким образом, итогом преобразований станет нижняя треугольная матрица. ■

Лемма 2. Нормальная форма матрицы, описывающая частичный порядок работ инвестиционного проекта, представима в виде $A = I \oplus F$.

Доказательство. Из утверждения 1 следует, что нормальная форма матрицы частичного порядка работ проекта представима в виде $A = D \oplus F$. Согласно лемме 1, нормальная форма матрицы частичного порядка работ проекта имеет нижний треугольный вид. Тогда на диагонали матрицы D в разложении $A = D \oplus F$ стоят блоки единичной размерности. Пусть в процессе приведения матрицы к нормальной форме мы поменяли местами столбцы i и j, а после этого переставили строки i и j. После таких преобразований элемент a_{ii} исходной матрицы займёт место элемента a_{jj} и наоборот. Таким образом, диагональные элементы матрицы в нормальной форме останутся равными $\mathbb{1}$. Отсюда следует, что D — диагональная матрица, все диагональные элементы которой равны $\mathbb{1}$. Значит, $D = I$ и $A = I \oplus F$. ■

Для каждой работы $j \in \{1, \ldots, N\}$ выполнение условий частичного порядка равносильно выполнению в смысле полукольца $\mathbb{R}_{\max,+}$ неравенства

$$s_j \geqslant a_{j1}s_1 \oplus a_{j2}s_2 \oplus \cdots \oplus a_{jN}s_N. \tag{9}$$

Для полумодуля $\mathbb{R}^N_{\max,+}$ неравенство (9) можно переписать в матричном виде:

$$A \cdot S \leqslant S.$$

Для нахождения расписания с минимальными возможными сроками начала работ рассмотрим уравнение

$$A \cdot S = S. \tag{10}$$

Лемма 3. Уравнение (10) имеет нетривиальное решение, если матрица A описывает частичный порядок работ инвестиционного проекта.

Доказательство. Согласно теореме 1, нетривиальное решение уравнения (10) для произвольной матрицы A существует, если выполняется равенство $\operatorname{Tr} A = \mathbb{1}$. В смысле полукольца $\mathbb{R}_{\max,+}$ след матрицы A^m — максимальное расстояние вдоль всех циклических путей длины m. Поскольку речь идёт о работах в рамках проекта, циклические пути длины, отличной от нуля, отсутствуют. В противном случае каждая работа в цикле будет одновременно предшествовать другим работам и следовать за ними. Нулевому расстоянию вдоль циклических путей соответствует $\operatorname{Tr} A = \mathbb{1}$. Значит, условие существования нетривиального решения гарантированно выполняется. ■

Теорема 3. Решение уравнения (10) представимо в виде

$$S = A^+ \upsilon, \tag{11}$$

где υ — произвольный вектор из $\mathbb{X}^N$.

Доказательство. Согласно лемме 1, матрица частичного порядка работ инвестиционного проекта является разложимой, а её нормальная форма может быть представлена в виде $A = D \oplus F$. Согласно лемме 3, для матрицы частичного порядка работ инвестиционного проекта уравнение (10) имеет нетривиальное решение. По теореме 1 для нормальной формы разложимой матрицы A искомый вектор S находится по формуле $S = G^+ D^* \upsilon$. Так как по лемме 2 для рассматриваемой матрицы проекта

$D = I$, то $D^+ = D^\times = D^* = I$. Отсюда $G = D^+F = F$, а $G^+D^* = F^+$. С другой стороны, так как $A = F \oplus I$, то

$$A^+ = I \oplus (F \oplus I) \oplus \cdots \oplus (F \oplus I)^{N-1} = I \oplus F \oplus \cdots \oplus F^{N-1} = F^+.$$

Следовательно, $G^+D^* = A^+$ и $S = A^+\upsilon$. ■

Теорема 4 (достаточное условие допустимости расписания проекта по срокам). Пусть A — матрица частичного порядка работ инвестиционного проекта в нормальной форме, $Q = (q_1, q_2, \ldots, q_N)$ — вектор длительностей выполнения работ, а T — директивный срок завершения проекта. Тогда расписание $S = A^+\upsilon$ допустимо с точки зрения частичного порядка и срока реализации проекта, если для всех $j = 1, \ldots, N$

$$\upsilon_j^{\min} \leqslant \upsilon_j \leqslant \upsilon_j^{\max}, \tag{12}$$

где $\upsilon^{\min} = (0, \ldots, 0)$, $\upsilon^{\max} = (b^- A^+)^-$ и b — строка длины N, j-й элемент которой равен $T - q_j$.

Доказательство. Чтобы j-я работа была завершена в срок, она должна быть начата не позднее периода $T - q_j$. Составим из предельных сроков начала строку b. Поскольку, согласно теореме 3, расписание S представимо в виде (11), можем записать неравенство $A^+\upsilon \leqslant b$. Согласно теореме 2, решением неравенства является вектор $\upsilon \leqslant (b^- A^+)^-$. Отсюда $\upsilon^{\max} = (b^- A^+)^-$. ■

3. Генетический алгоритм GASPIA

3.1. Описание алгоритма

Задача календарного планирования с критерием максимизации приведённой прибыли является NP-трудной в сильном смысле [7]. Не вызывает сомнения, что применение эвристических методов к решению задачи является перспективным направлением исследований [13], в частности, следует отметить успешный опыт решения задачи календарного планирования с временным критерием с помощью генетического алгоритма [3]. Для решения задачи (1)–(4) предлагается генетический алгоритм GASPIA, впервые представленный в [9]. Моделирование эволюции исходной популяции в GASPIA осуществляется на основе результатов теорем 3 и 4. Опишем этапы алгоритма.

Каждую особь в популяции будем характеризовать с помощью генотипа и фенотипа. Под генотипом понимается набор независимых характеристик особи, интересующий нас в контексте задачи. Фенотип — это характеристика особи, определённая как функция от её генотипа. Пусть особи популяции представляют собой допустимые расписания работ по проекту. Тогда в качестве генотипа можно было бы рассматривать векторы $S = (s_1, s_2, \ldots, s_N)$, компоненты которых определяют время начала соответствующей работы. Однако такой вариант генотипа затрудняет выполнение скрещиваний, поскольку компоненты родительских генотипов не могут быть произвольно смешаны без нарушения условий частичного порядка. В связи с этим в качестве генотипа особей будем рассматривать векторы υ, координаты которых удовлетворяют соотношениям (12).

Такие векторы позволяют получать расписания, удовлетворяющие требованию порядка выполнения работ и предельному сроку завершения проекта. В процессе формирования исходной популяции необходимо сгенерировать нужное число векторов, удовлетворяющих соотношениям (12). Соответствующее вектору υ расписание S будем определять с помощью формулы (11).

Помимо соответствия расписаний условиям частичного порядка необходимо обеспечить также выполнение бюджетных ограничений (4). Это можно сделать, например, включив штраф за нарушение ограничений в функцию приспособленности. Тогда в качестве фенотипа будет выступать размер NPV проекта, скорректированный на величину штрафа:

$$F(S) = \sum_{j=1}^{N} \sum_{\tau=0}^{q_j} \frac{c_j(\tau)}{(1+r_0)^{s_j+\tau}} - M \cdot \delta(S), \tag{13}$$

где $M > 0$ — достаточно большое число, а $\delta(S)$ равно 1, если хотя бы одно из ограничений (4) не выполняется. В противном случае $\delta(S) = 0$. Если бюджетные ограничения не слишком жёсткие, то такой подход позволит исключать недопустимые по бюджету расписания на стадии отбора особей с наибольшим значением функции приспособленности. При малом количестве расписаний, допустимых по бюджету, применение генетического алгоритма будет затруднено ввиду отсутствия генетического разнообразия.

Рассмотрим особенности нахождения оптимального решения в зависимости от прибыльности или убыточности отдельных работ. Введём вспомогательный вектор C_0, координаты которого рассчитываются по следующей формуле:

$$C_0(j) = \begin{cases} 1, & \text{если } NPV_j \geqslant 0, \\ -1, & \text{если } NPV_j < 0. \end{cases}$$

Если все компоненты вектора C_0 положительны, то все работы проекта приносят прибыль. Отсюда следует, что максимальное значение NPV по такому проекту обеспечивает расписание, удовлетворяющее бюджетным ограничениям, с минимальными возможными сроками начала работ. Для прибыльных работ вклад в NPV будет тем больше, чем раньше начинается работа, поскольку дисконтирующие множители находятся в прямой зависимости от срока начала работы. Если бюджетным ограничениям соответствует расписание $S^{\min} = A^{+}\upsilon^{\min}$, то именно оно обеспечивает оптимум целевой функции.

Если все компоненты C_0 отрицательны, то все работы убыточны, а стоимость затрат можно уменьшить, отодвинув работы как можно дальше к концу срока. Очевидно, что минимум целевой функции в этом случае обеспечивает вектор $S^{\max} = A^{+}\upsilon^{\max}$. Если расписание $S^{\max}$ не удовлетворяет бюджетным ограничениям, то и все прочие расписания не будут допустимыми.

Описанные варианты проектов на практике встречаются редко. Для проектов сложной структуры часть работ предполагает только несение затрат на реализацию, тогда как другая часть обеспечивает поступление доходов. Учтём этот факт при построении правил скрещивания особей.

Процедура скрещивания. При формировании генотипа потомков будем сравнивать компоненты генотипов родителей для одноимённых работ. Для пары родителей υ^k и υ^l генотип потомка будет задаваться вектором $\upsilon^{k\times l}$ с компонентами

$$\upsilon_j^{k\times l} = \left|\min\left\{C_0(j)\,\upsilon_j^k,\ C_0(j)\,\upsilon_j^l\right\}\right|. \tag{14}$$

Особи, появившиеся в результате скрещивания, соответствуют допустимым с точки зрения порядка работ расписаниям, но могут не соответствовать бюджетным ограничениям. На рис. 1 приведён пример скрещивания особей υ^k и υ^l. Белые круги соответствуют работам, для которых $NPV_j < 0$, а серые — работам, для которых $NPV_j \geqslant 0$. Применяя к векторам υ^k и υ^l формулу (14), получим вектор-потомок $\upsilon^{k\times l}$.

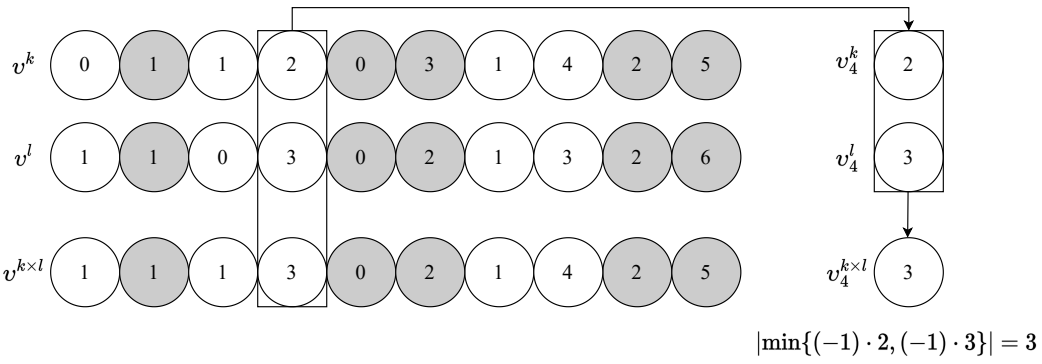

Рис. 1. Пример скрещивания особей v^k и v^l. Белые круги соответствуют работам, для которых $C_0 = -1$

Процедура мутации. Оператор мутации предназначен для внесения разнообразия в популяцию с целью избежать преждевременной сходимости алгоритма. В настройках алгоритма зададим долю особей популяции, генотип которых будет подвергаться мутации. В ходе применения оператора мутации будем заменять один случайно выбранный элемент генотипа так, чтобы выполнялись соотношения (12). Эта процедура также не ведёт к нарушению условий частичного порядка. Расписания, не удовлетворяющие бюджетным ограничениям, будут отсеиваться на стадии отбора лучших особей. Алгоритм мутации может быть описан следующим образом:

1) Выбрать в популяции вектор v.
2) Сгенерировать случайное число $rnd_v \in [0, 1]$.
3) Если $rnd_v < p_{\text{mut}}$, где p_{mut} — заданная вероятность мутации, выбрать случайный элемент вектора v_j.
4) Заменить элемент v_j случайным числом $v_j^{\min} \leqslant rnd_v_j \leqslant v_j^{\max}$.
5) Повторить для всех особей популяции.

Отбор лучших особей. После применения генетических операторов число особей в популяции гарантированно возрастает. Кроме того, меняются характеристики особей, подвергнувшихся мутациям. Чтобы определить наиболее жизнеспособные особи, отсортируем популяцию в порядке убывания функции приспособленности (13). В популяцию следующего поколения попадает заданное число лучших особей. Поскольку функция приспособленности включает штраф за нарушение бюджетных ограничений, в следующее поколение будут преимущественно попадать особи, для которых ограничения не нарушаются. Критерием остановки алгоритма будем считать прохождение популяции через заданное число поколений.

3.2. Вычислительные эксперименты

Для проверки работы генетического алгоритма GASPIA написана программа на высокоуровневом языке программирования Python версии 3.9 в среде разработки IDLE. Все эксперименты проводились на персональном компьютере с процессором 2,3 GHz CPU и 8 Gb RAM с операционной системой Windows 10.

Вычислительные эксперименты проводились на моделях проектов из базы тестовых задач PSLIB [14]. Для примеров из набора j14 генерировались потоки платежей. Параметры потоков были подобраны таким образом, чтобы в одном случае все работы оказывались прибыльными, в другом были представлены как прибыльные, так и убыточные работы. Для серий из 100 экспериментов получены средние значения целевой функции в зависимости от типа потока платежей. Число поколений равно 50, а чис-

ло особей в популяции — 100. Результаты экспериментов приведены в табл. 1. Точные решения поставленной задачи были найдены с помощью пакета программ IBM ILOG CPLEX. Для случая, когда все работы являются прибыльными, точному решению соответствует $NPV = 46{,}654$. Для смешанного варианта максимальное значение NPV составляет 31,224.

Таблица 1

Результаты экспериментов для проекта из 14 работ

Тип потока	Максимум NPV	Средняя NPV	Отклонение, %
Прибыльный	46,654	45,859	1,705
Смешанный	31,224	30,830	1,263

Результаты экспериментов показывают, что алгоритм демонстрирует высокую точность даже для небольшого числа поколений и особей в популяции. В серии экспериментов были получены в том числе расписания, соответствующие точным решениям. Для смешанного потока платежей оценки влияния параметров алгоритма на отклонение от наилучшего решения приведены в табл. 2. Из табл. 2 видно, что с ростом числа поколений и размера популяций существенно улучшается качество найденных решений.

Таблица 2

Отклонение от точного решения, %

Поколения	Особи		
	50	100	500
10	7,358	4,243	0,946
20	4,69	2,271	0,157
50	2,648	1,263	0,000

Обратим внимание также на структуру популяции с точки зрения допустимости расписаний по бюджету. На рис. 2 приведена динамика доли недопустимых расписаний в популяции для одного из экспериментов. Если в начальной популяции эта доля достаточно высока, то уже за первые несколько поколений она падает почти до нуля. В результате скрещивания и мутаций в дальнейшем в популяции продолжают появляться недопустимые расписания, но отбор позволяет их исключать.

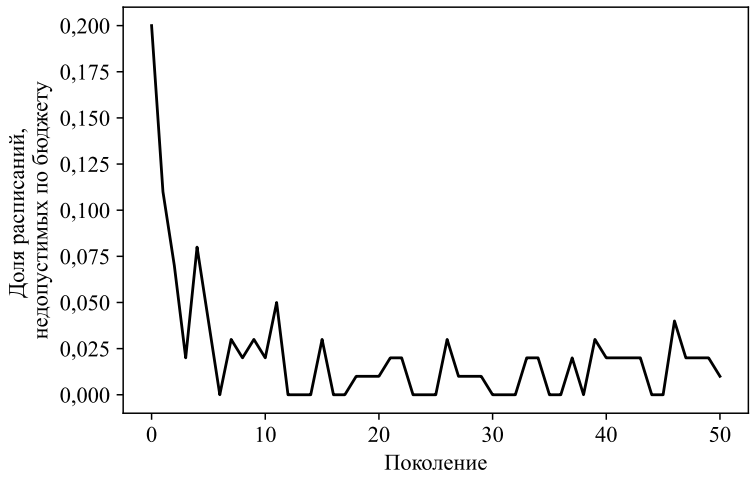

Рис. 2. Изменение доли допустимых по сроку, но недопустимых по бюджету расписаний

Благодаря включению штрафов за нехватку средств в функцию приспособленности алгоритм обеспечивает преимущество допустимым по бюджету расписаниям. Однако если таких расписаний в популяции мало, результативность генетического алгоритма снижается.

ЛИТЕРАТУРА

1. *Гимади Э. Х., Пузынина Н. М.* Задача календарного планирования крупномасштабного проекта в условиях ограниченных ресурсов: Опыт построения математического обеспечения // Управляемые системы. 1983. Вып. 23. С. 24–32.
2. *Гимади Э. Х., Залюбовский В. В., Севастьянов С. В.* Полиномиальная разрешимость задач календарного планирования со складируемыми ресурсами и директивными сроками // Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7. Вып. 1. С. 9–34.
3. *Гончаров Е. Н., Леонов В. В.* Генетический алгоритм для задачи календарного планирования с ограниченными ресурсами // Автоматика и телемеханика. 2017. № 6. С. 173–179.
4. *Гончаров Е. Н., Мишин Д. В.* Точный алгоритм для задачи календарного планирования с ограниченными ресурсами // Прикладная математика и фундаментальная информатика. 2017. Т. 4. № 1. С. 43–53.
5. *Hartmann S. and Briskorn D.* An updated survey of variants and extensions of the resource-constrained project scheduling problem // Eur. J. Operat. Res. 2022. Iss. 1. P. 1–14.
6. *Habibi F., Barzinpour F., and Sadjadi S.* Resource-constrained project scheduling problem: review of past and recent developments // J. Project Management. 2018. V. 3. Iss. 2. P. 55–88.
7. *Казаковцева Е. А., Сервах В. В.* Сложность задачи календарного планирования с кредитованием // Дискретный анализ и исследование операций. 2015. Т. 22. № 4. С. 35–49.
8. *Сервах В. В., Сухих С. Л.* Гибридный алгоритм для задачи календарного планирования с учетом реинвестирования прибыли // Автоматика и телемеханика. 2004. Вып. 3. С. 100–107.
9. *Bulavchuk A. M. and Semenova D. V.* Genetic algorithm based on idempotent algebra methods for RCPSP // IEEE 15th Intern. Conf. AICT. 2021. P. 1–4.
10. *Литвинов Г. Л., Маслов В. П., Соболевский А. Н.* Идемпотентная математика и интервальный анализ // Вычислительные технологии. 2001. Т. 6. № 6. С. 41–70.
11. *Кривулин Н. К.* Методы идемпотентной алгебры в задачах моделирования и анализа сложных систем. СПб.: Изд-во С.-Петерб. ун-та, 2009. 256 с.
12. *Омельченко А. В.* Теория графов. М.: МЦНМО, 2018. 416 с.
13. *Кочетов Ю. А., Столяр А. А.* Новые жадные эвристики для задачи календарного планирования с ограниченными ресурсами // Дискретный анализ и исследование операций. 2005. Т 12. № 1. С. 12–36.
14. `https://www.om-db.wi.tum.de/psplib/` — Project scheduling problem library. 2022.

REFERENCES

1. *Gimadi E. Kh. and Puzynina N. M.* Zadacha kalendarnogo planirovaniya krupnomasshtabnogo proyekta v usloviyakh ogranichennykh resursov: Opyt postroyeniya matematicheskogo obespecheniya [Problem of the calendar planning of a large-scale design under the conditions of limited resources: experience in the construction of software]. Upravliaemie Systemy, 1983, no. 23, pp. 24–32. (in Russian)
2. *Gimadi E. Kh., Zalyubovskii V. V., and Sevast'yanov S. V.* Polinomial'naya razreshimost' zadach kalendarnogo planirovaniya so skladiruyemymi 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)

3. *Goncharov E. N. and Leonov V. V.* Geneticheskiy algoritm dlya zadachi kalendarnogo planirovaniya s ogranichennymi resursami [Genetic algorithm for the resource-constrained project scheduling problem]. Avtomatika i Telemekhanika, 2017, no. 6, pp. 173–179. (in Russian)
4. *Goncharov E. N. and Mishin D. V.* Tochnyy algoritm dlya zadachi kalendarnogo planirovaniya s ogranichennymi resursami [Exact algorithm for the resource-constrained project scheduling problem]. Prikladnaya Matematika i Fundamental'naya Informatika, 2017, vol. 4, no. 1, pp. 43–53. (in Russian)
5. *Hartmann S. and Briskorn D.* An updated survey of variants and extensions of the resource-constrained project scheduling problem. Eur. J. Operat. Res., 2022, iss. 1, pp. 1–14.
6. *Habibi F., Barzinpour F., and Sadjadi S.* Resource-constrained project scheduling problem: review of past and recent developments. J. Project Management, 2018, vol. 3, iss. 2, pp. 55–88.
7. *Kazakovtseva E. A. and Servakh V. V.* Slozhnost' zadachi kalendarnogo planirovaniya s kreditovaniyem [Complexity of the project scheduling problem]. Diskretn. Analiz i Issled. Oper., 2015, vol. 22, no. 4, pp. 35–49. (in Russian)
8. *Servakh V. V. and Soukhikh S. L.* Gibridnyy algoritm dlya zadachi kalendarnogo planirovaniya s uchetom reinvestirovaniya pribyli [Hybrid algorithm for scheduling with regard for reinvestment of profits]. Avtomatika i Telemekhanika, 2004, no. 3, pp. 100–107. (in Russian)
9. *Bulavchuk A. M. and Semenova D. V.* Genetic Algorithm Based on Idempotent Algebra Methods for RCPSP. 2021 IEEE 15th Intern. Conf. AICT, 2021, pp. 1–4.
10. *Litvinov G. L., Maslov V. P., and Sobolevsky A. N.* Idempotentnaya matematika i interval'nyy analiz [Idempotent mathematics and interval analysis]. Vychislitel'nye Tekhnologii, 2001, vol. 6. no. 6. pp. 41–70. (in Russian)
11. *Krivulin N. K.* Metody idempotentnoy algebry v zadachakh modelirovaniya i analiza slozhnykh sistem [Methods of Idempotent Algebra in Problems of Complex Systems Modeling and Analysis]. St. Petersburg University Press, St. Petersburg, 2009. 255 p. (in Russian)
12. *Omelchenko A. V.* Teoriya grafov [Graph Theory]. Moscow, MCCME Publ., 2018. 416 p. (in Russian)
13. *Kochetov Y. A. and Stolyar A. A.* Novye zhadnye evristiki dlya zadachi kalendarnogo planirovaniya s ogranichennymi resursami [New greedy euristics for the the resource constrained project scheduling problem]. Diskretn. Analiz i Issled. Oper., 2005, vol. 12, no. 1, pp. 12–36. (in Russian)
14. `https://www.om-db.wi.tum.de/psplib/` — Project scheduling problem library, 2022.

СВЕДЕНИЯ ОБ АВТОРАХ

АБРОСИМОВ Михаил Борисович — доктор физико-математических наук, доцент, заведующий кафедрой теоретических основ компьютерной безопасности и криптографии Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского, г. Саратов. E-mail: **mic@rambler.ru**

БУЛАВЧУК Александр Михайлович — старший преподаватель Сибирского федерального университета, г. Красноярск. E-mail: **abulavchuk@sfu-kras.ru**

ДМУХ Андрей Александрович — кандидат физико-математических наук, ведущий научный сотрудник лаборатории математических проблем теоретической криптографии Академии криптографии Российской Федерации, г. Москва.
E-mail: **graphin@rambler.ru**

ИВАНОВ Борис Николаевич — доцент, кандидат физико-математических наук, доцент Дальневосточного федерального университета, Институт математики и компьютерных технологий, Департамент математики, г. Владивосток.
E-mail: **ivanov.bn@dvfu.ru**

КУНИНЕЦ Артем Андреевич — студент ОНК «Институт высоких технологий» БФУ им. И. Канта, г. Калининград. E-mail: **artkuninets@yandex.ru**

КУЩИНСКАЯ Людмила Александровна — Московский государственный университет им. М. В. Ломоносова, г. Москва.
E-mail: **lyudmila.kuschinskaja@yandex.ru**

ЛОБОВ Александр Андреевич — аспирант Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского, г. Саратов.
E-mail: **aisanekai@mail.ru**

МАЛЫГИНА Екатерина Сергеевна — кандидат физико-математических наук, доцент ОНК «Институт высоких технологий», младший научный сотрудник лаборатории «Математические методы защиты и обработки информации» научно-образовательного математического центра «Северо-Западный центр математических исследований имени Софьи Ковалевской» БФУ им. И. Канта, г. Калининград.
E-mail: **emalygina@kantiana.ru**

МЕЖЕННАЯ Наталья Михайловна — кандидат физико-математических наук, доцент, доцент кафедры прикладной математики Московского государственного технического университета им. Н. Э. Баумана, г. Москва.
E-mail: **natalia.mezhennaya@gmail.com**

МИХАЙЛОВ Владимир Гаврилович — доктор физико-математических наук, ведущий научный сотрудник Математического института им. В. А. Стеклова Российской академии наук, г. Москва. E-mail: **mikhail@mi-ras.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**

НОВОСЕЛОВ Семен Александрович — научный сотрудник БФУ им. И. Канта, г. Калининград. E-mail: **snovoselov@kantiana.ru**

ПАСЬКО Дмитрий Олегович — старший научный сотрудник лаборатории математических проблем теоретической криптографии Академии криптографии Российской Федерации, г. Москва. E-mail: **pasko.do@yandex.ru**

РЫБАЛОВ Александр Николаевич — кандидат физико-математических наук, старший научный сотрудник лаборатории комбинаторных и вычислительных методов алгебры и логики Института математики им. С. Л. Соболева СО РАН, г. Омск.
E-mail: **alexander.rybalov@gmail.com**

СЕМЕНОВА Дарья Владиславовна — кандидат физико-математических наук, доцент Сибирского федерального университета, г. Красноярск.
E-mail: **DVSemenova@sfu-kras.ru**

ШЕВЛЯКОВ Артём Николаевич — доктор физико-математических наук, старший научный сотрудник Института математики им. С. Л. Соболева СО РАН, профессор Омского государственного университета им. Ф. М. Достоевского, г. Омск.
E-mail: **art.shevlyakov@gmail.com**