МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2013 №2(20)

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

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Адрес редакции: 634050, г. Томск, пр. Ленина, 36
E-mail: vestnik_pdm@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 надежности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

Редактор *Н. И. Шидловская*
Верстка *И. А. Панкратовой*

Подписано к печати 11.06.2013.
Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 13,4. Уч.-изд. л. 15. Тираж 300 экз.

Издательство ТГУ. 634029, Томск, ул. Никитина, 4
Отпечатано в типографии ТПУ.

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/20/1

УДК 519.7

О НЕКОТОРЫХ СВОЙСТВАХ СЛАБО ПОЛОЖИТЕЛЬНЫХ И СЛАБО ОТРИЦАТЕЛЬНЫХ БУЛЕВЫХ ФУНКЦИЙ

С. П. Горшков

Институт криптографии, связи и информатики, г. Москва, Россия

E-mail: spg54@bk.ru

Исследуются некоторые свойства слабо положительных (антихорновских) и слабо отрицательных (хорновских) булевых функций. В частности, оценивается сложность задачи построения приведённых представлений этих функций, показывается, что нет ограничений на вес таких функций, оценивается возможная длина записи рассматриваемых функций.

Ключевые слова: *слабо положительная (антихорновская) булева функция, слабо отрицательная (хорновская) булева функция, вычислительная сложность.*

1. Некоторые известные свойства исследуемых функций

Определение 1. Булева функция $f(x_1, \ldots, x_n)$, $n \geqslant 1$, называется:

1) *слабо положительной (антихорновской)*, если $f \equiv 1$ или существует представление f в виде следующей КНФ:

$$f \equiv \bigwedge_{i=1}^{t} (x_{s_{i1}}^{\alpha_i} \vee x_{s_{i2}} \vee \ldots \vee x_{s_{ik_i}}),$$

где $\alpha_i \in \{0, 1\}$, $i = 1, \ldots, t$;

2) *слабо отрицательной (хорновской)*, если $f \equiv 1$ или существует представление f в виде следующей КНФ:

$$f \equiv \bigwedge_{i=1}^{t} (x_{s_{i1}}^{\alpha_i} \vee \bar{x}_{s_{i2}} \vee \ldots \vee \bar{x}_{s_{ik_i}}),$$

где $\alpha_i \in \{0, 1\}$, $i = 1, \ldots, t$.

Множество всех слабо положительных (слабо отрицательных) функций обозначим WP (WN). Указанные записи соответственно функций классов WP, WN называются *приведёнными представлениями*. Функции из классов WP, WN, зависящие от k переменных, обозначим WP_k, WN_k.

Актуальность задачи изучения указанных функций отмечена в работе [1].

Обозначим:

V_k — множество двоичных k-мерных векторов;

$V_k^{(i)} = \{\alpha \in V_k : \|\alpha\| = i\}$, где $i \in \{0, \ldots, k\}$ (слой векторов веса i);

$\lfloor c \rfloor$ — целая часть действительного числа c.

Пусть $q_1(n)$, $q_2(n)$ — действительные функции от натурального аргумента n; тогда если существует такое n_0, что для всех $n \geqslant n_0$ выполняется неравенство $q_1(n) > q_2(n)$, то будем записывать $q_1(n) \tilde{>} q_2(n)$.

Вектор $(\alpha_1, \ldots, \alpha_k) \in V_k$ назовём *выполняющим вектором* функции $f(x_1, \ldots, x_k)$, если $f(\alpha_1, \ldots, \alpha_k) = 1$. Множество всех выполняющих векторов функции f обозначим E_f.

Если $\alpha = (\alpha_1, \ldots, \alpha_k)$, $\beta = (\beta_1, \ldots, \beta_k)$ — двоичные векторы, $f(x_1, \ldots, x_k)$ — булева функция, то полагаем $f(\alpha) = f(\alpha_1, \ldots, \alpha_k)$, $\alpha \vee \beta = (\alpha_1 \vee \beta_1, \ldots, \alpha_k \vee \beta_k)$, $\alpha \wedge \beta = (\alpha_1 \wedge \beta_1, \ldots, \alpha_k \wedge \beta_k)$.

Замечание 1. Слабо положительные и слабо отрицательные функции близки между собой, поскольку $f(x_1, \ldots, x_k) \in WP$, если и только если $f(\bar{x}_1, \ldots, \bar{x}_k) \in WN$. Ввиду этой симметрии некоторые свойства будем формулировать только для слабо положительных функций.

Свойство 1 (критерий слабой положительности) [2]. Функция $f(x_1, \ldots, x_k) \not\equiv 0$ является слабо положительной, если и только если для любых двух выполняющих векторов $\{\alpha, \beta\} \subseteq E_f$ этой функции справедливо свойство

$$(\alpha \vee \beta) \in E_f.$$

Свойство 2 (критерий слабой отрицательности) [2]. Функция $f(x_1, \ldots, x_k) \not\equiv 0$ является слабо отрицательной, если и только если для любых двух выполняющих векторов $\{\alpha, \beta\} \subseteq E_f$ этой функции справедливо свойство

$$(\alpha \wedge \beta) \in E_f.$$

Замечание 2. В случае $|E_f| = 1$, $E_f = \{\alpha\}$, $\alpha = (\alpha_1, \ldots, \alpha_k) \in V_k$ для пары $\{\alpha, \alpha\} \subseteq E_f$ получим $(\alpha \vee \alpha) = \alpha \in E_f$ ($(\alpha \wedge \alpha) = \alpha \in E_f$) и, согласно критериям, $f \in WP$ и $f \in WN$, что справедливо, поскольку $f = x_1^{\alpha_1} \cdot \ldots \cdot x_k^{\alpha_k}$.

Замечание 3. Если $E_f = \varnothing$, то $f \in WP$ и $f \in WN$, поскольку $f = x_1 \cdot \bar{x}_1$.

Пусть $f(x_1, \ldots, x_k) \in WP$. На множестве E_f зададим отношение частичного порядка $\leqslant_f$ по правилу: если $\beta = (\beta_1, \ldots, \beta_k) \in E_f$, $\gamma = (\gamma_1, \ldots, \gamma_k) \in E_f$, то $\beta \leqslant_f \gamma$ тогда и только тогда, когда $\beta_i \leqslant \gamma_i$, $i = 1, \ldots, k$.

Свойство 3. Множество E_f слабо положительной функции f с отношением частичного порядка $\leqslant_f$ является верхней полурешёткой (определение верхней полурешётки см. в [3]).

Свойство 4 [4]. Для числа слабо положительных функций (а значит, и слабо отрицательных функций) справедлива оценка

$$\log_2 |WP_k| = \binom{k}{\lfloor k/2 \rfloor} (1 + \mathrm{O}(k^{-1/4} \log_2 k)). \tag{1}$$

Заметим, что для числа $m(k)$ монотонных функций от k переменных, которые составляют подкласс слабо положительных функций, верна оценка [5]

$$\log_2 m(k) = \binom{k}{\lfloor k/2 \rfloor} (1 + \mathrm{O}(2^{-k/2})).$$

Для задач распознавания свойств булевых функций

$$f \stackrel{?}{\in} WP, \quad f \stackrel{?}{\in} WN \tag{2}$$

справедливы следующие оценки сложности.

Свойство 5 [6]. Задачи (2) при задании булевых функций в виде КНФ или ДНФ являются NP-трудными. Эти задачи при задании булевых функций в виде СКНФ, СДНФ или многочлена Жегалкина являются полиномиальными.

Определение 2. Булева функция $f(x_1, \ldots, x_n), n \geqslant 1$, называется *биюнктивной*, если $f \equiv 1$ или существует представление f в виде 2-КНФ

$$f \equiv \left(\bigwedge_{i=1}^{t_1} x_{s_i}^{\alpha_i} \right) \wedge \left(\bigwedge_{i=1}^{t_2} (x_{r_{i1}}^{\beta_{i1}} \vee x_{r_{i2}}^{\beta_{i2}}) \right),$$

где $t_1 \geqslant 0$, $t_2 \geqslant 0$ (в случае $t_1 = 0$ или $t_2 = 0$ считаем, что соответствующий сомножитель отсутствует); $\alpha_i \in \{0, 1\}$, $i = 1, \ldots, t_1$; $\beta_{i1}, \beta_{i2} \in \{0, 1\}$, $i = 1, \ldots, t_2$.

Свойство 6 [7]. Справедливо соотношение

$$WP \cap WN = Bi \cap WP \cap WN,$$

где Bi — множество биюнктивных функций.

Свойство 7 [8]. Для любого целого $k \geqslant 2$ максимальной группой преобразований области определения функций от k переменных, относительно которых множество слабо положительных (слабо отрицательных) функций инвариантно, является группа всех перестановок переменных функций.

Свойство 8 [1]. Задача построения по СКНФ слабо положительной (слабо отрицательной) функции её сокращённой КНФ имеет полиномиальную сложность.

Свойство 9 [1]. Задача построения по многочлену Жегалкина монотонной (слабо положительной) функции её сокращённой КНФ не является полиномиальной.

2. Некоторые результаты о сложности нахождения приведённых представлений монотонных и антимонотонных булевых функций

В работе [1] приведены некоторые результаты о сложности нахождения приведённых представлений слабо положительных и слабо отрицательных булевых функций. В данной работе приведены результаты о сложности нахождения приведённого представления для подклассов слабо положительных и слабо отрицательных функций (а именно, монотонных и антимонотонных).

Определение 3. Булева функция f называется *антимонотонной*, если и только если $\bar{f}$ — монотонная функция.

В [1] не рассматриваются случаи исходного задания функций в виде КНФ или ДНФ. Нетрудно показать, что задачи распознавания «f — монотонная (антимонотонная) функция?» являются NP-трудными, однако представляет интерес задача нахождения приведённого представления указанных функций, когда заведомо известно, что данная КНФ или ДНФ есть монотонная (или антимонотонная) функция.

Функция $h(x_1, \ldots, x_k)$ называется *имплицентой* функции $f(x_1, \ldots, x_k)$, если

$$f(x_1, \ldots, x_k) \wedge h(x_1, \ldots, x_k) \equiv f(x_1, \ldots, x_k).$$

Имплиценты вида

$$x_{s_1}^{\alpha_1} \vee x_{s_2}^{\alpha_2} \vee \ldots \vee x_{s_r}^{\alpha_r},$$

где $s_i \neq s_j$ при $i \neq j$, называются *элементарными имплицентами*. Элементарная имплицента функции f называется *простой*, если никакая её собственная часть не является имплицентой функции f.

Любая функция $f(x_1,\ldots,x_k)$, $E_f \neq V_k$, представляется конъюнкцией всех своих простых имплицент. Конъюнктивная нормальная форма функции f, в которую входят все простые имплиценты функции f и только они, называется *сокращённой* КНФ функции f. Любая функция $f(x_1,\ldots,x_k)$ имеет единственную, с точностью до перестановки сомножителей, сокращённую КНФ.

Функция $g(x_1,\ldots,x_k)$ называется *импликантой* функции $f(x_1,\ldots,x_k)$, если

$$f(x_1,\ldots,x_k) \vee g(x_1,\ldots,x_k) \equiv f(x_1,\ldots,x_k).$$

Приведённым представлением монотонной функции называется представление f в виде следующей КНФ:

$$f \equiv \bigwedge_{i=1}^{t} (x_{s_{i_1}} \vee \ldots \vee x_{s_{ik_i}}).$$

Приведённым представлением антимонотонной функции называется представление f в виде следующей КНФ:

$$f \equiv \bigwedge_{i=1}^{t} (\bar{x}_{s_{i_1}} \vee \ldots \vee \bar{x}_{s_{ik_i}}).$$

Утверждение 1. Задача построения по КНФ монотонной функции минимальной (приведённой) КНФ имеет полиномиальную сложность.

Доказательство. Пусть монотонная функция $f(x_1,\ldots,x_k)$, не равная тождественно единице, записана в КНФ, которую можем представить в виде

$$f \equiv \bigwedge_{i\in I} (x_{s_{i1}} \vee \ldots \vee x_{s_{it_i}} \vee \bar{x}_{s_{it_i+1}} \vee \ldots \vee \bar{x}_{s_{ik_i}}). \tag{3}$$

Заметим, что если $t_i = 0$ для некоторого $i \in I$, то $f(1,\ldots,1) = 0$ и $f \equiv 0$. Случай $t_i < k_i$ для всех $i \in I$ невозможен, так как при этом $f(0,\ldots,0) = 1$, $f \equiv 1$, и функция f не может быть записана в виде (3). Далее предполагаем, что $t_i \geqslant 1$ для всех $i \in I$. По функции f построим функцию g:

$$g \equiv \bigwedge_{i\in I} (x_{s_{i1}} \vee \ldots \vee x_{s_{it_i}}). \tag{4}$$

Покажем, что

$$f \equiv g. \tag{5}$$

Пусть некоторый вектор $\alpha = (\alpha_1,\ldots,\alpha_k) \in V_k$ не является выполняющим вектором для функции f: $f(\alpha) = 0$. Тогда найдётся элементарная имплицента функции f (для простоты записи эту имплиценту обозначим $x_1 \vee \ldots \vee x_t \vee \bar{x}_{t+1} \vee \ldots \vee \bar{x}_q$), такая, что $\alpha_1 \vee \ldots \vee \alpha_t \vee \bar{\alpha}_{t+1} \vee \ldots \vee \bar{\alpha}_q = 0$. То есть $\alpha_1 = \ldots = \alpha_t = 0$, и вектор α не удовлетворяет имплиценте $x_1 \vee \ldots \vee x_t$ функции g и $g(\alpha) = 0$. Поэтому

$$f \geqslant g. \tag{6}$$

Предположим теперь, что $g(\beta) = 0$ для некоторого $\beta \in V_k$ и вектор β не удовлетворяет имплиценте $x_1 \vee \ldots \vee x_t$ функции g. Следовательно, $\beta_1 = \ldots = \beta_t = 0$. Рассмотрим вектор $\gamma = (\underbrace{0,\ldots,0}_{t},1,\ldots,1) \in V_k$. Ясно, что $g(\gamma) = f(\gamma) = 0$. Ввиду монотонности функции f получим $f(\beta) = 0$. Поэтому

$$f \leqslant g. \tag{7}$$

Из соотношений (6) и (7) следует справедливость (5).

Нетрудно показать, что для монотонных функций сокращённая КНФ совпадает с минимальной КНФ. Поэтому достаточно привести требуемый алгоритм построения сокращённой КНФ. Итак, пусть монотонная функция f задана конъюнктивной нормальной формой (4).

Докажем следующее свойство: если

$$x_1 \vee \ldots \vee x_t \tag{8}$$

— простая имплицента функции f, то эта имплицента содержится среди элементарных имплицент КНФ (4). Предположим противное: $x_1 \vee \ldots \vee x_t$ — имплицента функции f, но эта имплицента не содержится среди элементарных имплицент (4). Тогда, с одной стороны, поскольку $x_1 \vee \ldots \vee x_t$ — имплицента функции f, то

$$f(0, \ldots, 0, x_{t+1}, \ldots, x_k) \equiv 0. \tag{9}$$

С другой стороны, так как (8) — простая имплицента, то никакая её собственная часть не является имплицентой функции f и любая имплицента записи (4) содержит хотя бы одну переменную, отличную от $x_1, \ldots, x_t$. Следовательно, $f(0, \ldots, 0, 1, \ldots, 1) \equiv 1$, что противоречит условию (9). Из приведённого свойства вытекает, что все простые имплиценты содержатся в записи (4), достаточно их лишь выделить. Напомним, что необходимым и достаточным условием простоты имплиценты $x_1 \vee \ldots \vee x_t$ является выполнение условия (9) и условий

$$f(\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0, x_{t+1}, \ldots, x_k) \neq 0, i = 1, \ldots, t. \tag{10}$$

Всё это проверяется с полиномиальной сложностью. ■

Аналогично утверждению 1 можно доказать следующие результаты.

Утверждение 2. Задача построения по КНФ антимонотонной функции минимальной (приведённой) КНФ имеет полиномиальную сложность.

Утверждение 3. Задача построения по ДНФ монотонной (антимонотонной) функции минимальной ДНФ этой функции имеет полиномиальную сложность.

Утверждение 4. Задача построения по ДНФ монотонной функции её минимальной (приведённой) КНФ имеет экспоненциальную сложность.

Доказательство. Пусть $k \geqslant 2$. Рассмотрим монотонную функцию

$$f \equiv x_1x_2 \vee x_3x_4 \vee \ldots \vee x_{2\lfloor k/2 \rfloor - 1}x_{2\lfloor k/2 \rfloor}.$$

Легко показать, что сокращённая КНФ функции f имеет вид

$$f \equiv \bigwedge_{\substack{i_1 \in \{1,2\} \\ i_2 \in \{3,4\} \\ \ldots \\ i_{\lfloor k/2 \rfloor} \in \{2\lfloor k/2 \rfloor - 1, 2\lfloor k/2 \rfloor\}}} (x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_{\lfloor k/2 \rfloor}}). \tag{11}$$

Длина сокращённой КНФ (11) имеет порядок $2^{k/2}$, то есть в общем случае задача построения по ДНФ монотонной функции её сокращённой КНФ имеет экспоненциальную сложность. ■

Аналогично утверждению 4 может быть доказано

Утверждение 5. Задача построения по ДНФ антимонотонной функции её сокращённой КНФ имеет экспоненциальную сложность.

3. Некоторые другие свойства слабо положительных и слабо отрицательных булевых функций

Приведём некоторые другие ранее не известные свойства слабо положительных и слабо отрицательных функций. Эти свойства из разных областей исследования рассматриваемых булевых функций.

Теорема 1. Для любого $k \geqslant 1$ и любого $t \in \{1, \ldots, 2^k - 1\}$ найдётся слабо положительная функция веса t, существенно зависящая от всех k переменных.

Доказательство. Пусть $k \geqslant 1$ и $t \in \{1, \ldots, 2^k - 1\}$. Построим слабо положительную функцию $f(x_1, \ldots, x_k)$ с требуемыми свойствами. Полагаем $f(0, \ldots, 0) = 1$. Выберем наибольшее r, такое, что $\sum\limits_{i=k-r}^{k} \binom{k}{i} \leqslant t - 1$. Обозначим $q = \sum\limits_{i=k-r}^{k} \binom{k}{i}$. Для всех $i \geqslant k - r$ и $(\alpha_1, \ldots, \alpha_k) \in V_k^{(i)}$ полагаем $f(\alpha_1, \ldots, \alpha_k) = 1$.

Если $q = t - 1$, то на слоях с весом $1, \ldots, k - r - 1$ полагаем $f(\alpha_1, \ldots, \alpha_k) = 0$. В случае $q < t - 1$ на слое с весом $k - r - 1$ для $t - q - 1$ произвольно выбранных векторов полагаем $f(\alpha_1, \ldots, \alpha_k) = 1$. На остальных векторах этого слоя и на всех векторах слоёв с весом $1, \ldots, k - r - 2$ значение функции полагаем равным нулю.

В соответствии с приведённым выше критерием слабой положительности находим, что построенная функция является слабо положительной (но не монотонной, так как $f(0, \ldots, 0) = 1$, $\|f\| \neq 2^k$).

Если $t \leqslant 2^k - k$, то $f(\alpha) = 0$ для всех $\alpha \in V_k$, $\|\alpha\| = 1$, и поэтому функция f (помним, что $f(0, \ldots, 0) = 1$) существенно зависит от всех переменных. При $2^k - k < t \leqslant 2^k - 1$ найдётся вектор $\beta \in V_k^{(1)}$, такой, что $f(\beta) = 0$. Поскольку $f(\beta) = 1$ для всех $\beta \in V_k^{(2)}$ и $f(0, \ldots, 0) = 1$, то функция f существенно зависит от всех переменных. Теорема доказана. ■

Замечание 4. Утверждение теоремы не может быть распространено на значения $t = 0$ или $t = 2^k$, поскольку в этих случаях функция f от всех переменных зависит несущественно.

Замечание 5. Может быть построена монотонная функция с приведёнными в теореме 1 свойствами. Для доказательства этого факта необходимо несколько изменить функцию f, построенную выше, а именно положить $f(0, \ldots, 0) = 0$, а в самом верхнем слое, содержащем невыполняющие векторы функции f (пусть одним из таких векторов будет β), положить $f(\beta) = 1$.

Замечание 6. Поскольку функции-константы $f \equiv 0$ и $f \equiv 1$ являются и слабо положительными, и слабо отрицательными, то можно утверждать, что нет ограничений на вес слабо положительных и слабо отрицательных функций.

Замечание 7. Известно [7, 9], что любая мультиаффинная функция $f(x_1, \ldots, x_k)$ имеет вес 2^m, $m \in \{0, \ldots, k\}$, а равновероятных биюнктивных функций, существенно зависящих от $k \geqslant 5$ переменных, не существует [10]. Результат теоремы 1 показывает, что относительно веса функции классы слабо положительных и слабо отрицательных функций значительно богаче.

Аналогично теореме 1 может быть доказана

Теорема 2. Для любого $k \geqslant 1$ и любого $t \in \{1, \ldots, 2^k - 1\}$ найдётся слабо отрицательная функция веса t, существенно зависящая от всех k переменных.

Теорема 3. При любом виде записи слабо положительных (слабо отрицательных) функций в любом конечном алфавите A для любого полинома $p(k)$ можно ука-

зать k_0 (которое зависит от алфавита A и полинома $p(k)$), такое, что при $k \geqslant k_0$ найдутся функции из WP_k (WN_k), длина записи которых превышает $p(k)$.

Доказательство. Из формулы (1), представления биномиальных коэффициентов, оценок Стирлинга

$$\sqrt{2\pi k}\left(\frac{k}{e}\right)^k < k! < \sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{1/12k} \tag{12}$$

следует асимптотическое неравенство

$$|WP_k| \gtrsim 2^{2^{k-1}/\sqrt{k}}. \tag{13}$$

Для полинома $p(k)$ найдутся $c_1 \in N$, $c_2 \in N$, такие, что $p(k) \leqslant c_1 \cdot k^{c_2}$. Количество T слов в алфавите A длины не больше, чем $p(k)$, не более

$$T \leqslant |A|^{c_1 k^{c_2}}. \tag{14}$$

Поскольку $|A|$ — это некоторое натуральное число, то из соотношений (13) и (14) следует справедливость теоремы. ■

Замечание 8. Для любой мультиаффинной или биюнктивной функции $f(x_1, \ldots, x_k)$ существует короткая запись (не длиннее $p(k)$, где p — некоторый полином), такой короткой записью для этих функций является приведённое представление [7]. Из теоремы 3 следует, что для всех слабо положительных (и слабо отрицательных) функций короткой записи не существует.

В свете результата теоремы 3 представляет интерес рассмотрение слабо положительных или слабо отрицательных функций, которые коротко записываются в некотором базисе.

В работе [11] исследованы классы функций $Q_k = Bi_k \cap WP_k$, то есть функций, являющихся одновременно слабо положительными и биюнктивными. Поскольку длина записи приведённого представления любой биюнктивной функции не больше n^3, слабо положительные функции из класса Q_k имеют короткую запись. Аналогичным свойством обладают слабо отрицательные функции из класса $Bi_k \cap WN_k$. Отметим, что для числа функций класса Q_k справедливы оценки $2^{(k^2 - k\log_2 k)/4} < |Q_k| < 2^{(k^2/2)+2k\log_2 k+3k}$ [11]. В следующей теореме оценивается снизу число слабо отрицательных функций, для которых существует короткое представление многочленом Жегалкина.

Теорема 4. Существует не менее $\left\lfloor 2^{rk\log_2 k/2}\right\rfloor$ слабо отрицательных функций от k переменных, длина записи которых многочленом Жегалкина не больше чем n^r, где $r \geqslant 5$.

Доказательство. Будем рассматривать только антимонотонные функции вида

$$f(x_1, \ldots, x_k) = \bigwedge_{1 \leqslant i_1 < \ldots < i_t \leqslant k} (\lambda_{i_1,\ldots,i_t} \vee \bar{x}_{i_1} \vee \ldots \vee \bar{x}_{i_t}), \tag{15}$$

где $t \in \{1, \ldots, k\}$; $\overrightarrow{\lambda} = (\lambda_{12\ldots t}, \ldots, \lambda_{(k-t+1),\ldots,(k-1)k}) \in V_{\binom{k}{t}}$ — набор коэффициентов; $\|\overrightarrow{\lambda}\| = q$; q — некоторое число из интервала $\{1, \ldots, 2^{\binom{k}{t}}\}$, $2^{\binom{k}{t}} = \left|V_{\binom{k}{t}}\right|$ — мощность множества $V_{\binom{k}{t}}$. При каждом наборе $\overrightarrow{\lambda}$ имеем антимонотонную функцию, записанную в виде КНФ (считаем, что сомножители вида $(1 \vee \bar{x}_{i_1} \vee \ldots \vee \bar{x}_{i_t})$ удаляются).

Покажем (от противного), что функции вида (15) имеют единственное представление в виде (15). Пусть функция $f(x_1,\dots,x_k)$, кроме (15), представима в виде

$$f(x_1,\dots,x_k)=\bigwedge_{1\leqslant i_1<\ldots<i_t\leqslant k}(\mu_{i_1\dots i_t}\vee\bar{x}_{i_1}\vee\ldots\vee\bar{x}_{i_t}),\tag{16}$$

где $\overrightarrow{\mu}=(\mu_{12\dots t},\dots,\mu_{(k-t+1)\dots(k-1)k})\in V_{\binom{k}{t}}$, $\overrightarrow{\mu}\neq\overrightarrow{\lambda}$. Без ограничения общности можем считать, что $\lambda_{12\dots t}=1$, $\mu_{12\dots t}=0$. Тогда из (15) следует, что $f(\underbrace{1,\dots,1}_{t},0,\dots,0)=0$, а из записи (16) вытекает $f(\underbrace{1,\dots,1}_{t},0,\dots,0)=1$. Получили противоречие.

Число функций (15) равно

$$N=\binom{\binom{k}{t}}{q}.$$

Поскольку $\bar{x}_{i_1}\vee\ldots\vee\bar{x}_{i_t}=x_{i_1}\cdot\ldots\cdot x_{i_t}\oplus 1$, то длина записи функций (15) многочленом Жегалкина не больше

$$T\leqslant 2^q\cdot k^2,$$

так как число мономов в этом многочлене не больше 2^q, а каждый моном $x_{i_1}\cdot\ldots\cdot x_{i_u}$, $1\leqslant i_1<\ldots<i_u\leqslant k$, можно записать словом в алфавите $\{x,0,1,;\}$ длины не больше k^2 (при этом индексы при неизвестных записываем двоичными словами).

Далее положим $t=\lfloor k/2\rfloor$, $q=\lfloor(r-2)\log_2 k\rfloor$. При этом длина многочлена Жегалкина функции не более чем $T\leqslant k^r$, число функций (15) равно

$$N=\binom{\binom{k}{\lfloor k/2\rfloor}}{\lfloor(r-2)\log_2 k\rfloor}.$$

Учитывая соотношение (12), находим

$$N\tilde{\geqslant}\binom{\left\lfloor 2^k/(2\sqrt{k})\right\rfloor+1}{\lfloor(r-2)\log_2 k\rfloor}\tilde{\geqslant}\binom{\left\lfloor 2^k/(2\sqrt{k})\right\rfloor+1}{\lfloor(r-2)\log_2 k\rfloor}^{\lfloor(r-2)\log_2 k\rfloor}\tilde{\geqslant}2^{rk\log_2 k/2}.$$

Теорема доказана. ■

Замечание 9. При $k\geqslant 6$ функции (15) не являются биюнктивными, поскольку в этих условиях элементарные имплиценты, входящие в (15), простые и зависят более чем от трёх переменных.

ЛИТЕРАТУРА

1. *Горшков С. П.* О сложности нахождения приведённых представлений слабо положительных и слабо отрицательных булевых функций // Прикладная дискретная математика. 2008. Т. 1. Вып. 1. С. 7–9.
2. *Гизунов С. А., Носов В. А.* О классификации всех булевых функций четырёх переменных по классам Шеффера // Обозрение прикладной промышленной математики. Сер. дискретной математики. 1995. Т. 2. Вып. 3. С. 440–467.
3. *Гретцер Г.* Общая теория решёток. М.: Мир, 1982. 456 с.
4. *Алексеев В. Б.* О числе семейств подмножеств, замкнутых относительно пересечения // Дискретная математика. 1989. Т. 1. Вып. 2. С. 129–136.
5. *Коршунов А. Д.* О числе монотонных булевых функций. Проблемы кибернетики. Вып. 38. М.: Наука, 1981. С. 5–108.

6. *Горшков С. П.* О сложности распознавания мультиаффинности, биюнктивности, слабой положительности и слабой отрицательности булевых функций // Обозрение прикладной промышленной математики. Сер. дискретной математики. 1997. Т. 4. Вып. 2. С. 440–467.
7. *Горшков С. П.* Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обозрение прикладной промышленной математики. Сер. дискретной математики. 1995. Т. 2. Вып. 3. С. 325–398.
8. *Горшков С. П., Тарасов А. В.* О максимальных группах инвариантных преобразований мультиаффинных, биюнктивных, слабо положительных и слабо отрицательных булевых функций // Дискретная математика. 2009. Т. 21. Вып. 2. С. 94–101.
9. *Schaefer T.* Complexity of satisfiability problems // Proc. 10 Annual ACM Symposium on Theory of Computing Machinery. New York, NY, USA: ACM, 1978. P. 216–226.
10. *Тарасов А. В.* О свойствах функций, представимых в виде 2-КНФ // Дискретная математика. 2001. Т. 13. Вып. 4. С. 99–115.
11. *Горшков С. П.* О пересечении классов мультиаффинных, биюнктивных, слабо положительных и слабо отрицательных булевых функций // Обозрение прикладной промышленной математики. Сер. дискретной математики. 1997. Т. 4. Вып. 2. С. 238–259.

2013 Теоретические основы прикладной дискретной математики №2(20)

DOI 10.17223/20710410/20/2

УДК 510.52

НИЖНЯЯ И ВЕРХНЯЯ ОЦЕНКИ ПОРЯДКА АФФИННОСТИ ПРЕОБРАЗОВАНИЙ ПРОСТРАНСТВ БУЛЕВЫХ ВЕКТОРОВ

С. П. Горшков*, А. В. Двинянинов**

*Институт криптографии, связи и информатики, г. Москва, Россия

**Лаборатория ТВП, г. Москва, Россия

E-mail: spg54@bk.ru

Находятся нижняя и верхняя оценки порядка аффинности множества всех преобразований n-мерного пространства булевых векторов. Результаты работы могут быть использованы при оценке сложности одного метода решения систем булевых уравнений.

Ключевые слова: *преобразование пространства булевых векторов, аффинное отображение, сложность решения систем булевых уравнений.*

Будем использовать следующие обозначения:
$\mathbb{N}$ — множество натуральных чисел;
V_n — n-мерное пространство булевых векторов, $n \in \mathbb{N}$;
Φ_n — множество всех отображений $V_n \to V_n$;
$0_n^{\downarrow} = (\underbrace{0, \ldots, 0}_{n})^T$;
$\lfloor c \rfloor$ — целая часть числа c,
если действительные функции $g(n), h(n)$ определены для всех натуральных $n \in \mathbb{N}$ и существует такое $n_0 \in \mathbb{N}$, что для всех $n \geqslant n_0$ выполняется $g(n) < h(n)$, то будем записывать

$$g(n) \tilde{<} h(n).$$

Всякое отображение $F \in \Phi_n$ записывается системой координатных булевых функций $F = \{(f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)\}$.

Отображение F называется *линейным* (*аффинным*), если все его координатные функции линейны (аффинны).

Введём ещё некоторые определения, которые заимствованы из [1, с. 163–164].

Отображение $F \in \Phi_n$ *аффинно* на множестве M, где $M \subseteq V_n$, если найдётся аффинное отображение $A \in \Phi_n$, такое, что $F(\alpha) = A(\alpha)$ при любом $\alpha \in M$. Тогда отображение A назовём *линеаризующим F на множестве M*.

Пусть $M_1, \ldots, M_r$ — разбиение множества V_n, $A_1, \ldots, A_r$ — отображения, линеаризующие F на множествах $M_1, \ldots, M_r$ соответственно. При этом r назовём *порядком разбиения* множества V_n, линеаризующего F. *Порядком аффинности отображения F* (обозначим его $\operatorname{ard} F$) назовём наименьший из порядков разбиений множества V_n, линеаризующих F.

Прикладное значение введённых определений состоит в следующем. Предположим, что необходимо решить систему булевых уравнений вида

$$\begin{cases} f_1(x_1, \ldots, x_n) = \gamma_1, \\ \ldots \\ f_n(x_1, \ldots, x_n) = \gamma_n, \end{cases} \tag{1}$$

где $\{f_1, \ldots, f_n\}$ — координатные функции отображения F. Систему (1) будем записывать также в виде

$$F(x) = \gamma, \tag{2}$$

где $x = (x_1, \ldots, x_n)$, $\gamma = \begin{pmatrix} \gamma_1 \\ \ldots \\ \gamma_n \end{pmatrix}$.

Пусть $A_1, \ldots, A_{\operatorname{ard} F}$ — отображения, линеаризующие F. Тогда для решения (1) (или (2)) достаточно решить следующее множество линейных булевых систем уравнений:

$$A_i(x) = \gamma, \quad i = 1, \ldots, \operatorname{ard} F,$$

и полученные решения проверить по исходной системе (2). Ясно, что общее число решаемых линейных систем уравнений равно $\operatorname{ard} F$.

Дальнейшие определения введены авторами данной работы.

Пусть

$$\operatorname{ard} \Phi_n = \max_{F \in \Phi_n} \operatorname{ard} F.$$

Величину $\operatorname{ard} \Phi_n$ назовём *порядком аффинности* Φ_n. Параметр $\operatorname{ard} \Phi_n$ оценивает сверху число решаемых линейных систем при любой системе (2).

Теорема 1. При $n \geqslant 2$ для порядка аффинности справедлива нижняя оценка

$$\operatorname{ard} \Phi_n > \frac{2^n}{n^2}.$$

Доказательство. Нетрудно показать, что число аффинных отображений $V_n \to V_n$ равно $2^{n(n+1)}$. Пусть

$$A_1, \ldots, A_{2^{n(n+1)}} \tag{3}$$

— все аффинные отображения из Φ_n. Обозначим символом u порядок аффинности $\operatorname{ard} \Phi_n$. Тогда для каждого $F \in \Phi_n$ найдутся разбиение V_n на блоки $M_1, \ldots, M_k$, $k \leqslant u$, и аффинные отображения $A_{i_1}, \ldots, A_{i_k}$, такие, что A_{i_j} линеаризует F на M_j, $j \in \{1, \ldots, k\}$. Если считать, что некоторые блоки M_l могут быть пустыми, то можно полагать, что все линеаризующие наборы $A_{i_1}, \ldots, A_{i_k}$ состоят из u элементов.

Оценим сверху число h всех отображений из Φ_n, которые могут быть линеаризованы наборами из не более чем u аффинных отображений:

$$h \leqslant \binom{2^{n(n+1)}}{u} \cdot u^{2^n}. \tag{4}$$

Первый сомножитель в (4) — число вариантов, которыми можем выбрать $A_{i_1}, \ldots, A_{i_u}$ из всех аффинных отображений (3). Если набор $A_{i_1}, \ldots, A_{i_u}$ задан, то он линеаризует не более чем u^{2^n} отображений $F \in \Phi_n$, поскольку каждый элемент $F(\alpha)$, $\alpha \in V_n$, может принимать не более u значений.

Нетрудно видеть, что

$$h < 2^{un(n+1)} \cdot u^{2^n}. \tag{5}$$

Логарифмируя обе части неравенства (5) по основанию 2, получим

$$\log_2 h < un(n+1) + 2^n \cdot \log_2 u. \tag{6}$$

Предположим, что выполнено неравенство

$$\operatorname{ard}\Phi_n \leqslant \frac{2^n}{n^2}.$$

Тогда из (6) следует, что

$$\log_2 h < 2^n\left(n + \frac{n(n+1)}{n^2} - 2\log_2 n\right). \tag{7}$$

При $n \geqslant 2$ из неравенства (7) следует

$$\log_2 h < n \cdot 2^n. \tag{8}$$

Поскольку u — порядок аффинности Φ_n, то $h = |\Phi_n| = 2^{n\cdot 2^n}$,

$$\log_2 h = n \cdot 2^n. \tag{9}$$

Из полученного противоречия (8) и (9) вытекает, что $\operatorname{ard}\Phi_n > \dfrac{2^n}{n^2}$. ■

Найдём верхнюю оценку для $\operatorname{ard}\Phi_n$. Получение этой оценки предварим несколькими достаточно простыми утверждениями.

Легко показать, что справедлива следующая лемма.

Лемма 1. Пусть $\{(\beta^{(1)}, \delta^{(1)}), \ldots, (\beta^{(m)}, \delta^{(m)})\}$ — пары векторов из V_n, причём $m \leqslant n$, а векторы $\{\beta^{(1)}, \ldots, \beta^{(m)}\}$ линейно независимы. Тогда найдётся линейное отображение $L \in \Phi_n$, такое, что

$$L(\beta^{(i)}) = \delta^{(i)}, \quad i = 1, \ldots, m.$$

Лемма 2. Пусть $M \subseteq V_n$, $|M| \geqslant 2^k$, где $k \in \{0, \ldots, n-1\}$, все векторы в M ненулевые. Тогда в M найдётся $k+1$ линейно независимых векторов.

Доказательство. Предположим противное, что максимальная (точнее, одна из максимальных) система линейно независимых векторов состоит из $t < k+1$ векторов. Пусть это векторы $\varphi^{(1)}, \ldots, \varphi^{(t)}$. Символом M' обозначим линейное пространство векторов с базисом $\varphi^{(1)}, \ldots, \varphi^{(t)}$, $M'' = M' \setminus \{0_n^{\downarrow}\}$. Ясно, что $M'' \supseteq M$, то есть $|M''| \geqslant 2^k$. С другой стороны, $|M''| = 2^t - 1 \leqslant 2^k - 1 < 2^k$. Из полученного противоречия вытекает справедливость леммы. ■

Теорема 2. При $n \geqslant 2$ для порядка аффинности справедлива следующая верхняя оценка:

$$\operatorname{ard}\Phi_n \leqslant \frac{2^n}{n} \sum_{i=1}^{n-1} 2^{-i} \cdot \frac{n}{n-i}.$$

Доказательство. Пусть F — произвольное, но фиксированное отображение из Φ_n. Линеаризовывать отображение F будем следующим образом: все ненулевые векторы V_n разобьём на непересекающиеся группы линейно независимых векторов (число этих групп обозначим r). Выделим одну группу (пусть это будут векторы $\{\alpha_1^{(1)}, \ldots, \alpha_n^{(1)}\}$) и линеаризуем значение F на этих векторах некоторым аффинным преобразованием так, чтобы одновременно линеаризовалось значение $F(0_n^{\downarrow})$.

Согласно доказательству леммы 1, можем подобрать такое линейное отображение $L \in \Phi_n$, что $L(\alpha_i^{(1)}) = F(\alpha_i^{(1)}) \oplus F(0_n^{\downarrow})$, $i = 1, \ldots, n$. Тогда аффинное отображение

$L(x^{\downarrow}) \oplus F(0_n^{\downarrow})$ линеаризует F на векторах $\{\alpha_1^{(1)}, \ldots, \alpha_n^{(1)}, 0_n^{\downarrow}\}$. На каждой из остальных $r-1$ групп векторов, согласно лемме 1, отображение F можем линеаризовать соответствующим линейным отображением. Из этих рассуждений вытекает, что $\operatorname{ard} \Phi_n \leqslant r$.

Оценим значение r. Рассмотрим множество ненулевых векторов $V_n \setminus \{0_n^{\downarrow}\}$, при этом если $n \geqslant 1$, то $\left|V_n \setminus \{0_n^{\downarrow}\}\right| = 2^n - 1 \geqslant 2^{n-1}$. Согласно лемме 2, из множества $V_n \setminus \{0_n^{\downarrow}\}$ можем выбрать n линейно независимых векторов $\alpha_1^{(1)}, \ldots, \alpha_n^{(1)}$. Если для оставшегося множества $M' = V_n \setminus \{\alpha_1^{(1)}, \ldots, \alpha_n^{(1)}, 0_n^{\downarrow}\}$ будет выполняться $|M'| \geqslant 2^{n-1}$, то выберем ещё n линейно независимых векторов $\alpha_1^{(2)}, \ldots, \alpha_n^{(2)}$, и так далее, пока для оставшегося множества $M'' = V_n \setminus (\{\alpha_1^{(1)}, \ldots, \alpha_n^{(1)}, 0_n^{\downarrow}\} \cup \{\alpha_1^{(2)}, \ldots, \alpha_n^{(2)}\} \cup \ldots \cup \{\alpha_1^{(r_{n-1})}, \ldots, \alpha_n^{(r_{n-1})}\})$ не будет впервые выполняться $|M''| < 2^{n-1}$. Таким образом сформируем r_{n-1} групп линейно независимых векторов из n векторов, причём r_{n-1} такое, что

$$r_{n-1} = \frac{2^n - 1 - (2^{n-1} - q_{n-1})}{n} = \frac{2^{n-1} - 1 + q_{n-1}}{n},$$

где $q_{n-1} = n$, если $(2^{n-1} - 1)/n$ — целое; и $q_{n-1} \in \{1, \ldots, n-1\}$ таково, что $(2^{n-1} - 1 + {}+q_{n-1})/n$ целое, если $(2^{n-1} - 1)/n$ — дробное.

Далее формируем группы из $n-1$ линейно независимых векторов. Таких групп сформируем

$$r_{n-2} = \frac{2^{n-1} - q_{n-1} - (2^{n-2} - q_{n-2})}{n-1} = \frac{2^{n-2} - q_{n-1} + q_{n-2}}{n-1},$$

где $q_{n-2} \in \{1, \ldots, n-1\}$, и так далее.

Нетрудно видеть, что можно сформировать

$$r_i = \frac{2^{i+1} - q_{i+1} - (2^i - q_i)}{i+1} = \frac{2^i - q_{i+1} + q_i}{i+1}$$

групп из $i+1$ линейно независимых векторов, где $q_i \in \{1, \ldots, i+1\}$, $i \in 0, \ldots, n-2$.

В целом получаем

$$\begin{aligned}
r = r_{n-1} + r_{n-2} + \ldots + r_0 &= \frac{2^{n-1} - 1 + q_{n-1}}{n} + \frac{2^{n-2} - q_{n-1} + q_{n-2}}{n-1} + r_{n-3} + \ldots + r_0 = \\
&= \frac{1}{n-1}\left(\frac{n-1}{n}(2^{n-1} - 1 + q_{n-1}) + 2^{n-2} - q_{n-1} + q_{n-2}\right) + r_{n-3} + \ldots + r_0 = \\
&= \frac{1}{n-1}\left(2^{n-1} - \frac{2^{n-1}}{n} + 2^{n-2} + q_{n-2} - \left(\frac{q_{n-1}}{n} + \frac{n-1}{n}\right)\right) + r_{n-3} + \ldots + r_0 \leqslant \\
&\leqslant \frac{2^{n-1}}{n-1} + \frac{2^{n-2} - 1 + q_{n-2}}{n-1} + r_{n-3} + r_{n-4} + \ldots + r_0 = \\
&= \frac{2^{n-1}}{n-1} + \left(\frac{2^{n-2} - 1 + q_{n-2}}{n-1} + \frac{2^{n-3} - q_{n-2} + q_{n-3}}{n-2}\right) + r_{n-4} + \ldots + r_0 \leqslant \\
&\leqslant \frac{2^{n-1}}{n-1} + \frac{2^{n-2}}{n-2} + \left(\frac{2^{n-3} - 1 + q_{n-3}}{n-2} + \frac{2^{n-4} - q_{n-3} + q_{n-4}}{n-3}\right) + r_{n-5} + \ldots + r_0 \leqslant \\
&\leqslant \frac{2^{n-1}}{n-1} + \frac{2^{n-2}}{n-2} + \ldots + \frac{2^2}{2} + \left(\frac{2^1 - 1 + q_1}{2} + \frac{2^0 - q_1 + q_0}{1}\right) = \\
&= \frac{2^{n-1}}{n-1} + \frac{2^{n-2}}{n-2} + \ldots + \frac{2^2}{2} + \frac{2^1}{1} = \frac{2^n}{n}\left(2^{-1}\frac{n}{n-1} + 2^{-2}\frac{n}{n-2} + \ldots + 2^{-(n-1)}\frac{n}{1}\right).
\end{aligned}$$

Теорема доказана. ■

Лемма 3. Справедливо соотношение

$$\lim_{n\to\infty}\sum_{i=1}^{n-1}2^{-i}\frac{n}{n-i}=1.$$

Доказательство. Оцениваемую сумму разобьём на две подсуммы:

$$\sum_{i=1}^{n-1}2^{-i}\frac{n}{n-i}=\sum_{i=1}^{\lfloor 3\log_2 n\rfloor}2^{-i}\frac{n}{n-i}+\sum_{i=\lfloor 3\log_2 n\rfloor+1}^{n-1}2^{-i}\frac{n}{n-i}. \tag{10}$$

Первую сумму в левой части (10) обозначим S_1, вторую — S_2. Нетрудно видеть, что

$$\sum_{i=1}^{\lfloor 3\log_2 n\rfloor}2^{-i}<S_1<\frac{n}{n-\lfloor 3\log_2 n\rfloor}\sum_{i=1}^{\lfloor 3\log_2 n\rfloor}2^{-i}. \tag{11}$$

Верхняя и нижняя оценки в (11) при $n\to\infty$ стремятся к 1.

Для S_2 верны оценки $0<S_2<n\cdot 2^{-\lfloor 3\log_2 n\rfloor}\leqslant n^{-1}$, то есть $\lim\limits_{n\to\infty}S_2=0$. Отсюда и из (11) следует справедливость леммы. ■

Следствие 1. Для порядка аффинности Φ_n выполняется следующее асимптотическое неравенство:

$$\operatorname{ard}\Phi_n\tilde{<}1{,}01\cdot\frac{2^n}{n}.$$

Справедливость следствия 1 непосредственно вытекает из теоремы 2 и леммы 3.

ЛИТЕРАТУРА

1. *Фомичёв В. М.* Дискретная математика и криптология. М.: Диалог-МИФИ, 2003. 397 с.

2013 Теоретические основы прикладной дискретной математики №2(20)

DOI 10.17223/20710410/20/3

УДК 519.7

О СВЯЗЯХ МЕЖДУ НЕКОТОРЫМИ ПАРАМЕТРАМИ СОВЕРШЕННО УРАВНОВЕШЕННЫХ БУЛЕВЫХ ФУНКЦИЙ[1]

С. В. Смышляев

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

E-mail: smyshsv@gmail.com

Для класса совершенно уравновешенных булевых функций с барьером доказываются соотношения, связывающие между собой определённые параметры таких функций. В частности, получены общие результаты о свойствах полиномов функций с барьером, позволяющие установить новые оценки числа инверсионных функций для произвольной функции с барьером.

Ключевые слова: *совершенно уравновешенные функции, функции с барьером, криптография.*

Введение

Для обеспечения у используемых в криптографических целях кодирующих устройств, построенных на основе регистра сдвига и булевой функции усложнения [1 – 3], свойства сохранения равномерного распределения преобразуемой двоичной последовательности функции усложнения выбираются из множества совершенно уравновешенных булевых функций [2, 4, 5]. Среди таких функций особое место занимают функции с барьером [2, 6], обладающие также положительным свойством отсутствия предсказывания [2, 7]. Для функций с барьером в работе [7] введены целочисленные параметры, определённым образом характеризующие отображения, реализуемые кодирующими устройствами с такими функциями при фиксированном начале входной последовательности.

Настоящая работа посвящена вопросам о соотношениях между параметрами функций с барьером. В частности, доказано утверждение об определённых свойствах полиномиальных представлений таких функций — вопрос, результаты по которому ранее отсутствовали, за исключением частных и тривиальных.

Полученные результаты позволяют расширить область применимости утверждения о числе инверсионных функций, полученного в работе [8]. Данное утверждение представляет собой решение открытого вопроса, поставленного в работе Х. Лэя и Дж. Месси [9], и содержит выражение, позволяющее вычислить число инверсионных функций по трём параметрам функции: числу переменных, длине барьера и параметру e_f^R, описанному в [7]. С использованием полученных результатов возможно эффективное (с полиномиальной относительно длины входа сложностью) нахождение по полиномиальному представлению булевой функции нетривиальных оценок на e_f^R и, следовательно, на число инверсионных функций.

1. Определения и предварительные результаты

Будем использовать следующие обозначения. Множество двоичных наборов длины n будем обозначать через $V_n = \{0,1\}^n$, множество булевых функций n переменных — через $\mathcal{F}_n$.

[1]Работа поддержана РФФИ (проект № 12-01-00680-а).

Для всяких $n, l \in \mathbb{N}$, $f \in \mathcal{F}_n$ будем через f_l обозначать следующее отображение из V_{l+n-1} в V_l:

$$f_l(x_1, x_2, \ldots, x_{l+n-1}) \equiv (f(x_1, \ldots, x_n), f(x_2, \ldots, x_{n+1}), \ldots, f(x_l, \ldots, x_{l+n-1})).$$

Отображение f_l описывает преобразование, производимое l тактами работы кодирующего устройства, которое построено подсоединением входов реализующей булеву функцию f схемы к ячейкам двоичного регистра сдвига [1, 4].

Для всяких $n, l, p \in \mathbb{N}$, $p \leqslant l + n - 1$, $f \in \mathcal{F}_n$ и любого набора $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p) \in V_p$ будем через $f_{R,l,p}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_p)}$ обозначать следующее отображение из $V_{l+n-1-p}$ в V_l:

$$f_{R,l,p}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_p)}(x_1, x_2, \ldots, x_{l+n-1-p}) \equiv f_l(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p, x_1, x_2, \ldots, x_{l+n-1-p}),$$

и через $f_{L,l,p}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_p)}$ — следующее отображение из $V_{l+n-1-p}$ в V_l:

$$f_{L,l,p}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_p)}(x_1, x_2, \ldots, x_{l+n-1-p}) \equiv f_l(x_1, x_2, \ldots, x_{l+n-1-p}, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p).$$

Отображение $f_{R,l,p}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_p)}$ описывает преобразование, производимое l тактами работы кодирующего устройства с функцией усложнения f на двоичных последовательностях с фиксированным началом $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)$.

Приведём понятия совершенно уравновешенной булевой функции, а также барьера булевой функции.

Определение 1 [6]. Булева функция $f \in \mathcal{F}_n$ называется совершенно уравновешенной, если соотношение

$$\left|f_m^{-1}(y)\right| = 2^{n-1}$$

выполняется для любого $m \in \mathbb{N}$ и любого $y \in V_m$.

Множество совершенно уравновешенных функций из $\mathcal{F}_n$ будем обозначать $\mathcal{PB}_n$.

Определение 2 [6]. Булева функция $f \in \mathcal{F}_n$ называется функцией с правым барьером длины b, $b \in \mathbb{N}$, если система уравнений

$$\begin{cases} f_{b'}(x_1, x_2, \ldots, x_{b'+n-1}) = f_{b'}(z_1, z_2, \ldots, z_{b'+n-1}), \\ x_1 = z_1, \ldots, x_{n-1} = z_{n-1}, x_n = 0, z_n = 1 \end{cases}$$

имеет решение при всяком $b' \in \mathbb{N}$, таком, что $b' \leqslant b - 1$, а система уравнений

$$\begin{cases} f_b(x_1, x_2, \ldots, x_{b+n-1}) = f_b(z_1, z_2, \ldots, z_{b+n-1}), \\ x_1 = z_1, \ldots, x_{n-1} = z_{n-1}, x_n = 0, z_n = 1 \end{cases}$$

решений не имеет.

Булева функция $f \in \mathcal{F}_n$ называется функцией с левым барьером длины b, если $\overleftrightarrow{f}(x_1, x_2, \ldots, x_n) \equiv f(x_n, x_{n-1}, \ldots, x_1)$ является функцией с правым барьером длины b.

Булева функция $f \in \mathcal{F}_n$ имеет барьер, если она имеет правый или левый барьер, или оба сразу. При этом длиной барьера функции называется соответственно длина правого барьера, левого барьера или меньшая из длин барьеров.

Длины правого и левого барьера булевой функции f будем обозначать через b_f^R и b_f^L соответственно.

Наличие правого (левого) барьера длины 1, как нетрудно заметить, означает линейность функции по последнему (первому) аргументу.

Очевидным является тот факт, что для фиксированного c проверка неравенства $b_f^R \leqslant c$ по полиному функции, содержащему N мономов, требует порядка $(2N+1)^c$ операций. Для этого требуется лишь перемножить (пользуясь только полиномиальным представлением функций) выражения вида $f(x_i, x_{i+1}, \ldots, x_{i+n-1}) \oplus \oplus f(z_i, z_{i+1}, \ldots, z_{i+n-1}) \oplus 1$ для $i = 1, 2, \ldots, c$, затем в полученное полиномиальное представление подставить $x_1 = z_1, \ldots, x_{n-1} = z_{n-1}$, $x_n = 0$, $z_n = 1$, привести подобные и сравнить полученное выражение с нулём — при равенстве, очевидно, функция имеет правый барьер длины не более c.

В работе [7] доказан ряд утверждений о свойствах прообразов выходных наборов относительно отображений $f_{R,l,p}$ и $f_{L,l,p}$ в случае наличия у функции f правого или левого барьера.

Теорема 1 [7]. Для каждой функции $f \in \mathcal{F}_n$ с правым барьером можно определить величину $e_f^R \in \{0, 1, 2, \ldots, b_f^R - 1\}$, $e_f^R \leqslant n-1$, такую, что для любых $p \geqslant n-1$, $l \geqslant b_f^R + p - n$, $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p) \in V_p$ и любого набора $(y_1, y_2, \ldots, y_l) \in \mathrm{Im}\left(f_{R,l,p}^{(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)}\right)$ выполняется равенство

$$\left| f_{R,l,p}^{(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^{-1}}(y_1, y_2, \ldots, y_l) \right| = 2^{e_f^R}.$$

Теорема 2 [7]. Для каждой функции $f \in \mathcal{F}_n$ с левым барьером можно определить величину $e_f^L \in \{0, 1, 2, \ldots, b_f^L - 1\}$, $e_f^L \leqslant n-1$, такую, что для любых $p \geqslant n-1$, $l \geqslant b_f^L + p - n$, $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p) \in V_p$ и любого набора $(y_1, y_2, \ldots, y_l) \in \mathrm{Im}\left(f_{L,l,p}^{(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)}\right)$ выполняется равенство

$$\left| f_{L,l,p}^{(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^{-1}}(y_1, y_2, \ldots, y_l) \right| = 2^{e_f^L}.$$

Следствие 1 [7]. Для любой функции $f \in \mathcal{F}_n$, имеющей правый (левый) барьер, любых $l \geqslant b_f^R - 1$ ($l \geqslant b_f^L - 1$) и $x \in V_{n-1}$ верно равенство $\left|\mathrm{Im}(f_{R,l,n-1}^x)\right| = 2^{l - e_f^R}$ ($\left|\mathrm{Im}(f_{L,l,n-1}^x)\right| = 2^{l-e_f^L}$).

Лемма 1 [7]. Если у некоторой $f \in \mathcal{F}_n$ есть правый (левый) барьер, то $e_f^R = 0$ (соответственно, $e_f^L = 0$) тогда и только тогда, когда $b_f^R = 1$ ($b_f^L = 1$).

Лемма 2 [7]. Пусть $f(x_1, x_2, \ldots, x_n) \in \mathcal{F}_n$ имеет правый (левый) барьер. Тогда $e_f^R = b_f^R - 1$ (соответственно, $e_f^L = b_f^L - 1$) тогда и только тогда, когда функция f не зависит существенно от переменных $x_{n-b_f^R+2}, x_{n-b_f^R+3}, \ldots, x_n$ и линейна по переменной $x_{n-b_f^R+1}$ (не зависит существенно от переменных $x_1, x_2, \ldots, x_{b_f^L-1}$ и линейна по переменной $x_{b_f^L}$).

В работе [9] введено понятие инверсионной функции, которое в терминах функций с правым барьером может быть определено следующим образом.

Определение 3. Пусть $f \in \mathcal{F}_n$ имеет правый барьер. Тогда булева функция $\hat{f} \in \mathcal{F}_{b_f^R+n-1}$ является инверсионной для функции f, если для всякого набора $(x_1, x_2, \ldots, x_{b_f^R+n-1}) \in V_{b_f^R+n-1}$ выполняется равенство

$$\hat{f}(x_1, x_2, \ldots, x_{n-1}, f(x_1, x_2, \ldots, x_n), \ldots, f(x_{b_f^R}, x_{b_f^R+1}, \ldots, x_{b_f^R+n-1})) = x_n.$$

Исследования понятия инверсионной функции проводились в работах [10, 11]. Тем не менее оставался открытым поставленный в конце работы [9] вопрос о числе инверсионных функций для фиксированной функции с правым барьером. Из доказанных в работах [7, 8] свойств функций без предсказывания вытекает следующая теорема.

Теорема 3 [8]. Если $f \in \mathcal{F}_n$ имеет правый барьер, то для f существует в точности $2^{2^{b_f^R+n-1}\left(1-2^{-e_f^R}\right)}$ инверсионных функций.

Несмотря на полученное утверждение о числе инверсионных функций, при вычислении данной величины в случае представления функции полиномом существует серьёзная проблема. В отличие от параметров n и b_f^R, для нахождения значения параметра e_f^R (или для проверки неравенства $e_f^R \leqslant c$ для фиксированного c) по полиномиальному представлению нет известных алгоритмов полиномиальной относительно длины входа сложности. То есть нахождение числа инверсионных функций для булевой функции от n переменных, представленной полиномом длины $\mathrm{O}(n)$, требует порядка 2^n операций. Таким образом, актуальна задача нахождения соотношений между параметром e_f^R и легко вычислимыми по полиному параметрами функций, которые позволят строить эффективные алгоритмы для получения нетривиальных оценок e_f^R (а значит, и нетривиальных оценок числа инверсионных функций) по полиному функции f.

При дальнейших построениях будет использован известный критерий равенства алгебраической степени булевой функции числу её переменных.

Лемма 3 [2]. Пусть $f \in \mathcal{F}_n$. Функция f имеет алгебраическую степень n тогда и только тогда, когда её вес нечётен.

2. Основные результаты

Обозначим для произвольной функции $f \in \mathcal{F}_n$ через t_f^R наибольшее целое t', такое, что в полиноме функции f присутствует моном, который содержит все переменные $x_n, x_{n-1}, \ldots, x_{n-t'+1}$, через t_f^L — наибольшее целое t'', такое, что в полиноме функции f присутствует моном, который содержит все переменные $x_1, x_2, \ldots, x_{t''}$.

Теорема 4. Пусть функция $f \in \mathcal{F}_n$ имеет правый барьер, $b_f^R \geqslant 2$. Тогда выполняется соотношение $e_f^R + t_f^R \leqslant b_f^R - 1$.

Доказательство. Если $t_f^R = 0$, то утверждение непосредственно следует из теоремы 1, поэтому далее предполагаем $t_f^R \geqslant 1$. Так как функция f имеет барьер, то она совершенно уравновешена, а следовательно, и уравновешена, откуда по лемме 3 получаем $t_f^R \leqslant n-1$. Введём обозначение $t = \min\{t_f^R, b_f^R - 1\}$ и рассмотрим набор $(a_1^*, a_2^*, \ldots, a_{n-t}^*) \in V_{n-t}$, такой, что функция $f'(x_{n-t+1}, x_{n-t+2}, \ldots, x_n) \equiv \equiv f(a_1^*, a_2^*, \ldots, a_{n-t}^*, x_{n-t+1}, x_{n-t+2}, \ldots, x_n)$ имеет алгебраическую степень t.

Пусть $a^* = (0, 0, \ldots, 0, a_1^*, a_2^*, \ldots, a_{n-t}^*)$. Рассмотрим отображение $f^{a^*}_{R,b_f^R-1,n-1}$. Оно представимо в следующем виде:

$$f^{a^*}_{R,b_f^R-1,n-1}(x_n, x_{n+1}, \ldots, x_{b_f^R+n-2}) \equiv$$

$$\equiv \left(g^{(1)}(x_n), g^{(2)}(x_n, x_{n+1}), \ldots, g^{(t)}(x_n, x_{n+1}, \ldots, x_{n+t-1}), \ldots, g^{(b_f^R-1)}(x_n, x_{n+1}, \ldots, x_{n+b_f^R-2})\right).$$

Заметим, что функция $g^{(t)}$ в точности равна функции f'. Введя для функции $g^{(t)}$ фиктивные переменные в количестве $b_f^R - 1 - t$, перейдём к рассмотрению $\tilde{g}(x_1, x_2, \ldots, x_{b_f^R-1}) \equiv g^{(t)}(x_1, x_2, \ldots, x_t)$.

С одной стороны, $\mathrm{wt}(\tilde{g}) = 2^{b_f^R-1-t} \cdot \mathrm{wt}(g^{(t)})$. Так как $g^{(t)} \equiv f'$ имеет алгебраическую степень t, то в соответствии с леммой 3 значение $\mathrm{wt}(g^{(t)})$ нечётно, поэтому $\mathrm{wt}(\tilde{g})$ не делится на $2^{b_f^R-t}$.

С другой стороны, как следует из теоремы 1,

$$\mathrm{wt}(\tilde{g})=\left|\tilde{g}^{-1}(1)\right|=\sum_{\substack{y_1,y_2,\dots,y_{t-1},\\ y_{t+1},\dots,y_{b_f^R-1}\in\{0,1\}}}\left|{f^{x^*}_{R,b_f^R-1,n-1}}^{-1}(y_1,y_2,\dots,y_{t-1},1,y_{t+1},\dots,y_{b_f^R-1})\right|=C\cdot 2^{e_f^R},$$

где C — некоторое целое положительное число.

Таким образом, $2^{e_f^R}$ не делится на $2^{b_f^R-t}$, откуда получаем $e_f^R\leqslant b_f^R-t-1$. Если $t=\min\{t_f^R,b_f^R-1\}=b_f^R-1$, то получим $e_f^R\leqslant 0$, то есть, с учётом леммы 1, $b_f^R=1$, что противоречит условию. Таким образом, $t=t_f^R$ и $e_f^R\leqslant b_f^R-t_f^R-1$, что завершает доказательство теоремы. ■

Данное утверждение означает, что при наличии в полиномиальном представлении булевой функции n переменных монома, содержащего одновременно переменные $x_n,x_{n-1},\dots,x_{n-t+1}$ (то есть при $t_f^R\geqslant t$), выполнено $e_f^R\leqslant b_f^R-1-t$, что, с учётом теоремы 3, означает, что данная функция имеет не более $2^{2^{b_f^R+n-1}\left(1-2^{t+1-b_f^R}\right)}$ инверсионных функций.

Замечание 1. Заметим, что неравенство $e_f^R+t_f^R\leqslant b_f^R-1$ может обращаться, а может и не обращаться в равенство. Например, для функции $f^{(1)}(x_1,x_2,x_3,x_4)=$ $=x_3\oplus x_2x_4\oplus x_1x_2x_4$ верно, что $b^R_{f^{(1)}}=3$, $e^R_{f^{(1)}}=1$, $t^R_{f^{(1)}}=1$, то есть $e^R_{f^{(1)}}+t^R_{f^{(1)}}=b^R_{f^{(1)}}-1$; для функции $f^{(2)}(x_1,x_2,x_3,x_4,x_5)=x_4\oplus x_1x_5\oplus x_1x_2x_3x_5$ верно, что $b^R_{f^{(2)}}=4$, $e^R_{f^{(2)}}=1$, $t^R_{f^{(2)}}=1$, то есть $e^R_{f^{(2)}}+t^R_{f^{(2)}}<b^R_{f^{(2)}}-1$.

Абсолютно аналогично теореме 4 доказывается следующее утверждение.

Теорема 5. Пусть функция $f\in\mathcal{F}_n$ имеет левый барьер, $b_f^L\geqslant 2$. Тогда выполняется соотношение $e_f^L+t_f^L\leqslant b_f^L-1$.

Следствие 2. Пусть функция $f\in\mathcal{F}_n$ имеет правый барьер, $2\leqslant b_f^R\leqslant n$. Тогда в полиноме функции $f(x_1,x_2,\dots,x_n)$ нет мономов, содержащих одновременно переменные $x_n,x_{n-1},\dots,x_{n-b_f^R+2}$.

Доказательство. Как следует из теоремы 1 и леммы 1, для функции f верно $e_f^R\geqslant 1$. По теореме 4 верно $e_f^R+t_f^R\leqslant b_f^R-1$, откуда $t_f^R\leqslant b_f^R-2$, то есть в полиноме функции $f(x_1,x_2,\dots,x_n)$ не может быть мономов, содержащих одновременно переменные $x_n,x_{n-1},\dots,x_{n-b_f^R+2}$. ■

Для произвольной функции $f\in\mathcal{F}_n$ с правым барьером длины 3 с помощью следствия 2 можно получить, что в разложении $f(x_1,x_2,\dots,x_n)=f_{(00)}(x_1,x_2,\dots,x_{n-2})\oplus$ $\oplus x_{n-1}f_{(01)}(x_1,x_2,\dots,x_{n-2})\oplus x_nf_{(10)}(x_1,x_2,\dots,x_{n-2})\oplus x_nx_{n-1}f_{(11)}(x_1,x_2,\dots,x_{n-2})$ функция $f_{(11)}$ тождественно равна нулю. Таким образом, полученное в работе [6] необходимое условие барьера длины 3 является тривиальным частным следствием полученного общего утверждения.

Следствие 3. Пусть функция $f\in\mathcal{F}_n$ имеет левый барьер, $2\leqslant b_f^L\leqslant n$. Тогда в полиноме функции $f(x_1,x_2,\dots,x_n)$ нет мономов, содержащих одновременно переменные $x_1,x_2,\dots,x_{b_f^L-1}$.

Теорема 6. Пусть $n\geqslant 2$. Если для некоторой функции $f\in\mathcal{F}_n$, отличной от x_1 и $x_1\oplus 1$, верно $b_f^R\geqslant n$, то $e_f^R\leqslant b_f^R-3$.

Доказательство. Будем доказывать индукцией по n. В случае $n\leqslant 2$ утверждение очевидно, так как все функции из $\mathcal{F}_2$ с правым барьером длины 2 — это функции x_1 и $x_1\oplus 1$, а функций двух переменных с барьером большей длины не существует.

Пусть $n \geqslant 3$. Предположим противное: существует функция $f \in \mathcal{F}_n$, не равная x_1 и $x_1 \oplus 1$, такая, что $b_f^R \geqslant n$, $e_f^R \geqslant b_f^R - 2$. При этом, с учётом теоремы 1 и леммы 2, $e_f^R \leqslant n-1$, $e_f^R < b_f^R - 1$, поэтому необходимо рассмотреть только такие функции f, что $b_f^R = n$, $e_f^R = b_f^R - 2 = n-2$. Если при этом f не зависит существенно от x_n, то для функции $f'(x_1, x_2, \ldots, x_{n-1}) \equiv f(x_1, x_2, \ldots, x_{n-1}, 0)$, $f' \in \mathcal{F}_{n'}$, $n' = n-1$, верно $b_{f'}^R = b_f^R - 1 = n'$, $e_{f'}^R = e_f^R - 1 = n' - 2$ и противоречие следует из предположения индукции. Пусть теперь f существенно зависит от x_n. В соответствии с теоремой 4 $t_f^R \leqslant (b_f^R - 1) - (b_f^R - 2) = 1$. Таким образом:

1) для всякого набора констант $(a_1, a_2, \ldots, a_{n-2}) \in V_{n-2}$ функция $f(a_1, a_2, \ldots, a_{n-2}, x_{n-1}, x_n)$ либо линейна по x_n, либо не зависит существенно от x_n;
2) существует набор констант $(a_1^*, a_2^*, \ldots, a_{n-2}^*) \in V_{n-2}$, такой, что функция $f(a_1^*, a_2^*, \ldots, a_{n-2}^*, x_{n-1}, x_n)$ существенно зависит от x_n.

Отсюда следует, что функция $f(a_1^*, a_2^*, \ldots, a_{n-2}^*, x_{n-1}, x_n)$ линейна по переменной x_n, а значит, функции $f(a_1^*, a_2^*, \ldots, a_{n-2}^*, 0, x_n)$ и $f(a_1^*, a_2^*, \ldots, a_{n-2}^*, 1, x_n)$ также линейны по x_n. При этом, так как $b_f^R = n$ и $e_f^R = n-2$, то, по следствию 1,

$$\left|\mathrm{Im}\left(f_{R,n-1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,0)}\right)\right| = \left|\mathrm{Im}\left(f_{R,n-1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,1)}\right)\right| = 2. \tag{1}$$

С учётом линейной зависимости первых компонент значений векторных отображений $f_{R,n-1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,0)}$ и $f_{R,n-1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,1)}$ от переменной x_n имеем

$$\left|\mathrm{Im}\left(f_{R,1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,0)}\right)\right| = \left|\mathrm{Im}\left(f_{R,1,n-1}^{(a_1^*,a_2^*,\ldots,a_{n-2}^*,1)}\right)\right| = 2. \tag{2}$$

Из равенств (1) и (2) получаем

$$\left|\mathrm{Im}\left(f_{R,n-2,n-1}^{(a_2^*,a_3^*,\ldots,a_{n-2}^*,d_1,d_2)}\right)\right| = 1 \tag{3}$$

для любых $d_1, d_2 \in \{0,1\}$.

Из (3) следует, что при всяких $d_1, d_2 \in \{0,1\}$ после фиксации начальных $n-1$ переменных отображения f_{n-2} значениями $a_2^*, a_3^*, \ldots, a_{n-2}^*, d_1, d_2$ получаем постоянный вектор: $f_{n-2}(a_2^*, a_3^*, \ldots, a_{n-2}^*, d_1, d_2, x_1, x_2, \ldots, x_{n-2}) \equiv (q_1^{d_1,d_2}, q_2^{d_1,d_2}, \ldots, q_{n-2}^{d_1,d_2}) \in V_{n-2}$. Следовательно, при произвольной фиксации первых двух переменных d_1 и d_2 функция f обращается в константную: $f(d_1, d_2, x_1, x_2, \ldots, x_{n-2}) \equiv q_{n-2}^{d_1,d_2}$. Следовательно, f зависит существенно только от первых двух переменных, что противоречит существенной зависимости от последней переменной. Полученное противоречие завершает доказательство утверждения. ■

Теорема 7. Пусть $n \geqslant 2$. Если для некоторой функции $f \in \mathcal{F}_n$, отличной от x_n и $x_n \oplus 1$, верно $b_f^L \geqslant n$, то $e_f^L \leqslant b_f^L - 3$.

ЛИТЕРАТУРА

1. *Preparata F. P.* Convolutional transformations of binary sequences: Boolean functions and their resynchronizing properties // IEEE Trans. Electron. Comput. 1966. V. 15. No. 6. P. 898–909.
2. *Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В.* Булевы функции в теории кодирования и криптологии. 2-е изд. М.: МЦНМО, 2012.
3. *Golić J. Dj.* On the security of nonlinear filter generators // LNCS. 1996. V. 1039. P. 173–188.

4. *Сумароков С. Н.* Запреты двоичных функций и обратимость для одного класса кодирующих устройств // Обозрение прикладной и промышленной математики. 1994. Т. 1. Вып. 1. С. 33–55.
5. *Smyshlyaev S. V.* Perfectly balanced Boolean functions and Golić conjecture // J. Cryptology. 2012. No. 25(3). P. 464–483.
6. *Логачев О. А., Смышляев С. В., Ященко В. В.* Новые методы изучения совершенно уравновешенных булевых функций // Дискретная математика. 2009. Т. 21. Вып. 2. С. 51–74.
7. *Смышляев С. В.* Булевы функции без предсказывания // Дискретная математика. 2011. Т. 23. Вып. 1. С. 102–118.
8. *Смышляев С. В.* О свойствах булевых функций без предсказывания // Материалы Шестой Междунар. науч. конф. по проблемам безопасности и противодействия терроризму (МГУ им. М. В. Ломоносова, Москва, 11–12 ноября 2010). М.: МЦНМО, 2011. С. 47–56.
9. *Lai X. and Massey J.* Some connections between scramblers and invertible automata // Proc. 1988 Beijing Int. Workshop on Info. Theory. Beijing, China, July 4–8, 1988. P. DI-5.1–DI-5.5.
10. *Смышляев С. В.* О преобразовании двоичных последовательностей с помощью совершенно уравновешенных булевых функций // Материалы Пятой Междунар. науч. конф. по проблемам безопасности и противодействия терроризму (МГУ им. М. В. Ломоносова, Москва, 29–30 октября 2009). М.: МЦНМО, 2010. С. 31–41.
11. *Смышляев С. В.* О криптографических слабостях некоторых классов преобразований двоичных последовательностей // Прикладная дискретная математика. 2010. № 1(7). С. 5–15.

2013 Теоретические основы прикладной дискретной математики №2(20)

DOI 10.17223/20710410/20/4

УДК 519.719.325

АДДИТИВНЫЙ ПОДХОД К ОПРЕДЕЛЕНИЮ СТЕПЕНИ НЕЛИНЕЙНОСТИ ДИСКРЕТНОЙ ФУНКЦИИ НА ЦИКЛИЧЕСКОЙ ГРУППЕ ПРИМАРНОГО ПОРЯДКА[1]

А. В. Черемушкин

Институт криптографии связи и информатики, г. Москва, Россия

E-mail: avc238@mail.ru

Предлагается подход к определению степени нелинейности дискретных функций, заданных на циклической группе примарного порядка. Найдены верхние оценки степени нелинейности. Показано, что для полиномиальных функций над кольцом $\mathbb{Z}_{p^n}$ степень нелинейности функции совпадает с минимальной степенью многочлена, задающего эту функцию.

Ключевые слова: *дискретные функции, степень нелинейности.*

Введение

В работе [1] предложен аддитивный подход к определению степени нелинейности функции на основе свойств конечных производных. Его суть заключается в следующем. Для функций $F : G \to H$, у которых на множествах G и H заданы структуры абелевых групп, *производная по направлению* $\Delta_a F$, $a \in G$, функции F определяется равенствами

$$\Delta_a F(x) = F(x + a) - F(x),$$

где $x \in G$. *Степенью нелинейности* функции $F : G \to H$ (обозначается $\mathrm{dl}\, F$) называется минимальное натуральное число m, такое, что

$$\Delta_{a_1} \dots \Delta_{a_{m+1}} F(x) = 0$$

при всех $a_1, \dots, a_{m+1}, x \in G$. Если такого числа m не существует, то полагаем $\mathrm{dl}\, F = \infty$.

В [1] показано, что в случае элементарных абелевых p-групп степень нелинейности функции p^n-значной логики полностью определяется свойствами операции сложения. Поэтому при любом способе задания на этой группе операции умножения так, чтобы в результате получилось поле из p^n элементов, степень нелинейности функций инвариантна по отношению к выбору операции умножения.

В случае произвольных абелевых групп вопрос о свойствах параметра $\mathrm{dl}\, F$ остается открытым.

В данной работе изучается случай циклических групп. Показано, что параметр dl может принимать конечное значение только в случае, когда циклические группы являются p-группами при некотором $p \geqslant 2$ (теорема 1). Для случая примарных циклических групп найдены верхние оценки степени нелинейности (теоремы 2 и 3) и показано, что для полиномиальных функций над кольцом $\mathbb{Z}_{p^n}$ степень нелинейности функции совпадает с минимальной степенью многочлена, задающего эту функцию (теорема 6).

[1]Работа выполнена при поддержке гранта Президента РФ НШ №6260.2012.10.

1. Вспомогательные утверждения

В дальнейшем потребуются следующие свойства биномиальных коэффициентов.

Лемма 1. Пусть $\nu_p\{k\}$ — максимальная степень числа p, делящая число k. Тогда при $1 \leqslant k \leqslant p^n$ справедливо равенство

$$\nu_p\left\{\binom{p^n}{k}\right\} = n - \nu_p\{k\}.$$

Доказательство. Рассмотрим числитель и знаменатель выражения для $\binom{p^n}{k}$:

$$\binom{p^n}{k} = \frac{p^n(p^n-1)\cdot\ldots\cdot(p^n-k+1)}{1\cdot 2\cdot\ldots\cdot k}.$$

Нетрудно видеть, что при $1 \leqslant i \leqslant k-1$ выполнено равенство $\nu_p\{p^n - i\} = \nu_p\{i\}$. Поэтому $\nu_p\left\{\binom{p^n}{k}\right\} = \nu_p\left\{\frac{p^n}{k}\right\} = \nu_p\{n\} - \nu_p\{k\} = n - \nu_p\{k\}$. ■

В частности, минимальное значение при $1 \leqslant k \leqslant p^n - 1$ достигается для $k = ip^{n-1}$, $1 \leqslant i \leqslant p-1$, и равно

$$\nu_p\left\{\binom{p^n}{ip^{n-1}}\right\} = \nu_p\left\{\binom{p^n}{p^{n-1}}\right\} = 1.$$

Лемма 2. При всех $2 \leqslant k \leqslant n$ и $1 \leqslant i \leqslant p-1$ справедливо равенство

$$\binom{p^k}{ip^{k-1}} \equiv \binom{p^{k-1}}{ip^{k-2}} \pmod{p^k}.$$

Доказательство. Имеем

$$\binom{p^k}{ip^{k-1}} \equiv \frac{p^k(p^k-1)\cdot\ldots\cdot(p^k-pa)\cdot\ldots\cdot(p^k-ip^{k-1}+1)}{1\cdot 2\cdot\ldots\cdot pa\cdot\ldots\cdot(ip^{k-1}-1)ip^{k-1}} \equiv$$
$$\equiv \frac{p(p^k-1)(p^k-2)\cdot\ldots\cdot(p^{k-1}-a)\cdot\ldots\cdot(p^k-ip^{k-1}+1)}{1\cdot 2\cdot\ldots\cdot a\cdot\ldots\cdot(ip^{k-1}-1)i} \pmod{p^k}.$$

Заметим, что в знаменателе все сомножители, не кратные p, являются обратимыми элементами кольца $\mathbb{Z}_{p^n}$. Каждому сомножителю вида pa соответствует в числителе сомножитель вида $p^k - pa$, $1 \leqslant a < ip^{k-1} - 1$. Всего сомножителей, кратных p, ровно ip^{k-2}.

Представим теперь последнюю дробь в виде произведения двух сомножителей, в первый из которых включим все сомножители числителя и знаменателя, взаимно простые с p, а во второй — все кратные p, разделив каждый из них на p:

$$\frac{(p^k-1)(p^k-2)\cdot\ldots\cdot(p^k-ip^{k-1}+1)}{1\cdot 2\cdot\ldots\cdot(ip^{k-1}-1)} \cdot \frac{p^{k-1}(p^{k-1}-1)\cdot\ldots\cdot(p^{k-1}-ip^{k-2}+1)}{1\cdot 2\cdot\ldots\cdot(ip^{k-2}-1)ip^{k-2}}.$$

Осталось заметить, что для каждого j, лежащего в интервале $1 \leqslant j \leqslant ip^{k-1}-1$ и взаимно простого с p, выполняется сравнение

$$\frac{p^k-j}{j} \equiv \frac{-j}{j} \equiv -1 \pmod{p^k}.$$

Таким образом,

$$\binom{p^k}{ip^{k-1}} \equiv \frac{(-1)(-2)\cdot\ldots\cdot(-ip^{k-1}+1)}{1\cdot 2\cdot\ldots\cdot(ip^{k-1}-1)}\binom{p^{k-1}}{ip^{k-2}} \equiv (-1)^{ip^{k-1}-ip^{k-2}}\binom{p^{k-1}}{ip^{k-2}} \pmod{p^k}.$$

При $p>2$ или при $p=2$ и $k>2$ число $ip^{k-1}-ip^{k-2}=ip^{k-2}(p-1)$ является чётным. При $p=k=2$ имеем $\binom{4}{2}\equiv\binom{2}{1} \pmod{2^2}$. ■

Следствие 1. При всех $2\leqslant k\leqslant n$ и $1\leqslant i\leqslant p-1$ справедливы сравнения

$$\binom{p^k}{ip^{k-1}} \equiv \binom{p^{k-j}}{ip^{k-j-1}} \pmod{p^{k-j}}, \quad 1\leqslant j\leqslant k-1,$$

в частности,

$$\binom{p^n}{ip^{n-1}} \equiv \binom{p}{i} \pmod{p}.$$

Доказательство. Достаточно показать, что при $3\leqslant k\leqslant n$ выполнено сравнение

$$\binom{p^k}{ip^{k-1}} \equiv \binom{p^{k-2}}{ip^{k-3}} \pmod{p^{k-1}}.$$

Имеем

$$\binom{p^k}{ip^{k-1}} \equiv \binom{p^{k-1}}{ip^{k-2}} \pmod{p^k},$$

$$\binom{p^{k-1}}{ip^{k-2}} \equiv \binom{p^{k-2}}{ip^{k-3}} \pmod{p^{k-1}}.$$

Осталось заметить, что если $a\equiv b \pmod{p^k}$, то $a\equiv b \pmod{p^{k-1}}$. ■

Аналогично лемме 2 доказывается

Лемма 3. При $1\leqslant k\leqslant n$, $1\leqslant r\leqslant p-1$ и $1\leqslant i\leqslant p-1$ справедливо равенство

$$\binom{p^k-rp^{k-1}}{ip^{k-1}} \equiv \binom{p^{k-1}-rp^{k-2}}{ip^{k-2}} \pmod{p^k}.$$

Следствие 2. При всех $2\leqslant k\leqslant n$, $1\leqslant r\leqslant p-1$ и $1\leqslant i\leqslant p-1$ справедливы сравнения

$$\binom{p^k-rp^{k-1}}{ip^{k-1}} \equiv \binom{p^{k-j}-rp^{k-j-1}}{ip^{k-j-1}} \pmod{p^{k-j}}, \quad 1\leqslant j\leqslant k-1,$$

в частности,

$$\binom{p^k-rp^{k-1}}{ip^{k-1}} \equiv \binom{p-r}{i} \pmod{p}.$$

Доказательство проводится аналогично.

Следствие 3. При всех $2\leqslant k\leqslant n$r и $1\leqslant i\leqslant p-1$ выполняется сравнение

$$(-1)^i\binom{p^k-p^{k-1}}{ip^{k-1}} \equiv (-1)^i\binom{p-1}{i} \equiv 1 \pmod{p}.$$

Доказательство. С учётом предыдущего следствия достаточно показать, что

$$(-1)^i\binom{p-1}{i} \equiv 1 \pmod p.$$

Воспользуемся индукцией по i. При $i = 1$ сравнение очевидно. Если доказано, что

$$(-1)^{i-1}\binom{p-1}{i-1} \equiv 1 \pmod p,$$

то с учётом $(i, p) = 1$ получаем

$$(-1)^i\binom{p-1}{i} - 1 = (-1)^i\binom{p-1}{i-1} \cdot \frac{p-i}{i} - 1 \equiv (-1)\left(\frac{p-i}{i} + 1\right) \equiv 0 \pmod p.$$

Следствие доказано. ■

2. Сведение к случаю примарных циклических групп

Покажем, что параметр dl F в случае, когда на множестве аргументов и значений заданы структуры циклических групп, может принимать конечные значения только тогда, когда порядки групп являются примарными числами.

Теорема 1. Если $F : G^m \to H$, где G и H — циклические группы порядков $g \geqslant 2$ и $h \geqslant 2$ соответственно, причём dl $F < \infty$, то выполнены равенства $g = p^n$ и $h = p^k$ при некоторых простом $p \geqslant 2$, натуральных $n \geqslant 1$ и $k \geqslant 1$.

Доказательство. Рассмотрим сначала случай $m = 1$. Пусть p — простой делитель числа h и $F'(x) = F(x) \pmod p$, $x \in G$. Представим значения функции F в виде вектора $(F'(1), \ldots, F'(g-1), F'(0))$. Действие производной Δ_1 на функцию F' можно представить как умножение вектора значений на матрицу A над полем $\mathbb{Z}_p$, где

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

В этой матрице сумма строк равна нулю. Характеристический многочлен матрицы A над полем $\mathbb{Z}_p$ равен

$$\chi_A(\lambda) = |A - \lambda E| = \lambda\left((-1)^{g-1}(\lambda+1)^{g-1} + (-1)^{g-2}\right) = (-1)^{g-1}\lambda((\lambda+1)^{g-1} - 1) =$$
$$= (-1)^{g-1}\lambda\left(\lambda^{g-1} + \binom{g}{1}\lambda^{g-2} + \binom{g}{2}\lambda^{g-3} + \cdots + \binom{g}{g-2}\lambda + \binom{g}{g-1}\right).$$

При $g = p^n$ все биномиальные коэффициенты в записи характеристического многочлена равны нулю по модулю p (лемма 1). Поэтому $\chi_A(\lambda) = (-1)^{g-1}\lambda^g \pmod p$ и граф линейного преобразования с матрицей A содержит один единичный цикл в точке $(0, 0, \ldots, 0)$, а все остальные векторы лежат на подходах к этому циклу (см., например, [2]). Отсюда следует, что последовательность производных $\Delta_1^i F'$, $i = 1, 2, \ldots$, всегда сходится к нулю.

Если же g не является примарным числом вида p^n, то среди биномиальных коэффициентов в записи характеристического многочлена найдутся не сравнимые с нулем по

модулю p. Поэтому характеристический многочлен $\chi_A(\lambda)$ не равен $(-1)^{g-1}\lambda^g \pmod p$ и в графе линейного преобразования с матрицей A есть циклы, отличные от единичного в точке $(0,0,\dots,0)$. Следовательно, последовательность значений производных при соответствующих значениях аргумента является периодической и степени производной $\Delta_1^i F$ никогда не обратятся в нуль.

Теперь легко видеть, что число h также должно быть примарным. Если оно составное, то, проводя аналогичные рассуждения для делителей числа h, отличных от p, убеждаемся, что сходимость к нулю не будет иметь места.

Пусть теперь $m > 1$. Если для функции от m переменных последовательность производных сходится к нулю, то и для функций от одной переменной, полученных фиксацией всех переменных, кроме одной, также выполнено это свойство. Поэтому, в силу доказанного выше, порядок группы G должен быть примарным числом. ■

Заметим, что теорему 1 можно сформулировать в более общем виде.

Теорема 1′. Если $F : G_1 \times \cdots \times G_m \to H_1 \times \cdots \times H_t$, где G_i и H_j — циклические группы, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant t$, причём $\operatorname{dl} F < \infty$, то найдётся простое число $p \geqslant 2$, такое, что $|G_i| = p^{n_i}$ и $|H_j| = p^{k_j}$ при некоторых $n_i \geqslant 1$, $1 \leqslant i \leqslant m$, $k_j \geqslant 1$, $1 \leqslant j \leqslant t$.

Доказательство по сути остается неизменным.

3. Оценка параметра $\operatorname{dl}(F)$ для функций на примарных циклических группах

Пусть на множестве аргументов и значений произвольной функции $F : G \to H$ задана структура циклической абелевой p-группы примарного порядка: $G = \mathbb{Z}_{p^n}$ и $H = \mathbb{Z}_{p^k}$, $p \geqslant 2$, $n, k \geqslant 1$. Покажем, что параметр $\operatorname{dl} F$ принимает конечное значение.

Теорема 2. Если $F : G \to H$, $G = \mathbb{Z}_{p^n}$, $H = \mathbb{Z}_{p^k}$, $p \geqslant 2$, то

$$\operatorname{dl} F \leqslant p^n + (k-1)(p-1)p^{n-1} - 1.$$

Доказательство. Так как производная суммы функций равна сумме производных, то достаточно рассмотреть случай, когда функция F принимает единственное ненулевое значение. Пусть $F(1) = \cdots = F(p^n - 1) = 0$ и $F(0) = 1$. Рассмотрим последовательность производных

$$\Delta_a F, \quad \Delta_a^2 F = \Delta_a \Delta_a F, \quad \Delta_a^3 F = \Delta_a \Delta_a \Delta_a F, \quad \dots$$

Сначала рассмотрим случай $a = 1$. Соберём значения производных в табл. 1 для $p = 2$ и табл. 2 для $p > 2$ (все значения в таблицах приводятся по модулю p^k).

Имеем при $0 \leqslant i \leqslant p^n - 1$

$$\Delta_1^j F(p^n - i) = (-1)^{i+j} \binom{j}{i} \mod p^k.$$

При $i = p^n$ значение $\Delta_1^{p^n} F(0)$ равно 2 при $p = 2$ и нулю при $p > 2$.

Из таблиц и из леммы 1 следует, что производная $\Delta_1^{p^n} F$ является первой, для которой все значения принадлежат идеалу $p\mathbb{Z}_{p^k}$. Сформулируем свойства элементов в строке с номером p^n более точно:

1) p значений, соответствующих значениям аргумента вида ip^{n-1}, $0 \leqslant i < p$, лежат в $p\mathbb{Z}_{p^k}$;
2) $p^2 - p$ значений, соответствующих оставшимся значениям аргумента вида ip^{n-2}, $0 \leqslant i < p^2$, лежат в $p^2\mathbb{Z}_{p^k}$;

Таблица 1

Значения производных $\Delta_1^i F$ для случая $p = 2$

x	1	2	...	2^n-4	2^n-3	2^n-2	2^n-1	0
F	0	0	...	0	0	0	0	1
$\Delta_1 F$	0	0	...	0	0	0	1	-1
$\Delta_1^2 F$	0	0	...	0	0	1	-2	1
$\Delta_1^3 F$	0	0	...	0	1	-3	3	-1
$\Delta_1^4 F$	0	0	...	1	-4	6	-4	1
...	...	...	...	...	...	...	...	...
$\Delta_1^{2^n-1} F$	1	$-(2^n-1)$	...	$-\binom{2^n-1}{4}$	$\binom{2^n-1}{3}$	$-\binom{2^n-1}{2}$	2^n-1	-1
$\Delta_1^{2^n} F$	0	0	...	$\binom{2^n}{4}$	$\binom{2^n}{3}$	$\binom{2^n}{2}$	0	2
...	...	...	...	...	...	...	...	...

Таблица 2

Значения производных $\Delta_1^i F$ для случая $p > 2$

x	1	2	...	p^n-4	p^n-3	p^n-2	p^n-1	0
F	0	0	...	0	0	0	0	1
$\Delta_1 F$	0	0	...	0	0	0	1	-1
$\Delta_1^2 F$	0	0	...	0	0	1	-2	1
$\Delta_1^3 F$	0	0	...	0	1	-3	3	-1
$\Delta_1^4 F$	0	0	...	1	-4	6	-4	1
...	...	...	...	...	...	...	...	...
$\Delta_1^{p^n-1} F$	1	$-(p^n-1)$	...	$\binom{p^n-1}{4}$	$-\binom{p^n-1}{3}$	$\binom{p^n-1}{2}$	$-(p^n-1)$	1
$\Delta_1^{p^n} F$	0	0	...	$-\binom{p^n}{4}$	$\binom{p^n}{3}$	$-\binom{p^n}{2}$	0	0
...	...	...	...	...	...	...	...	...

3) p^3-p^2 значений, соответствующих оставшимся значениям аргумента вида ip^{n-3}, $0 \leqslant i < p^3$, лежат в $p^3\mathbb{Z}_{p^k}$, и т. д.;
4) $p^{n-1} - p^{n-2}$ значений, соответствующих оставшимся значениям аргумента вида ip, $0 \leqslant i < p^{n-1}$, лежат в $p^{n-1}\mathbb{Z}_{p^k}$;
5) остальные $(p-1)p^{n-1}$ элементов лежат в $p^n\mathbb{Z}_{p^k}$.

Из лемм 1–3 следует также, что для других строк выполнены следующие свойства:

1) через каждые p шагов в строках с номерами, кратными p, стоят производные, у которых все значения кратны p, за исключением тех, которые соответствуют аргументам вида ip, $0 \leqslant i < p^{n-1}$;
2) через каждые p^2 шагов в строках с номерами, кратными p^2, стоят производные, у которых все значения кратны p^2, за исключением p^{n-2} значений, соответствующих аргументам вида ip^2, $0 \leqslant i < p^{n-2}$, лежащих в $\mathbb{Z}_{p^k}$, и $p^{n-1}-p^{n-2}$ значений, соответствующих оставшимся значениям аргумента вида ip, $0 \leqslant i < p^{n-1}$, лежащих в $p\mathbb{Z}_{p^k}$, и т. д.;
3) через каждые p^{n-1} шагов в строках с номерами, кратными p^{n-1}, стоят производные, у которых все значения кратны p^{n-1}, за исключением тех, которые соответствуют аргументам вида ip, $0 \leqslant i < p^{n-1}$.

Покажем, что все значения $\Delta_1^{p^n+(p-1)p^{n-1}} F$ лежат в $p^2\mathbb{Z}_{p^k}$.

Для этого представим функцию $\Delta_1^{p^n} F$ в виде суммы $\Delta_1^{p^n} F = F_0 + F_1$, где функция F_0 отличается от $\Delta_1^{p^n} F$ тем, что у неё значения в точках $x = ip^{n-1}$, $i = 1, \ldots, p-1$, те же, а остальные равны нулю. В этом случае все значения функции F_1 лежат в $p^2\mathbb{Z}_{p^k}$.

Рассмотрим значения функции F_0.

С помощью леммы 2 и следствия из неё можно сводить изучение значений элементов по модулю p^{t-1}, стоящих в строке с номером p^t, $1 \leqslant t \leqslant n$, на местах с номерами p^{t-1}, $2p^{t-1}$, …, $(p-1)p^{t-1}$, p^t, к рассмотрению элементов, стоящих в строке с номером p^{t-1} на местах с номерами p^{t-2}, $2p^{t-2}$, …, $(p-1)p^{t-2}$, p^{t-1}, и т. д.

Тем самым фактически для изучения значений элементов таблиц по модулю p, стоящих в строках с номерами ip^{t-1}, $1 \leqslant t \leqslant n$, $0 \leqslant i \leqslant p-1$, можно редуцировать табл. 2, уменьшив её в p^{n-1} раз до размера $p \times p$, и изучить поведение ненулевых по модулю p значений в строках с номерами i, $0 \leqslant i \leqslant p-1$. Пример, иллюстрирующий вид табл. 2 для случая $p = 5$ и $n = k = 2$, приведён в табл. 3.

Таблица 3

Значения производных для случая $p = 5$ и $n = k = 2$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
1																								**1**	**24**
2																							**1**	**23**	**1**
3																						**1**	**22**	**3**	**24**
4																					**1**	**21**	**6**	**21**	**1**
5																				**1**	**20**	**10**	**15**	**5**	**24**
6																			1	19	15	5	15	19	1
7																		1	18	21	15	10	4	7	24
8																	1	17	3	19	20	19	3	17	1
9																1	16	11	16	1	24	9	14	9	24
10															**1**	15	20	5	10	**23**	10	5	20	15	**1**
11														1	14	5	10	5	13	12	20	15	20	11	24
12													1	13	16	5	20	8	24	8	20	5	16	13	1
12												1	12	3	14	15	13	16	9	12	10	11	22	13	24
14											1	11	16	11	1	23	3	18	3	23	1	11	16	11	1
15										**1**	10	5	20	15	**22**	5	15	10	20	**3**	10	5	20	15	**24**
16									1	9	20	15	20	7	8	10	20	10	8	7	20	15	20	9	1
17								1	8	11	20	5	12	1	2	10	15	23	24	13	20	5	14	17	24
18							1	7	3	9	10	7	14	1	8	5	8	1	14	7	10	9	3	7	1
19						1	6	21	6	1	22	7	12	7	22	3	18	13	18	3	24	19	4	19	24
20					**1**	5	15	10	20	**21**	10	5	20	15	**6**	15	20	5	10	**21**	20	10	15	5	**1**
21				1	4	10	20	10	1	14	20	15	20	16	9	5	10	5	11	24	15	5	15	21	24
22			1	3	6	10	15	16	13	6	20	5	21	18	21	5	20	6	13	16	15	10	6	3	1
23		1	2	3	4	5	1	22	18	14	10	16	22	3	9	15	11	7	3	24	20	21	22	23	24
24	1	1	1	1	1	21	21	21	21	21	6	6	6	6	6	21	21	21	21	21	21	1	1	1	1
25	0	0	0	0	**20**	0	0	0	0	**10**	0	0	0	0	**15**	0	0	0	0	**5**	0	0	0	0	**0**
26	0	0	0	20	5	0	0	0	10	15	0	0	0	15	10	0	0	0	5	20	0	0	0	0	0
27	0	0	20	10	20	0	0	10	5	10	0	0	15	20	15	0	0	5	15	5	0	0	0	0	0
28	0	20	15	10	5	0	10	20	5	15	0	15	5	20	10	0	5	10	20	20	0	0	0	0	0
29	20	20	20	20	20	10	10	10	10	10	15	15	15	15	15	5	5	5	5	5	0	0	0	0	0
30	0	0	0	0	**15**	0	0	0	0	**5**	0	0	0	0	**15**	0	0	0	0	**20**	0	0	0	0	**20**
31	0	0	0	15	10	0	0	0	5	20	0	0	0	15	10	0	0	0	20	5	0	0	0	20	5
32	0	0	15	20	15	0	0	5	15	5	0	0	0	0	15	0	0	20	10	20	0	0	20	10	20
33	0	15	5	20	10	0	5	10	15	20	0	0	0	0	15	0	20	15	10	5	0	20	15	10	5
34	15	15	15	15	15	5	5	5	5	5	15	15	15	15	15	20	20	20	20	20	20	20	20	20	20
35	0	0	0	0	**15**	0	0	0	0	**10**	0	0	0	0	**5**	0	0	0	0	**0**	0	0	0	0	**20**
36	0	0	0	15	10	0	0	0	10	15	0	0	0	5	20	0	0	0	0	0	0	0	0	20	5
37	0	0	15	20	15	0	0	10	5	10	0	0	5	15	5	0	0	0	0	0	0	0	20	10	20
38	0	15	5	0	10	0	10	20	5	15	0	5	10	15	20	0	0	0	0	0	0	20	15	10	5
39	15	15	15	15	15	10	10	10	10	10	5	5	5	5	5	0	0	0	0	0	20	20	20	20	20
40	0	0	0	0	**20**	0	0	0	0	**20**	0	0	0	0	**20**	0	0	0	0	**20**	0	0	0	0	**20**
41	0	0	0	20	5	0	0	0	20	5	0	0	0	20	5	0	0	0	20	5	0	0	0	20	5
42	0	0	20	10	20	0	0	20	10	20	0	0	20	10	15	0	0	20	10	20	0	0	20	10	20
43	0	20	15	10	5	0	20	15	10	5	0	20	15	10	5	0	20	15	10	5	0	20	15	10	5
44	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
45	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**

Рассмотрим кратные производные для функции, полученной в результате такого редуцирования. Пусть

$$pR(i) = F_0(ip^{n-1}) = \Delta_1^{p^n} F(ip^{n-1}) \mod p^2, \quad 1 \leqslant i \leqslant p-1,$$

где $R : \mathbb{Z}_p \to \mathbb{Z}_p$ — функция, полученная в результате редуцирования.

Воспользуемся также следующей очевидной леммой.

Лемма 4. Для любой функции $F : G \to G$, $G = \mathbb{Z}_M$, $M > 2$, удовлетворяющей условию

$$\sum_{x=0}^{M-1} F(x) \equiv 0 \pmod M,$$

существует функция $F^* : G \to H$, такая, что $\Delta_1 F^* = F$.

Доказательство. Используя равенства $F(x) = F^*(x+1) - F^*(x)$, $0 \leqslant x \leqslant M-1$, составим систему уравнений в $\mathbb{Z}_M$ для значений функции F^*:

$$\begin{aligned}
F^*(1) &= F^*(0) + F(0),\\
F^*(2) &= F^*(1) + F(1),\\
&\dots \quad \dots\\
F^*(M-1) &= F^*(M-2) + F(M-2),\\
F^*(0) &= F^*(M-1) + F(M-1) = F^*(0) + \sum_{x=0}^{M-1} F(x).
\end{aligned}$$

Теперь, выбирая произвольное значение $F^*(0)$, можно однозначно вычислить все остальные значения функции F^*. ■

В силу леммы 4 можно считать, что сама функция R является производной некоторой функции R^*. Так как для любой функции от одной переменной по модулю p её производная степени p равна нулю, то

$$\Delta_1^{p-1} R = \Delta_1^p R^* = 0 \mod p.$$

Это означает, что $\Delta_1^{p^n+(p-1)p^{n-1}} F(x) = 0 \mod p^2$ при всех $x \in \mathbb{Z}_{p^n}$. (Другой способ доказательства этого факта основан на следствии 3. Если увеличить на 1 значения функции F_0 в точках ip^{n-1}, то получаются значения, кратные p, причем значения последующих производных в этих точках не изменятся. Поэтому через p^n шагов все значения будут кратны p^2.)

Для рассмотренного выше примера из табл. 3 для $p^n = 25$ функция R и её производные приведены в табл. 4.

Таблица 4

Значения производных функции R для случая $p = 5$ и $n = k = 2$

x	1	2	3	4	0
R^*	0	4	1	4	0
R	4	2	3	1	0
$\Delta_1^1 R$	3	1	3	4	4
$\Delta_1^2 R$	3	2	1	0	4
$\Delta_1^3 R$	4	4	4	4	4
$\Delta_1^4 R$	0	0	0	0	0

Теперь строка значений функции $\Delta_1^{p^n+(p-1)p^{n-1}} F$ обладает свойствами, аналогичными рассмотренной выше строке значений $\Delta_1^{p^n} F$:

1) p значений, соответствующих значениям аргумента вида ip^{n-1}, $0 \leqslant i < p$, лежат в $p^2\mathbb{Z}_{p^k}$;
2) $p^2 - p$ значений, соответствующих оставшимся значениям аргумента вида ip^{n-2}, $0 \leqslant i < p^2$, лежат в $p^3\mathbb{Z}_{p^k}$;
3) $p^3 - p^2$ значений, соответствующих оставшимся значениям аргумента вида ip^{n-3}, $0 \leqslant i < p^3$, лежат в $p^4\mathbb{Z}_{p^k}$, и т. д.;
4) $p^{n-2} - p^{n-3}$ значений, соответствующих оставшимся значениям аргумента вида ip^2, $0 \leqslant i < p^{n-2}$, лежат в $p^{n-1}\mathbb{Z}_{p^k}$;
5) остальные $p^n - p^{n-2}$ элементов лежат в $p^n\mathbb{Z}_{p^k}$.

Аналогично рассуждая, получаем, что все значения $\Delta_1^{p^n+2(p-1)p^{n-1}}F$ лежат в $p^3\mathbb{Z}_{p^k}$, и т. д. Таким образом, при

$$N = p^n + (k-1)(p-1)p^{n-1}$$

для всех $x \in \mathbb{Z}_{p^n}$ выполнено равенство

$$\Delta_1^N F(x) = 0 \mod p^k.$$

Теперь рассмотрим случай произвольной последовательности кратных производных. При $1 < a \leqslant p^n - 1$ равенство $\Delta_a^N F = 0 \mod p^k$ также остается справедливым. Это вытекает из следующей леммы.

Лемма 5. Для любой функции $F : \mathbb{Z}_M \to \mathbb{Z}_M$ и любого $1 < a < M$ справедливо равенство

$$\Delta_a F(x) = \Delta_1 \left(\sum_{i=0}^{a-1} F(x+i) \right).$$

Доказательство вытекает из очевидного равенства

$$F(x+a) - F(x) = \sum_{i=0}^{a-1} (F(x+i+1) - F(x+i)).$$

Рассмотрим кратную производную $\Delta_{a_1}\Delta_{a_2}\dots\Delta_{a_N}F$ для произвольной последовательности $a_1, a_2 \dots, a_N \in \mathbb{Z}_{p^n}$. Очевидным следствием леммы 5 является

Лемма 6. Для любой функции $F : \mathbb{Z}_M \to \mathbb{Z}_M$ и любых $1 < a_i < M$, $1 \leqslant i \leqslant N$, справедливо равенство

$$\Delta_{a_1}\Delta_{a_2}\dots\Delta_{a_N}F(x) = \sum_{i_1=0}^{a_1-1} \dots \sum_{i_N=0}^{a_N-1} \Delta_1^N F(x + i_1 + \dots + i_N).$$

Из леммы 6 вытекает, что при $N = p^n + (k-1)(p-1)p^{n-1}$ все кратные производные степени N обращаются в нуль. Следовательно, $\mathrm{dl}\, F \leqslant p^n + (k-1)(p-1)p^{n-1} - 1$. ■

Перейдём теперь к случаю нескольких переменных.

Теорема 3. Если $F : G_1 \times \dots \times G_m \to H$, $G_i = \mathbb{Z}_{p^{n_i}}$, $1 \leqslant i \leqslant m$, $H = \mathbb{Z}_{p^k}$, $p \geqslant 2$, то

$$\mathrm{dl}\, F \leqslant \sum_{i=1}^{m} (p^{n_i} - (k-1)(p-1)p^{n_i-1} - 1).$$

Доказательство. Рассмотрим случай $m = 2$. При всех $a_1, a_2 \in \mathbb{Z}_{p^n}$ имеем

$$\Delta_{(a_1,a_2)}F(x_1,x_2) = \Delta_{(a_1,0)}F(x_1,x_2+a_2) + \Delta_{(0,a_2)}F(x_1,x_2).$$

Пользуясь леммой 5, получаем при всех $a_1, a_2, b_1, b_2 \in \mathbb{Z}_{p^n}$

$$\Delta_{(b_1,b_2)}\Delta_{(a_1,a_2)}F(x_1,x_2) = \Delta_{(b_1,b_2)}\left(\Delta_{(1,0)}\sum_{i=0}^{a_1-1}F(x_1+i,x_2+a_2)+\Delta_{(0,1)}\sum_{j=0}^{a_2-1}F(x_1,x_2+j)\right) =$$
$$= \Delta^2_{(1,0)}H_{1,0}(x_1,x_2)+\Delta_{(1,0)}\Delta_{(0,1)}H_{1,1}(x_1,x_2)+\Delta^2_{(0,1)}H_{0,1}(x_1,x_2)$$

при некоторых функциях $H_{1,0}, H_{1,1}$ и $H_{0,1}$.

Следовательно, каждую кратную производную степени N

$$\Delta_{(a_1^{(1)},\,a_2^{(1)})}\dots\Delta_{(a_1^{(N)},\,a_2^{(N)})}F(x_1,x_2)$$

можно преобразовать к виду

$$\sum_{i=0}^{N}\Delta^i_{(1,0)}\Delta^{N-i}_{(0,1)}F_{i,N-i}(x_1,x_2)$$

при некоторых функциях $F_{i,N-i}$, $0 \leqslant i \leqslant N$.

Для того чтобы все кратные производные обратились в нуль, достаточно, чтобы число i было больше, чем $p^{n_1}+(k-1)(p-1)p^{n_1-1}$, или число $(N-i)$ — больше, чем $p^{n_2}+(k-1)(p-1)p^{n_2-1}$. Значит, при

$$N = (p^{n_1}+(k-1)(p-1)p^{n_1-1}-1)+(p^{n_2}+(k-1)(p-1)p^{n_2-1}-1)+1$$

все производные будут равны нулю. Отсюда в случае $m=2$ справедлива оценка

$$\mathrm{dl}\,F \leqslant (p^{n_1}+(k-1)(p-1)p^{n_1-1}-1)+(p^{n_2}+(k-1)(p-1)p^{n_2-1}-1).$$

В общем случае имеем оценку $\mathrm{dl}\,F \leqslant \sum\limits_{i=1}^{m}(p^{n_i}+(k-1)(p-1)p^{n_i-1}-1)$. ■

Теорема 4. Пусть $F : G_1\times\dots\times G_m \to H_1\times\dots\times H_t$, где $G=\mathbb{Z}_{p^{n_i}}$, $H=\mathbb{Z}_{p^{k_j}}$, $1\leqslant i\leqslant m$, $1\leqslant j\leqslant t$; $p\geqslant 2$; $m\geqslant 1$; $t\geqslant 1$. Если $F=(F_1,\dots,F_t)$, где F_j — соответствующие координатные функции $F_j : G_1\times\dots\times G_m \to H_j$, $1\leqslant j\leqslant t$, то

$$\mathrm{dl}\,F = \max_{1\leqslant j\leqslant t}\mathrm{dl}\,F_i \leqslant \sum_{i=1}^{m}\left(p^{n_i}+\left(\max_{1\leqslant j\leqslant t}k_j-1\right)(p-1)p^{n_i-1}-1\right).$$

Доказательство вытекает из определения параметра dl и теоремы 3.

Следствие 4. Если $F : G^m \to G^t$, где $G=\mathbb{Z}_{p^n}$, $p\geqslant 2$, $m\geqslant 1$, $t\geqslant 1$, то

$$\mathrm{dl}\,F \leqslant m(p^n+(n-1)(p-1)p^{n-1}-1) = m(p^n n-p^{n-1}(n-1)-1) = m\left(p^n\left(n-\frac{n-1}{p}\right)-1\right).$$

Преимущество данного подхода к определению степени нелинейности заключается в том, что он полностью определяется свойствами только операции сложения. Так, например, из определения степени нелинейности легко вытекает

Теорема 5. Если G и H — циклические p-группы примарного порядка и R — подгруппа в G, то для степеней нелинейности функции $F : G\to H$ и её ограничения на подгруппу R, $F|_R : R\to H$, выполнено неравенство $\mathrm{dl}\,(F|_R)\leqslant \mathrm{dl}\,F$.

4. Полиномиальные функции

В случае циклических групп, в отличие от элементарных абелевых групп, вопрос о способе выбора операции умножения представляется не таким однозначным. Если естественным образом рассматривать циклические группы примарного порядка как аддитивные группы колец вычетов $G = \mathbb{Z}_{p^n}$ и $H = \mathbb{Z}_{p^k}$ с имеющимися в них операциями умножения, то определение степени нелинейности через степень многочлена является неудобным, так как не всякая функция F может быть задана многочленом (или набором многочленов координатных функций). Полиномиальные функции над кольцом вычетов, то есть функции, которые могут быть заданы многочленом над этим кольцом, составляют относительно малую долю функций (см., например, [3, 4]). Покажем, что для полиномиальных функций над кольцом $\mathbb{Z}_{p^n}$ степень нелинейности функции совпадает с минимальной степенью многочлена, задающего эту функцию.

Теорема 6. Если $F : G^m \to G$ — полиномиальная функция над кольцом $G = \mathbb{Z}_{p^n}$, $p \geqslant 2$, $n \geqslant 1$, $m \geqslant 1$, и $P(x_1, \dots x_n) \in \mathbb{Z}_{p^n}[x_1, \dots x_n]$ — многочлен минимальной степени, задающий эту функцию, то степень нелинейности совпадает со степенью многочлена $P(x_1, \dots x_n)$:

$$\mathrm{dl}\, F = \deg P.$$

Доказательство. Как известно [3], многочлен минимальной степени, задающий функцию F, определён однозначно и может быть записан в виде

$$P(x_1, \dots x_n) = \sum_{1 \leqslant i_1 < \dots < i_l \leqslant n} \; \sum_{i + \alpha(j_1) + \dots + \alpha(j_l) < m} b^{j_1 \dots j_l}_{i_1 \dots i_l i}\, p^i (x_{i_1})_{j_1} \dots (x_{i_l})_{j_l},$$

где $(x)_j = x(x-1)\dots(x-i+1)$, $\alpha(j) = \max\{t : j! \equiv 0 \pmod{p^t}\}$, причём все коэффициенты $b^{j_1 \dots j_l}_{i_1 \dots i_l i}$ являются целыми числами, удовлетворяющими условию $1 \leqslant b^{j_1 \dots j_l}_{i_1 \dots i_l i} \leqslant p-1$.

Из свойств факториальной степени следует, что при всех $1 \leqslant i_1 < \dots < i_l \leqslant n$ и $i + \alpha(j_1) + \dots + \alpha(j_l) < m$

$$\mathrm{dl}\, (x_{i_1})_{j_1} \dots (x_{i_l})_{j_l} = j_1 + \dots + j_l = \deg (x_{i_1})_{j_1} \dots (x_{i_l})_{j_l}.$$

Поэтому

$$\mathrm{dl}\, F \leqslant \max\{j_1 + \dots + j_l\} = \deg P,$$

где максимум берется по всем одночленам, входящим в запись многочлена $P(x_1, \dots x_n)$.

Докажем, что имеет место равенство. Пусть $b^{j_1 \dots j_l}_{i_1 \dots i_l i}\, p^i (x_{i_1})_{j_1} \dots (x_{i_l})_{j_l}$ — один из одночленов максимальной степени. Для него выполняется

$$\Delta^{j_1}_{e_{i_1}} \dots \Delta^{j_l}_{e_{i_l}}\, b^{j_1 \dots j_l}_{i_1 \dots i_l i}\, p^i (x_{i_1})_{j_1} \dots (x_{i_l})_{j_l} = b^{j_1 \dots j_l}_{i_1 \dots i_l i}\, p^i i_1! \dots i_l! \not\equiv 0 \pmod{p^n}.$$

Здесь через e_i обозначен вектор $(0, \dots 0, 1, 0, \dots, 0)$, в котором единица стоит на i-м месте, соответствующем переменной x_i.

Покажем, что все остальные одночлены после взятия производной $\Delta^{j_1}_{e_{i_1}} \dots \Delta^{j_l}_{e_{i_l}}$ обратятся в нуль. Если $b^{q_1 \dots q_s}_{k_1 \dots k_s k}\, p^k (x_{k_1})_{q_1} \dots (x_{k_s})_{q_s}$ — другой одночлен максимальной степени, $j_1 + \dots + j_l = q_1 + \dots + q_s$, для которого

$$\Delta^{j_1}_{e_{i_1}} \dots \Delta^{j_l}_{e_{i_l}}\, b^{j_1 \dots j_s}_{k_1 \dots k_s k}\, p^k (x_{k_1})_{q_1} \dots (x_{k_s})_{q_s} \not\equiv 0 \pmod{p^n},$$

то для наборов переменных должно выполняться включение $\{i_1, \dots, i_l\} \subseteq \{k_1, \dots, k_s\}$, так как иначе взятие производной по несущественной переменной даст нулевое значение. Кроме того, чтобы производная не обращалась в нуль, должны выполняться

неравенства $j_h \leqslant q_h$, $1 \leqslant h \leqslant l$, но это противоречит тому, что одночлены различны. Поэтому

$$\Delta_{e_{i_1}}^{j_1} \dots \Delta_{e_{i_l}}^{j_l} F(x) \equiv b_{i_1 \dots i_l i}^{j_1 \dots j_l} p^i i_1! \dots i_l! \not\equiv 0 \pmod{p^n},$$

откуда окончательно получаем $\mathrm{dl}\, F = j_1 + \dots + j_l = \deg P$. ■

В заключение рассмотрим псевдополиномиальный способ задания функций над кольцом вычетов, в котором вместо факториальных степеней используются биномиальные коэффициенты. Рассмотрим целочисленные функции

$$\binom{x}{i} = \frac{x(x-1)\cdots(x-i+1)}{i!}$$

при $i \in \{1, \dots, n-1\}$, $x \in \mathbb{Z}$. При $i = 0$ полагают $\binom{x}{0} = 1$. При $0 \leqslant a \leqslant b \leqslant n-1$ имеют место соотношения

$$\sum_{i=a}^{b} (-1)^{i+a} \binom{i}{a} \binom{b}{i} = \delta_{ab},$$

где δ_{ab} — символ Кронекера. Если рассматривать функцию $F : \mathbb{Z}_n \to \mathbb{Z}_n$ как целочисленную $F : \mathbb{Z} \to \mathbb{Z}$, принимающую конечное число ненулевых значений в точках $0, 1, \dots, n-1$, то из данных соотношений вытекает справедливость разложения функции $F : \mathbb{Z}_n \to \mathbb{Z}_n$

$$F(x) = \sum_{i=0}^{n-1} h(i) \binom{x}{i}$$

с коэффициентами вида

$$h(i) = \sum_{a=i}^{n-1} F(a)(-1)^{i+a} \binom{i}{a},$$

называемого разложением Ньютона функции F (см., например, [5]).

Для конечной разности $\Delta_1 F(x) = F(x+1) - F(x)$ над $\mathbb{Z}$ справедливы равенства

$$\Delta_1^j \binom{x}{i} = \binom{x}{i-j}, \quad 0 \leqslant j \leqslant i.$$

Поэтому набор коэффициентов $h(i)$, $0 \leqslant i \leqslant n-1$, также можно рассматривать как целочисленную функцию, причём значение $h(i)$ совпадает со значением i-й конечной производной над $\mathbb{Z}$ функции F в точке $x = 0$:

$$h(i) = \Delta_1^i F(0).$$

Для случая нескольких переменных разложение Ньютона строится аналогично:

$$F(x_1, \dots, x_m) = \sum_{i_1, \dots, i_m = 0}^{n-1} h(i_1, \dots, i_m) \binom{x_1}{i_1} \cdots \binom{x_m}{i_m},$$

где

$$h(i_1, \dots, i_m) = \Delta_{e_1}^{i_1} \cdots \Delta_{e_m}^{i_m} f(0, \dots, 0), \qquad i_1, \dots, i_m \in \{0, \dots, n-1\},$$

$e_i = (0, \dots, 0, 1, 0, \dots, 0)$ — вектор, i-я координата которого равна 1, $1 \leqslant i \leqslant m$.

Здесь наблюдается кажущееся противоречие с рассмотренным выше подходом, так как максимальная степень нелинейности функции в целочисленном представлении равна $n-1$ (для случая одной переменной), в то время как степень нелинейности

этой же функции над кольцом вычетов может оказаться значительно превосходящей число $n-1$. Различие в значениях степени нелинейности объясняется тем, что действия оператора Δ_1 на множестве целочисленных функций и функций над кольцом вычетов отличаются. Например, над $\mathbb{Z}$ имеем

$$\Delta_1\binom{x}{i} = \binom{x+1}{i} - \binom{x}{i} = \binom{x}{i-1},$$

в то время как над $\mathbb{Z}_n$ имеем

$$\Delta_1\binom{x}{i} = \left(\binom{x+1 \mod n}{i} - \binom{x}{i}\right) \mod n, \quad 0 \leqslant i \leqslant n-1,$$

причём при $x = n-1$ получаются разные выражения. Например, при $n = 25$ и $x = 24$ имеем

$$\Delta_1\binom{x}{5}(24) = \binom{25}{5} - \binom{24}{5} = \binom{24}{5} \equiv 5 - 4 \equiv 1 \pmod{25},$$

в то же время, если аргументы приводить по модулю 25, то

$$\Delta_1\binom{x}{5}(24) = \binom{0}{5} - \binom{24}{5} \equiv 0 - 4 \equiv 21 \pmod{25}.$$

Более подробно о разложении Ньютона можно прочитать, например, в [5].

ЛИТЕРАТУРА

1. *Черемушкин А. В.* Аддитивный подход к определению степени нелинейности дискретной функции // Прикладная дискретная математика. 2010. № 2(8). С. 22–33.
2. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра. Учебник в 2-х т. Т. II. М.: Гелиос АРВ, 2003.
3. *Keller G. and Olson F.* Counting polynomial functions (mod p^n) // Duke Math. J. 1968. V. 35. P. 835–838.
4. *Chen Z.* On polynomial functions from $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \mathbb{Z}_{n_r}$ to $\mathbb{Z}_m$ // Discrete Math. 1996. V. 162. P. 67–76.
5. *Davio M., Deschamps J. P., and Thayse A.* Discrete and switching functions. Budapest: Academiai Kiado, 1974.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/20/5

УДК 519.7

КОД АУТЕНТИФИКАЦИИ С СЕКРЕТНОСТЬЮ НА ОСНОВЕ ПРОЕКТИВНОЙ ГЕОМЕТРИИ

А. Ю. Зубов

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

E-mail: Zubovanatoly@yandex.ru

Изучается код аутентификации с секретностью на основе проективной геометрии, обеспечивающий стойкую аутентификацию и конфиденциальность для равновероятного источника информации.

Ключевые слова: *код аутентификации с секретностью, совершенный шифр, проективная геометрия.*

1. Определения и постановка задачи

Код аутентификации (далее — A-код) — это тройка $A = (S, E, M)$ конечных множеств, называемых соответственно множествами *состояний источника*, *правил кодирования* и *сообщений*. Правило кодирования $e \in E$ — это инъективное отображение $e\colon S \to M$. Для защиты сообщений от активных атак, к которым относятся атаки имитации и подмены, отправитель и получатель выбирают общее (секретное) правило кодирования e. Отправитель вычисляет $m = e(s)$ и направляет сообщение m получателю. Критерием аутентичности полученного сообщения является условие $e^{-1}(m) \neq \varnothing$. Здесь $e^{-1} : M \to S \cup \{\varnothing\}$ — обратное к e отображение, определяемое формулой

$$e^{-1}(m) = \begin{cases} s, & \text{если } m = e(s), \\ \varnothing, & \text{если } m \notin e(S), \end{cases}$$

где $e(S) = \{e(s)\colon s \in S\}$. Предполагается, что $\varnothing \notin S$. Для удобства выбора правила кодирования индексируются *ключами* из множества K.

Матрицей кодирования A-кода A называется матрица $C(A) = (c(e, m))$ размеров $|E| \times |M|$. Строки матрицы $C(A)$ занумерованы правилами кодирования $e \in E$, столбцы — сообщениями $m \in M$. Элементы матрицы определяются равенством $c(e, m) = e^{-1}(m)$. *Матрица инцидентности* A-кода — это матрица $I(A) = (i(e, m))$ тех же размеров, где

$$i(e, m) = \begin{cases} 1, & \text{если } e^{-1}(m) \neq \varnothing, \\ 0, & \text{если } e^{-1}(m) = \varnothing. \end{cases}$$

Если в каждом столбце матрицы кодирования A-кода имеются вхождения лишь одного состояния источника, то A-код называется *декартовым* или *A-кодом без секретности*. Он характеризуется тем, что $e_1^{-1}(m) = e_2^{-1}(m)$ для любых $m \in M$ и $e_1, e_2 \in E(m)$, где $E(m) = \{e \in E\colon e^{-1}(m) \neq \varnothing\}$. Это означает, что сообщение однозначно определяет состояние источника, поэтому состояние источника не является секретом для противника. Недекартов A-код называют *A-кодом с секретностью*.

Фактически A-код с секретностью может обеспечивать конфиденциальность состояния источника. В связи с этим можно говорить о криптографической стойкости A-кода.

Стойкость A-кода к активной атаке оценивается величиной вероятности успеха атаки. При имитации противник вводит в канал связи поддельное сообщение (имитируя передачу сообщения от одного пользователя другому), а при подмене — подменяет наблюдаемое сообщение (например, модифицируя его). В обоих случаях противник рассчитывает на то, что поддельное сообщение будет принято как аутентичное. Не зная правила кодирования, противник добьётся успеха при выборе поддельного сообщения лишь с некоторой вероятностью. Если противник сможет определить по наблюдаемому сообщению правило кодирования, то он добьётся успеха в подмене с вероятностью равной 1. При этом противник сможет навязать получателю любое поддельное сообщение. Чем меньше эта вероятность, тем более стойким является A-код.

В простейшем случае речь идёт об атаках на основе не более чем одного наблюдаемого сообщения. В этом случае естественно полагать, что каждое правило кодирования используется для передачи лишь одного состояния источника. Кроме того, имеется в виду атака, в которой противник добивается успеха, когда лишь некоторое (а не выбранное по его усмотрению) сообщение принимается как аутентичное. Другими словами, противник рассчитывает лишь на навязывание «наугад». Только такие атаки рассматриваются в этой работе. Любые другие атаки, в которых противник может использовать имитацию или подмену на основе наблюдения нескольких сообщений, образованных с помощью одного и того же правила кодирования или имеющих целью навязывание выбранного состояния источника, должны рассматриваться отдельно.

Определим вероятности успеха атак. Будем полагать, что состояния источника и правила кодирования выбираются случайно и независимо друг от друга в соответствии с заданными на множествах S и E распределениями вероятностей $P(S) = (p_S(s)\colon s{\in}S)$ и $P(E) = (p_E(e)\colon e \in E)$. Пусть $\overline{S}$ и $\overline{E}$ — соответствующие случайные величины. Они полагаются независимыми и естественным образом индуцируют случайную величину $\overline{M}$ с множеством исходов M. Вероятность $p_M(m)$ вычисляется по формуле

$$p_M(m) = \sum_{e\in E(m)} p_E(e)p_S(e^{-1}(m)).$$

Обычно полагают, что правила кодирования выбираются случайно и равновероятно (в общем случае это не обязательное условие [1, 2]). Будем и мы полагать, что распределение $P(E)$ — равномерное.

Пусть $p_{\text{и}}(m)$ — вероятность того, что (при случайном выборе правила кодирования) имитируемое сообщение $m \in M$ будет принято как аутентичное. Ясно, что

$$p_{\text{и}}(m) = \sum_{e\in E(m)} p_E(e).$$

Тогда (учитывая равномерность распределения $P(E)$) определим вероятность $p_{\text{и}}$ *успеха имитации* формулой

$$p_{\text{и}} = \max_{m\in M} p_{\text{и}}(m) = \max_{m\in M} \frac{|E(m)|}{|E|}. \tag{1}$$

Вероятность $p_{\text{п}}$ *успеха подмены* определяется формулой

$$p_{\text{п}} = \sum_{m\in M} p_M(m)p_{\text{п}}(m),$$

где

$$p_{\text{п}}(m) = \max_{n \neq m} p_{\text{п}}(n \mid m),$$

а $p_{\text{п}}(n \mid m)$ — вероятность того, что (для случайно выбранной пары (e, s)) противник добьётся успеха при подмене наблюдаемого $m \in M$ сообщением $n \neq m$. Эта вероятность вычисляется по формуле

$$p_{\text{п}}(n \mid m) = \frac{1}{p_M(m)} \sum_{e \in E(m,n)} p_E(e) p_S(e^{-1}(m)),$$

где $E(m, n) = E(m) \cap E(n)$. Учитывая равномерность распределения $P(E)$, получаем

$$p_{\text{п}} = \frac{1}{|E|} \sum_{m \in M} \max_{n \neq m} \sum_{e \in E(m,n)} p_S(e^{-1}(m)). \tag{2}$$

Если распределение $P(S)$ равномерно, то формула (2) принимает вид

$$p_{\text{п}} = \frac{1}{|E| \cdot |S|} \sum_{m \in M} \max_{n \neq m} |E(m, n)|. \tag{3}$$

Будем называть ε-*стойким* A-код, для которого $\max\{p_{\text{и}}, p_{\text{п}}\} \leqslant \varepsilon$, где ε — выбранный порог стойкости. В настоящее время достаточной считается величина ε порядка 2^{-128}. Известны (см., например, [3]) достижимые нижние оценки вероятностей $p_{\text{и}}, p_{\text{п}}$:

$$p_{\text{и}} \geqslant \frac{|S|}{|M|}, \ p_{\text{п}} \geqslant \frac{|S| - 1}{|M| - 1}, \ \max\{p_{\text{и}}, p_{\text{п}}\} \geqslant \frac{1}{\sqrt{|E|}}.$$

Актуальной является разработка конструкций A-кодов, использующих общий секретный ключ для обеспечения аутентификации и конфиденциальности передаваемых сообщений. Этой теме посвящён целый ряд работ [4 – 18]. В данной работе предлагается конструкция A-кода на основе проективной геометрии, обеспечивающего ε-стойкую аутентификацию и совершенное шифрование для равновероятного источника.

2. Теорема Зингера

Пусть v, k, λ — натуральные числа. Тогда (v, k, λ)-*схемой* или *блок-схемой* называется пара (M, B), где M — множество мощности v и B — семейство k-подмножеств (блоков) множества M, таких, что любое 2-подмножество множества M содержится точно в λ блоках.

Пусть b — число блоков схемы и r — число блоков, содержащих любой выбранный элемент множества M. Тогда для параметров блок-схемы выполняются следующие соотношения:

$$\begin{cases} b \cdot k = v \cdot r, \\ r \cdot (k - 1) = \lambda \cdot (v - 1). \end{cases} \tag{4}$$

Для блок-схемы выполняется неравенство Фишера $b \geqslant v$. Блок-схема, для которой $b = v$, называется *симметричной*. Известно, что для симметричной блок-схемы справедливо равенство $r = k$ и любые два блока имеют ровно λ общих точек.

Пусть F_q — поле из q элементов. Пространство всех векторов $(a_n, \ldots, a_0)$, $a_i \in F_q$, называется *проективной геометрией* размерности n над F_q и обозначается $PG(n, q)$. Вектор $\overline{0} = (0, \ldots, 0)$ образует пустое подпространство размерности -1. Точка — подпространство размерности 0. Это множество векторов $b\overline{x}$, где $\overline{x} = (x_n, \ldots, x_0) \neq \overline{0}$,

а b пробегает все элементы из F_q. Если векторы $\overline{y}_0, \ldots, \overline{y}_h$ линейно независимы, то множество векторов $\{b_0\overline{y}_0 + \ldots + b_h\overline{y}_h \colon b_i \in F_q\}$ — подпространство S_h размерности h. Подпространство S_{n-1} называется *гиперплоскостью*. Если $(c_n, \ldots, c_0) \neq \overline{0}$, то множество всех точек $(x_n, \ldots, x_0)$, удовлетворяющих уравнению

$$c_n x_n + \ldots + c_0 x_0 = 0,$$

образует гиперплоскость, и обратно, каждая гиперплоскость может быть определена таким образом, причём $(c_n, \ldots, c_0)$ и $(sc_n, \ldots, sc_0)$, $s \neq 0$, определяют одну и ту же гиперплоскость.

Каждый из $q^{n+1} - 1$ ненулевых векторов определяет точку, и, поскольку $(x_n, \ldots, x_0)$ и $(dx_n, \ldots, dx_0)$, $d \neq 0$, определяют одну и ту же точку, всего имеется $v = (q^{n+1} - 1)/(q - 1)$ точек. Аналогично, поскольку $(c_n, \ldots, c_0)$ и $(sc_n, \ldots, sc_0)$, $s \neq 0$, определяют одну и ту же гиперплоскость, имеется v гиперплоскостей. Подпространство S_h содержит $v = (q^{h+1} - 1)/(q - 1)$ точек, а гиперплоскость — $k = (q^n - 1)/(q - 1)$ точек.

С проективной геометрией $PG(n, q)$ ассоциируется симметричная блок-схема. Это следует из теоремы Зингера [19].

Теорема 1. Гиперплоскости геометрии $PG(n, q)$, взятые в качестве блоков, и точки, взятые в качестве элементов, образуют симметричную блок-схему с параметрами

$$v = \frac{q^{n+1} - 1}{q - 1}, \quad k = \frac{q^n - 1}{q - 1}, \quad \lambda = \frac{q^{n-1} - 1}{q - 1}. \tag{5}$$

Эта схема является циклической, и точки в любой гиперплоскости определяют (v, k, λ)-разностное множество.

Симметричная блок-схема называется *циклической*, если она имеет автоморфизм α, который переставляет элементы и блоки по циклу длины v. Автоморфизмом блок-схемы называется взаимно однозначное отображение α множеств элементов и блоков схемы на множества элементов и блоков той же схемы, такое, что если x_i — элемент, а B_j — блок и

$$\begin{cases} \alpha : x_i \to x_i' = (x_i)\alpha, \\ \alpha : B_j \to (B_j')\alpha, \end{cases}$$

то $x_i \in B_j$ тогда и только тогда, когда $x_i' \in B_j'$.

Множество D, состоящее из k вычетов $a_1, \ldots, a_k$ по модулю v, называется (v, k, λ)-*разностным множеством*, если для каждого $d \neq 0 \pmod v$ существует ровно λ упорядоченных пар (a_i, a_j), $a_i, a_j \in D$, таких, что $a_i - a_j \equiv d \pmod v$.

3. A-код на основе проективной геометрии

Рассмотрим A-код, множеством сообщений и ключей которого служит множество точек геометрии $PG(n, q)$, а множества $e_\varkappa(S)$, $\varkappa \in K$, совпадают с множествами точек гиперплоскостей (произвольным образом упорядоченными). Таким образом, правило кодирования $e_\varkappa$, отвечающее ключу $\varkappa$, удовлетворяет равенству $e_\varkappa(S) = B_\varkappa$, где S — множество состояний источника, а $B_\varkappa$ — гиперплоскость, определяемая точкой $\varkappa$.

По теореме Зингера для построенного A-кода множества $e_\varkappa(S)$ составляют блоки симметричной (v, k, λ)-схемы с параметрами, указанными в (5). Для такой схемы выполняется равенство $r = k$ и любые два блока имеют ровно λ общих точек. Отсюда

следует, что для любого сообщения m и любой пары различных сообщений m_1 и m_2

$$|E(m)| = r = k = \frac{q^n - 1}{q - 1} \text{ и } |E(m, n)| = \lambda = \frac{q^{n-1} - 1}{q - 1}.$$

Поскольку для параметров блок-схемы выполняются соотношения (4), из формул (1), (3) получаем значения вероятностей успеха имитации и подмены для построенного A-кода:

$$p_{\text{и}} = \frac{q^n - 1}{q^{n+1} - 1}, \ \ p_{\text{п}} = \frac{q^{n-1} - 1}{q^n - 1}.$$

Таким образом, $\max\{p_{\text{и}}, p_{\text{п}}\} \approx 1/q$. Например, при $q = 2^{128}$ шансы на успех атаки имитации или подмены оцениваются величиной порядка 2^{-128}.

Уточним теперь выбор гиперплоскости $e_{\varkappa}(S)$ для каждого $\varkappa \in K$ и значение $e_{\varkappa}(s)$ для каждого $s \in S$, при котором A-код, рассматриваемый как шифр, реализует совершенное (по К. Шеннону [20]) шифрование. В связи с этим отметим, что любой A-код с секретностью может быть формально рассмотрен как шифр. При этом S и M — множества открытых и шифрованных текстов, $e_{\varkappa} : S \to M$ и $e_{\varkappa}^{-1} : M \to S \cup \{\varnothing\}$ — правила зашифрования и расшифрования на ключе $\varkappa \in K$. Такой взгляд на код аутентификации позволяет формально ставить вопрос о его стойкости как шифра.

Теорема 2. Правило кодирования A-кода на основе проективной геометрии можно выбрать так, чтобы A-код являлся совершенным шифром.

Доказательство. Пусть A — код аутентификации на основе проективной геометрии и $I(A)$ — его матрица инцидентности. Как было отмечено, для параметров кода A справедливы равенства $|M| = v = b = |E|$ и $r = k = |S|$, означающие, что $I(A)$ — квадратная $(0, 1)$-матрица, в каждой строке и каждом столбце которой содержится ровно k единиц. Согласно [21, теорема 4, с. 364], такая матрица представляется в виде суммы k подстановочных матриц. Поместив на места единиц в каждой из подстановочных матриц одно из k состояний источника, получим матрицу, в каждой строке и каждом столбце которой содержатся все состояния источника. Обозначим полученную матрицу через $C(A)$. Она является матрицей кодирования кода аутентификации A с данными множествами S, E, M. Покажем, что A реализует совершенное шифрование.

Для этого воспользуемся критерием совершенности, согласно которому для любых $m \in M$ и $s \in S$ выполняется равенство $p_M(m) = p_{M|S}(m|s)$. Это равенство можно записать в виде

$$\sum_{e \in E(s,m)} p_E(e) = \sum_{e \in E(m)} p_E(e) \cdot p_S(e^{-1}(m)), \tag{6}$$

где $E(s, m) = \{e \in E \colon e(s) = m\}$. С учётом условия равновероятности выбора состояний источника и правил кодирования (6) равносильно равенству

$$\frac{1}{b}|E(s, m)| = \frac{1}{bk}|E(m)|. \tag{7}$$

Из свойств матрицы $C(A)$ следует, что $|E(s, m)| = 1$ для любых $s \in S$ и $m \in M$. А поскольку $E(m) = k$, приходим к выводу о том, что равенство (7), а вместе с ним и (6) справедливо. ■

Остаётся найти подходящее разложение матрицы $I(A)$ в сумму подстановочных матриц. Для того чтобы сделать это, обратимся к доказательству (из [19]) цикличности блок-схемы в теореме Зингера.

Пусть θ — примитивный элемент поля $\mathrm{GF}(q^{n+1})$. Тогда θ — корень многочлена $F(x)$ степени $n+1$, неприводимого над $\mathrm{GF}(q)$. Пусть

$$F(x) = x^{n+1} + c_n x^n + \ldots + c_0, \;\; c_i \in \mathrm{GF}(q).$$

Поскольку $F(\theta) = 0$, выполняется соотношение

$$\theta^{n+1} = -c_0 - c_1\theta - \ldots - c_n\theta^n.$$

Отсюда следует, что для любой степени i элемента θ для подходящих элементов $a_0, \ldots, a_n \in \mathrm{GF}(q)$ выполняется равенство

$$\theta^i = a_0 + a_1\theta + \ldots + a_n\theta^n. \tag{8}$$

Тем самым устанавливается взаимно однозначное соответствие между множеством степеней элемента θ и множеством ненулевых векторов $(a_0, \ldots, a_n)$ над $\mathrm{GF}(q)$, рассматриваемых как точки геометрии $PG(n, q)$.

В условиях теоремы $v = \dfrac{q^{n+1}-1}{q-1}$, поэтому элементы $0, 1, \theta^v, \ldots, \theta^{(q-2)v}$ дают q решений уравнения $z^q = z$ и образуют подполе $\mathrm{GF}(q)$ поля $\mathrm{GF}(q^{n+1})$. Если $\theta^{sv} = t$, то $t \in \mathrm{GF}(q)$, и поэтому если (8) выполняется, то

$$\theta^{i+sv} = ta_0 + ta_1\theta + \ldots + ta_n\theta^n.$$

Следовательно, θ^i и θ^j соответствуют одной и той же точке из $PG(n, q)$ тогда и только тогда, когда $i \equiv j \pmod{v}$. Таким образом, если задано отображение α поля $\mathrm{GF}(q^{n+1})$ в себя

$$\begin{cases} \alpha : 0 \to 0, \\ \alpha : \theta^i \to \theta^{i+1}, \end{cases} \tag{9}$$

то, ввиду (8), α можно рассматривать и как отображение

$$\alpha : (a_0, a_1, \ldots, a_n) \to (-a_n c_0, a_0 - a_n c_1, \ldots, a_{n-1} - a_n c_n) \tag{10}$$

множества точек в себя. Поскольку $\theta^{v(q-1)} = 1$, отображение (9) является взаимно однозначным, и поэтому взаимно однозначно и отображение (10). Так как θ^i и θ^{i+v} соответствуют одной и той же точке, а $1, \theta, \ldots, \theta^{v-1}$ — различным точкам, отображение α в (10) переставляет все v точек по полному циклу.

Заметим теперь, что $v - qk = 1$. Из этого равенства следует, что k и v взаимно просты. Если для некоторого $j \neq 0$ отображение α^j переводит точки некоторой гиперплоскости в себя, то α^j переставляет их по циклам длины t, где t делит k. Но тогда, поскольку $\alpha^v = 1$, t должно делить v и, раз $v - qk = 1$, приходим к выводу, что $t = 1$ и $\alpha^j = 1$. Так как никакая степень отображения α, за исключением степени v, не фиксирует гиперплоскость, α должно переставлять гиперплоскости по циклу длины v. Это доказывает цикличность блок-схемы.

Пусть $q = p^d$. Рассмотрим башню полей $\mathrm{GF}(p) \subset \mathrm{GF}(q) \subset \mathrm{GF}(q^{n+1})$. Пусть θ — примитивный элемент поля $\mathrm{GF}(q^{n+1})$ над $\mathrm{GF}(q)$ и ω — примитивный элемент поля $\mathrm{GF}(q)$ над $\mathrm{GF}(p)$. Тогда для подходящих $b_i \in \mathrm{GF}(p)$ и $c_j \in \mathrm{GF}(q)$ выполняются равенства

$$\begin{array}{l} \omega^d + b_{d-1}\omega^{d-1} + \ldots + b_0 = 0, \\ \theta^{n+1} + c_n\theta^n + \ldots + c_0 = 0. \end{array} \tag{11}$$

Пусть $\theta^i \leftrightarrow (a_0, \ldots, a_{n-1})$ — взаимно однозначное соответствие из (8) и α — автоморфизм (9). Можно полагать, что множества ключей и сообщений нашего A-кода представлены в виде $K = M = \{\theta^0, \theta^1, \ldots, \theta^{v-1}\}$. Пусть $S = \{s_0, s_1, \ldots, s_{k-1}\}$ — произвольное множество состояний источника. Блоками блок-схемы, ассоциированной с геометрией $PG(n, q)$, служат гиперплоскости. Пусть

$$B_0 = \{\theta^i \mid i : \theta^i = a_0 + a_1\theta + \ldots + a_{n-1}\theta^{n-1},\ \ a_0, a_1, \ldots, a_{n-1} \in \mathrm{GF}(q)\}$$

— одна из гиперплоскостей. Запишем для удобства её элементы в виде

$$B_0 = \{\theta^{\gamma_0}, \theta^{\gamma_1}, \ldots, \theta^{\gamma_{k-1}}\}. \tag{12}$$

Положим

$$B_j = (B_0)\alpha^j = \{\theta^{\gamma_0+j}, \theta^{\gamma_1+j}, \ldots, \theta^{\gamma_{k-1}+j}\},\ \ j = 1, 2, \ldots, v-1.$$

Тогда $B_0, B_1, \ldots, B_{v-1}$ — все блоки блок-схемы, связанные в один цикл преобразованием α:

$$B_0 \overset{\alpha}{\to} B_1 \overset{\alpha}{\to} \ldots \overset{\alpha}{\to} B_{v-1} \overset{\alpha}{\to} B_0.$$

Если теперь строки и столбцы матрицы кодирования упорядочить в соответствии с последовательностью степеней $\theta : \theta^0, \theta^1, \ldots, \theta^{v-1}$, и для $\varkappa \in K$ положить $e_\varkappa(S) = B_\varkappa$, то, согласно сказанному выше, получим циклическую матрицу. При этом элементы множества S упорядочены естественным образом: $s_0, s_1, \ldots, s_{k-1}$. Мы получили искомую «раскраску» клеток матрицы инцидентности и соответствующую формулу для правила кодирования. Её можно записать в виде

$$e_{\theta^j}(s_i) = \theta^{(\gamma_i+j) \bmod v}, \tag{13}$$

где γ_i определены формулой (12).

Итоги проведённых рассуждений сформулируем в следующем утверждении.

Теорема 3. Код аутентификации с правилом кодирования (13) гарантирует успех активной атаки с вероятностью, не превосходящей $1/q$, и совершенное шифрование.

Для лучшего понимания устройства такого A-кода приведём пример.

4. Пример

Пусть $q = 2^2$ и $n = 2$. В этом случае $\mathrm{GF}(q^{n+1}) = \mathrm{GF}(2^6)$. Выберем в качестве ω корень примитивного многочлена $f(y) = y^2 + y + 1$ над полем $\mathrm{GF}(2)$, а в качестве θ — корень примитивного многочлена $F(x) = x^3 + \omega x^2 + \omega x + \omega$ над полем $\mathrm{GF}(2^2)$. Используя равенства (11) $\omega^2 = \omega + 1$ и $\theta^3 = \omega\theta^2 + \omega\theta + \omega$, вычисляем степени θ^i для $i = 0, 1, \ldots, v-1$, где $v = q^2 + q + 1 = 21$ (табл. 1).

Для большей наглядности закодируем элементы поля $\mathrm{GF}(2^2)$ числами $0, 1, 2, 3$, а операции сложения и умножения этих элементов будем производить в соответствии с операциями фактор-кольца $\mathrm{GF}(2)[y]/f(y)$ (табл. 2).

При этом элемент ω заменится на 2, а $\omega + 1$ — на 3. Учитывая, что векторы $(c_n, \ldots, c_0)$ и $(sc_n, \ldots, sc_0)$, $s \neq 0$, определяют одну и ту же гиперплоскость, получаем следующее представление элементов множеств $K = M$ (табл. 3).

Заменим θ^i вычетом $i \bmod 21$ или соответствующей тройкой $a_0a_1a_2$ согласно табл. 3. Тогда гиперплоскость B_0 из (11) имеет вид

$$B_0 = \{0, 1, 6, 8, 18\} \quad \text{или} \quad B_0 = \{100, 010, 130, 110, 120\}.$$

Таблица 1

Степени θ^i

θ^0	=	1					$(1,0,0)$
θ^1	=			θ			$(0,1,0)$
θ^2	=					θ^2	$(0,0,1)$
θ^3	=	ω	+	$\omega\theta$	+	$\omega\theta^2$	(ω,ω,ω)
θ^4	=	$(\omega+1)$	+	θ	+	θ^2	$(\omega+1,1,1)$
θ^5	=	ω	+	θ	+	$(\omega+1)\theta^2$	$(\omega,1,\omega+1)$
θ^6	=	1	+	$(\omega+1)\theta$			$(1,\omega+1,0)$
θ^7	=			θ	+	$(\omega+1)\theta^2$	$(0,1,\omega+1)$
θ^8	=	1	+	θ			$(1,1,0)$
θ^9	=			θ	+	θ^2	$(0,1,1)$
θ^{10}	=	ω	+	$\omega\theta$	+	$(\omega+1)\theta^2$	$(\omega,\omega,\omega+1)$
θ^{11}	=	1	+	$(\omega+1)\theta$	+	$(\omega+1)\theta^2$	$(1,\omega+1,\omega+1)$
θ^{12}	=	1	+			$\omega\theta^2$	$(1,0,\omega)$
θ^{13}	=	$(\omega+1)$	+	$\omega\theta$	+	$(\omega+1)\theta^2$	$(\omega+1,\omega,\omega+1)$
θ^{14}	=	1	+	$\omega\theta$	+	$(\omega+1)\theta^2$	$(1,\omega,\omega+1)$
θ^{15}	=	1	+			$(\omega+1)\theta^2$	$(1,0,\omega+1)$
θ^{16}	=	1	+			$\omega\theta^2$	$(1,0,\omega)$
θ^{17}	=	ω	+	$(\omega+1)\theta$	+	$\omega\theta^2$	$(\omega,\omega+1,\omega)$
θ^{18}	=	$(\omega+1)$	+	θ			$(\omega+1,1,0)$
θ^{19}	=			$(\omega+1)\theta$	+	θ^2	$(0,\omega+1,1)$
θ^{20}	=	ω	+	$\omega\theta$	+	θ^2	$(\omega,\omega,1)$
θ^{21}	=	ω					$(\omega,0,0)$

Таблица 2

Операции в кольце $\mathrm{GF}(2)[y]/f(y)$

$+$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Таблица 3

Представление степеней θ

θ^0	100
θ^1	010
θ^2	001
θ^3	111
θ^4	122
θ^5	132
θ^6	130

θ^7	013
θ^8	110
θ^9	011
θ^{10}	112
θ^{11}	133
θ^{12}	102
θ^{13}	131

θ^{14}	123
θ^{15}	103
θ^{16}	101
θ^{17}	121
θ^{18}	120
θ^{19}	012
θ^{20}	113

Отметим, что элементы множества $\{0,1,6,8,18\}$ образуют $(21,5,1)$-разностное множество и дают $(21,5,1)$-блок-схему с блоками

$$B_i = \{i, i+1, i+6, i+8, i+18\},\ \ i = 0,1,\ldots,20,$$

являющимися гиперплоскостями геометрии $PG(2,4)$.

Приведём теперь матрицу кодирования A-кода с правилом кодирования (13) (табл. 4). В матрице первый столбец соответствует номеру $\varkappa$ правила кодирования, а первая строка — сообщению m. Например, $e_{122}(s_3) = 102$.

Таблица 4

Матрица кодирования 1

	100	010	001	111	122	132	130	013	110	011	112	133	102	131	123	103	101	121	120	012	113
100	s_0	s_1					s_2		s_3										s_4		
010		s_0	s_1					s_2		s_3										s_4	
001			s_0	s_1					s_2		s_3										s_4
111	s_4			s_0	s_1					s_2		s_3									
122		s_4			s_0	s_1					s_2		s_3								
132			s_4			s_0	s_1					s_2		s_3							
130				s_4			s_0	s_1					s_2		s_3						
013					s_4			s_0	s_1					s_2		s_3					
110						s_4			s_0	s_1					s_2		s_3				
011							s_4			s_0	s_1					s_2		s_3			
112								s_4			s_0	s_1					s_2		s_3		
133									s_4			s_0	s_1					s_2		s_3	
102										s_4			s_0	s_1					s_2		s_3
131	s_3										s_4			s_0	s_1					s_2	
123		s_3										s_4			s_0	s_1					s_2
103	s_2		s_3										s_4			s_0	s_1				
101		s_2		s_3										s_4			s_0	s_1			
121			s_2		s_3										s_4			s_0	s_1		
120				s_2		s_3										s_4			s_0	s_1	
012					s_2		s_3										s_4			s_0	s_1
113	s_1					s_2		s_3										s_4			s_0

5. Обсуждение

Подчеркнём, что при построении полученной матрицы важен порядок следования состояний источника в строках (он соответствует разложению матрицы инцидентности в сумму подстановочных матриц). Перестановки элементов множества S в строках матрицы кодирования могут привести к тому, что получится код аутентификации без секретности. Такой A-код на основе проективной плоскости изучается в работе [22]. Этот A-код без секретности и построенный здесь A-код с секретностью «диаметрально удалены» друг от друга с точки зрения криптографической стойкости. «Между ними» расположены многие другие A-коды с секретностью. Приведём один пример (табл. 5).

Пусть $S = \{(1, a)\colon a \in \mathrm{GF}(q)\} \cup \{(0, 1)\}$ и правило кодирования задаётся формулой

$$e_{(\varkappa_2,\varkappa_1,\varkappa_0)}(s_1, s_0) = \begin{cases} (0, s_1, s_0), & \text{если } \varkappa = (1, 0, 0), \\ (s_1, -\varkappa_2\varkappa_1^{-1}s_1, s_0), & \text{если } \varkappa = (\varkappa_2, \varkappa_1, 0),\ \varkappa_1 \neq 0, \\ (s_1, s_0, -\varkappa_0^{-1}(\varkappa_2 s_1 + \varkappa_1 s_0)), & \text{если } \varkappa_0 \neq 0. \end{cases} \qquad (14)$$

Операции в (14) производятся в поле $\mathrm{GF}(q)$. Непосредственно из (14) получаем следующее утверждение.

Теорема 4. Сообщение $m = (m_2, m_1, m_0)$ однозначно определяет состояние источника $s = (s_1, s_0)$ тогда и только тогда, когда $m \in \{(0, 0, 1), (1, a, a),\ a \in \mathrm{GF}(q)\}$. Если при этом $m = (0, 0, 1)$, то $s = (0, 1)$, а если $m = (1, a, a)$, то $s = (1, a)$. Сообщение

Таблица 5

Матрица кодирования 2

	001	010	011	012	013	100	101	102	103	110	111	112	113	120	121	122	123	130	131	132	133
011		01				10				11				12				13			
010	01					10	11	12	13												
011			01			10					11					12					13
012					01	10							11		12					13	
013				01		10						11					12		13		
100	01	10	11	12	13																
101		01					10				11				12				13		
102		01							10				11				12				13
103		01						10				11				12				13	
110	01									10	11	12	13								
111			01				10			11							12			13	
112					01				10	11						12			13		
113				01				10		11					12						13
120	01																	10	11	12	13
121				01			10						11			12		13			
122			01						10			11			12			13			
123					01			10			11						12	13			
130	01													10	11	12	13				
131					01		10					11		12							13
132				01					10		11			12						13	
133			01					10					11	12					13		

$m = (0, 1, a)$ в одном случае определяет $s = (1, a)$ и в q случаях — $s = (0, 1)$; сообщение $m = (1, a, b)$ в одном случае определяет $s = (1, b)$ и в q случаях — $s = (1, a)$.

Мы построили A-код, позволяющий злоумышленнику однозначно определить состояние источника для $q + 1$ сообщений, а для q^2 сообщений — по два состояния источника, причём одно из них с вероятностью $\dfrac{q}{q+1}$, а другое — с вероятностью $\dfrac{1}{q+1}$. Матрица кодирования такого A-кода при $q = 4$ приведена в табл. 5.

Следует отметить, что конструкция кода аутентификации с секретностью на основе проективной плоскости изучается (в числе других конструкций) в [17], где указывается возможность построения совершенного шифра (частный случай теоремы 2), но не получена явная формула для соответствующего правила кодирования (аналог формулы (13)).

В заключение отметим, что основной сложностью в практическом использовании предлагаемого A-кода является нахождение явного выражения степеней θ^i, что связано с выбором примитивных многочленов $f(y)$ и $F(x)$. Вместе с тем эффективно вычисляемое правило кодирования (14) оставляет надежду на то, что неким подобным образом может быть построен A-код, допускающий совершенное или почти совершенное [4] шифрование.

ЛИТЕРАТУРА

1. *Зубов А. Ю.* К теоретико-игровому подходу исследования кодов аутентификации // Дискретная математика. 2009. Т. 21. Вып. 3. С. 45–72.
2. *Зубов А. Ю.* О выборе оптимальной стратегии для кода аутентификации с двумя состояниями источника // Дискретная математика. 2009. Т. 21. Вып. 4. С. 135–147.
3. *Зубов А. Ю.* Математика кодов аутентификации. М.: Гелиос АРВ, 2007.
4. *Зубов А. Ю.* Почти совершенные шифры и коды аутентификации // Прикладная дискретная математика. 2011. № 4(14). С. 28–33.
5. *Casse L. R. A., Martin K. M., and Wild P. R.* Bound and characterizations of authentication / secrecy schemes // Designs, Codes and Cryptography. 1998. V. 13. P. 107–129.
6. *Ding C. and Tian X.* Three constructions of authentication codes with perfect secrecy // Designs, Codes and Cryptography. 2004. V. 33. P. 227–239.
7. *Huber M.* Authentication and secrecy codes for equiprobable source probability distributions // `arxiv.org/abs/0904.0109`, 2009.
8. *Huber M.* Constructing optimal authentication codes with perfect multi-fold secrecy // Proc. IEEE Int. Zurich Seminar on Communications (IZS), March 3–5, 2010. Zurich, 2010. P. 86–89.
9. *Mitchell C. J., Piper C., Walker M., and Wild P.* Authentication schemes, perfect local randomizers, perfect secrecy and secret sharing schemes // Designs, Codes and Cryptography. 1996. V. 7. P. 101–110.
10. *Rees R. and Stinson D. R.* Combinatorial characterizations of authentication codes II // Designs, Codes and Cryptography. 1996. V. 7. P. 239–259.
11. *Sgarro A.* An introduction to the theory of unconditional secrecy and authentication // Geometries, Codes and Cryptography. CISM Courses and Lectures. No. 313. Springer Verlag, 1990. P. 131–160.
12. *Saygi Z.* Constructions of authentication codes. A thesis submitted to the Graduate School of Applied Mathematics of the METU. Ankara, 2007. 74 p.
13. *Smeets B., Vanrose P., and Wan C.-X.* On the constraction of authentication codes with secrecy and codes withstanding spoofing attack of order L // Eurocrypt'90. LNCS. 1990. V. 473. P. 307–312.
14. *De Soete M.* Some constructions for authentication — secrecy codes // Eurocrypt'88. LNCS. 1988. V. 330. P. 57–75.
15. *De Soete M.* Authentication/secrecy codes. Geometries, Codes and Cryptography // CISM Courses and Lectures. No. 313. Springer Verlag, 1990. P. 187–199.
16. *Stinson D. R.* A construction for authentication secrecy codes from certain combinatorial designs // Crypto'87. LNCS. 1988. V. 293. P. 355–366.
17. *Stinson D. R.* The combinatorics of authentication and secrecy codes // J. Cryptology. 1990. No. 2. P. 23–49.
18. *Van Tran T.* On the construction of authentication and secrecy codes // Designs, Codes and Cryptography. 1995. V. 5. P. 269–280.
19. *Холл М.* Комбинаторика. М.: ИЛ, 1970.
20. *Зубов А. Ю.* Криптографические методы защиты информации. Совершенные шифры. М.: Гелиос АРВ, 2005.
21. *Сачков В. Н.* Введение в комбинаторные методы дискретной математики. М.: МЦНМО, 2004.
22. *Gilbert E., MacWilliams F. J., and Sloane J. A.* Codes which detect deception // Bell System Tech. J. 1974. V. 53. P. 405–424.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/20/6

УДК 004.032.26(06)

ОТОБРАЖЕНИЕ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ НА МНОГОЯДЕРНЫЕ КОМПЬЮТЕРЫ РЕКУРРЕНТНЫМИ НЕЙРОННЫМИ СЕТЯМИ

М. С. Тарков

Институт физики полупроводников им. А. В. Ржанова СО РАН, г. Новосибирск, Россия

E-mail: tarkov@isp.nsc.ru

Рассмотрена задача отображения графа параллельной программы со взвешенными вершинами (процессами) и рёбрами (межпроцессными обменами) на произвольный взвешенный граф распределённой вычислительной системы. Предложен алгоритм решения этой задачи, основанный на использовании сетей Хопфилда. Алгоритм протестирован на отображении ряда графов параллельных программ на многоядерный компьютер. Эксперименты показали, что алгоритм позволяет получить хорошо сбалансированные субоптимальные отображения.

Ключевые слова: *графы параллельных программ, многоядерные системы, выравнивание нагрузки процессоров, нейрон, сети Хопфилда.*

Введение

В связи с постоянным увеличением мощности компьютеров появилась возможность создавать на их основе высокопроизводительные мультикомпьютерные вычислительные системы (ВС) [1 – 3]. Такая система в общем случае представляет собой объединённое линиями связи множество вычислительных узлов, которые обладают высокой степенью обособленности, т. е. каждый узел имеет свой процессор, зачастую многоядерный, свою память, возможно, свой собственный жёсткий диск и свой способ взаимодействия с другими узлами (сетевая карта, модем и т. п.). Узлы могут иметь разную производительность и разные коммуникационные возможности. В общем случае структура вычислительной сети произвольна и зависит от нескольких факторов, таких, как, например, аппаратные возможности, цель, для которой создаётся вычислительная сеть, финансовые ограничения.

В отличие от систем с общей памятью, в мультикомпьютерных ВС более остро стоит вопрос об уменьшении объёма межмашинных взаимодействий, так как пропускная способность сети связи низка. Необходимо таким образом распределить процессы по доступным вычислительным узлам, чтобы минимизировать время выполнения параллельной программы. Для этого нужно загрузить эти процессоры с учетом их производительности (чем выше производительность, тем выше нагрузка) и минимизировать взаимодействия между процессами, находящимися на разных узлах, что позволяет уменьшить простои процессоров, снижающие эффективность использования ВС.

В общем случае эта задача сводится к вложению графа параллельной программы в граф вычислительной системы [2, 3]. Целью вложения является минимизация

времени выполнения программы. Ввиду сложности задачи (она NP-полна) широко используются разнообразные эвристики для поиска оптимального вложения. В настоящее время популярностью пользуются методы, основанные на аналогиях с физикой и биологией, например такие, как метод имитации отжига, генетические алгоритмы и нейронные сети [4]. К последним относятся сети Хопфилда [5, 6].

1. Задача отображения

Цель оптимального распределения процессов (ветвей) параллельной программы по процессорам ВС заключается в минимизации времени выполнения программы, что эквивалентно минимизации времени простоя каждого процессора системы, участвующего в решении задачи, и одновременно минимизации затрат на взаимодействия между процессами, размещёнными в разных процессорах.

Пусть:

$G_p(V_p, E_p)$ — граф параллельной программы;

V_p — множество ветвей программы, $|V_p| = n$;

E_p — множество логических (виртуальных) каналов, реализующих взаимодействия между ветвями;

$G_s(V_s, E_s)$ — граф вычислительной системы;

V_s — множество процессоров (ядер), $|V_s| = m \leqslant n$;

E_s — множество связей между элементарными машинами (ЭМ);

w_x — вес (вычислительная сложность) ветви $x \in V_p$;

ϑ_i — производительность процессора $i \in V_s$;

$\tau_{xi} = w_x/\vartheta_i$ — время выполнения ветви $x \in V_p$ на процессоре $i \in V_s$;

c_{xy} — вес ребра $(x, y) \in E_p$, равный числу единиц информации, которыми обмениваются ветви x и y;

d_{ij} — время передачи единицы информации между процессорами (ядрами) i и j.

Пусть $f_m : G_p \to G_s$ — вложение графа программы G_p в граф вычислительной системы G_s. Качество вложения будем оценивать целевой функцией

$$H_g(f_m) = H_{gc}(f_m) + H_{gt}(f_m),$$

где $H_{gc}(f_m)$ — оценка небаланса вычислительной нагрузки; $H_{gt}(f_m)$ — оценка полного времени межпроцессорных взаимодействий.

Для вложения f_m полное время вычислений i-й ЭМ равно

$$t_i = \sum_{f_m(x)=i} \tau_{xi}.$$

Отсюда

$$H_{gc}(f_m) = \sum_{i=1}^{m} (t_i - t_{\min})^2,$$

где $t_{\min} = \sum\limits_x w_x / \sum\limits_i \vartheta_i$ — минимально возможное (идеальное) время выполнения параллельной программы (время выполнения программы на одном процессоре с производительностью, равной суммарной производительности всех ЭМ системы, при нулевых затратах на взаимодействия между ветвями программы).

Затраты на взаимодействия оцениваются функцией

$$H_{gt}(f_m) = \sum_{\substack{x \neq y, \\ i=f_m(x), j=f_m(y)}} c_{xy} d_{ij},$$

где суммирование производится по всем парам (x, y) взаимодействующих ветвей параллельной программы.

2. Сеть Хопфилда для задачи отображения

Рассмотрим матрицу нейронов v размером $n \times m$, каждая строка которой соответствует ветви параллельной программы, каждый столбец — процессору (ядру). Каждая строка матрицы v должна содержать один и только один ненулевой элемент, равный единице, остальные элементы должны быть равны нулю (ветвь параллельной программы не может быть отображена одновременно на несколько процессоров). Каждый столбец матрицы v может содержать произвольное (в том числе и нулевое) число элементов, равных единице, но суммарное количество единичных элементов должно быть равно числу ветвей n параллельной программы.

Энергия соответствующей нейронной сети Хопфилда описывается функцией Ляпунова

$$H = \frac{A}{2} H_c + \frac{B}{2} H_g. \tag{1}$$

Здесь A и B — параметры функции Ляпунова. Минимум H_c обеспечивает выполнение вышеуказанных ограничений на элементы матрицы v; H_g — целевая функция;

$$H_c = H_{c_1} + H_{c_2}, \quad H_{c_1} = \left(\sum_x \sum_i v_{xi} - n \right)^2, \quad H_{c_2} = \sum_x \left(\sum_i v_{xi} - 1 \right)^2; \tag{2}$$

минимум H_{c_1} обеспечивает наличие в матрице v ровно n единиц; минимум H_{c_2} обеспечивает наличие в каждой строке матрицы v ровно одной единицы;

$$H_g = H_{gc} + H_{gt}, \quad H_{gc} = \sum_i \left(\sum_x v_{xi} \tau_{xi} - t_{\min} \right)^2, \quad H_{gt} = \sum_x \sum_i v_{xi} \sum_y \sum_j v_{yj} c_{xy} d_{ij}. \tag{3}$$

Здесь v_{xi} — состояние нейрона строки x и столбца i матрицы v.

Динамика сети Хопфилда, минимизирующей функцию (1), описывается системой уравнений

$$\frac{\partial u_{xi}}{\partial t} = -\frac{\partial E}{\partial v_{xi}}, \tag{4}$$

где u_{xi} — активация нейрона с индексами $x = 1, \ldots, n$, $i = 1, \ldots, m$; $v_{xi} = (1 + {} + \exp(-\beta u_{xi}))^{-1}$ — состояние (выходной сигнал) нейрона, β — параметр функции активации. Из (1)–(4) получаем

$$\begin{aligned} \frac{\partial u_{xi}}{\partial t} = &-A \left(\sum_y \sum_j v_{yj} + \sum_j v_{xj} + \sum_y v_{yi} - n - 1 \right) - \\ &-B \left(\left(\sum_y v_{yi} \tau_{yi} - t_{\min} \right) \tau_{xi} + \sum_y \sum_j v_{yj} c_{xy} d_{ij} \right). \end{aligned} \tag{5}$$

Соответствующее (5) разностное уравнение имеет вид (t — номер текущей итерации, Δt — величина временного шага)

$$\begin{aligned} u_{xi}^{t+1} = &u_{xi}^t - \Delta t \left(A \left(\sum_y \sum_j v_{yj} + \sum\nolimits_j v_{xj} + \sum_y v_{yi} - n - 1 \right) + \right. \\ &\left. + B \left(\left(\sum_y v_{yi} \tau_{yi} - t_{\min} \right) \tau_{xi} + \sum_y \sum_j v_{yj} c_{xy} d_{ij} \right) \right). \end{aligned} \tag{6}$$

С целью ускорения сходимости сеть Хопфилда (6) преобразуется в сеть Вана [7] умножением целевой функции оптимизационной задачи на $\exp(-t/\tau)$, где τ — параметр:

$$u_{xi}^{t+1} = u_{xi}^{t} - \Delta t \left(A \left(\sum_y \sum_j v_{yj} + \sum_j v_{xj} + \sum_y v_{yi} - n - 1 \right) + \right.$$
$$\left. + B \left(\left(\sum_y v_{yi}\tau_{yi} - t_{\min})\tau_{xi} + \sum_y \sum_j v_{yj} c_{xy} d_{ij} \right) \exp(-t/\tau \right) \right). \tag{7}$$

Новое значение v_{xi}^{t+1} вычисляется сразу же после вычисления соответствующего значения u_{xi}^{t+1} (метод Гаусса — Зейделя).

3. Эксперименты

В экспериментах исследовано отображение на компьютер с $m = 2, 4, 8$ ядрами следующих вариантов параллельных программ с одинаковыми весами w_x, $x = 1, \ldots, n$, вершин (ветвей) и весами $c_{xy} = 0$ или $c_{xy} = 1$, $x, y = 1, \ldots, n$, $x \neq y$, рёбер графа программы:

1) множество независимых ветвей (отсутствуют обмены между ветвями, $c_{xy} = 0$);
2) типовые графы параллельных программ (линейка, кольцо и решётка) (рис. 1), $c_{xy} = 1$;
3) нерегулярные сетки, $c_{xy} = 1$.

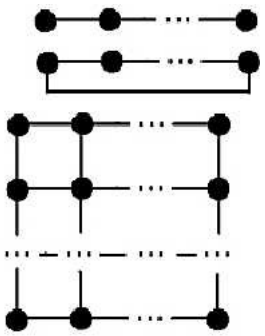


Рис. 1. Типовые графы параллельных программ (линейка, кольцо, решётка)

Изменяя параметры d_{ij} при фиксированной производительности ядер ϑ_i, мы изменяем связность подмножеств вершин, отображённых на одно и то же ядро, т. е. оптимальность затрат на межъядерные обмены информацией.

Параметры вычислительной системы: производительность ядер $\vartheta_i = 1$, $i = 1, \ldots, m$;

$$d_{ij} = \begin{cases} 0, & i = j, \\ 1, & i \neq j. \end{cases} \tag{8}$$

Согласно (8), затраты на обмены данными между ветвями программы внутри ядра считаются пренебрежимо малыми по отношению к межъядерным обменам. Иначе говоря, данные в ядре рассматриваются как массивы, обрабатываемые одной ветвью.

При $w_x = 1$ и $\vartheta_i = 1$ имеем $\tau_{xi} = 1$, $x = 1, \ldots, n$, $i = 1, \ldots, m$. Параметры нейронной сети: $A = 1000$; $B = 100$; $\Delta t = 1$; $\beta = 1$; $\tau = 100$.

Процедура отображения параллельной программы на вычислительную систему имеет следующий вид:

```
do
{initialize();
  do
  {iterate(); iter = iter+1;
   δ = nobalance();
  } while (δ > 0 && iter < maxiter);
} while (δ > maxnb || noncorrect());
```

Здесь:

— `initialize` задает начальные значения элементов матриц u^0 (и соответственно v^0), используя генератор псевдослучайных чисел;

— `iterate` выполняет шаг итерационного процесса (7), вычисляя новые значения элементов матриц u^{t+1} и v^{t+1};

— `nobalance` вычисляет небаланс нагрузки по формуле

$$\delta = \frac{\sqrt{\sum_{i=1}^{m}(t_i - t_{\min})^2}}{m \cdot t_{\min}};$$

— `noncorrect` проверяет выполнение условий (2) корректности решения v^{t+1}.

Итерационный процесс (7) продолжается до тех пор, пока не будет достигнут баланс нагрузки ($\delta = 0$) или не будет выполнено максимально допустимое количество итераций `maxiter`. Если по указанным условиям итерационный процесс завершён, проверяется корректность полученного решения v^{t+1}. Если решение некорректно или небаланс нагрузки превышает допустимый максимум `maxnb`, то задаются новые начальные условия и итерационный процесс повторяется (производится рестарт алгоритма).

В табл. 1–4 приведены следующие результаты серий из 100 испытаний алгоритма при отображении графов программ на компьютер с четырьмя ядрами ($m = 4$), полученные на процессоре Pentium (R) Dual-Core CPU E 52000, 2,5 ГГц:

I_a — среднее число итераций по 100 испытаниям;

I_m — максимальное число итераций;

t_a — среднее время выполнения алгоритма отображения (в секундах);

t_m — максимальное время выполнения алгоритма (в секундах);

N_a — среднее количество рестартов алгоритма при невыполнении условия `(δ > maxnb || noncorrect())`;

N_m — максимальное количество рестартов алгоритма при невыполнении условия `(δ > maxnb || noncorrect())`;

C_a — средний суммарный объём данных, передаваемых между ядрами компьютера для вычисленного отображения;

C_m — максимальный суммарный объём данных, передаваемых между ядрами компьютера для вычисленного отображения.

Отметим, что при указанных выше весах c_{xy} величины C_a и C_m задают количество рёбер графа программы $G_p(V_p, E_p)$, соединяющих вершины этого графа, отображённые на разные ядра компьютера.

Таблица 1

Независимые задания

n	I_a	I_m	t_a	t_m	N_a	N_m
4	61	374	0,0084	0,109	0,51	6
8	50	265	0,026	0,235	0,59	5
16	50	275	0,05	0,579	0,41	7
32	57	356	0,066	0,781	0,28	3
64	70	222	0,441	2,92	0,82	5

Таблица 2

Граф программы — линейка

n	I_a	I_m	t_a	t_m	N_a	N_m	C_a	C_m
4	75	312	0,0027	0,032	0,19	2	3	3
8	68	423	0,003	0,062	0,05	2	4,23	7
16	70	515	0,014	0,156	0,17	3	5,13	10
32	83	285	0,046	0,454	0,21	2	7,24	13
64	112	529	0,334	2,985	0,47	4	11,17	17

Таблица 3

Граф программы — кольцо

n	I_a	I_m	t_a	t_m	N_a	N_m	C_a	C_m
4	110	348	0,0061	0,047	0,37	3	4	4
8	71	243	0,0044	0,047	0,12	2	4,86	7
16	59	213	0,0078	0,141	0,07	2	5,53	10
32	74	273	0,055	0,796	0,26	2	7,82	15
64	97	216	0,336	5,93	0,43	5	11,68	19

Таблица 4

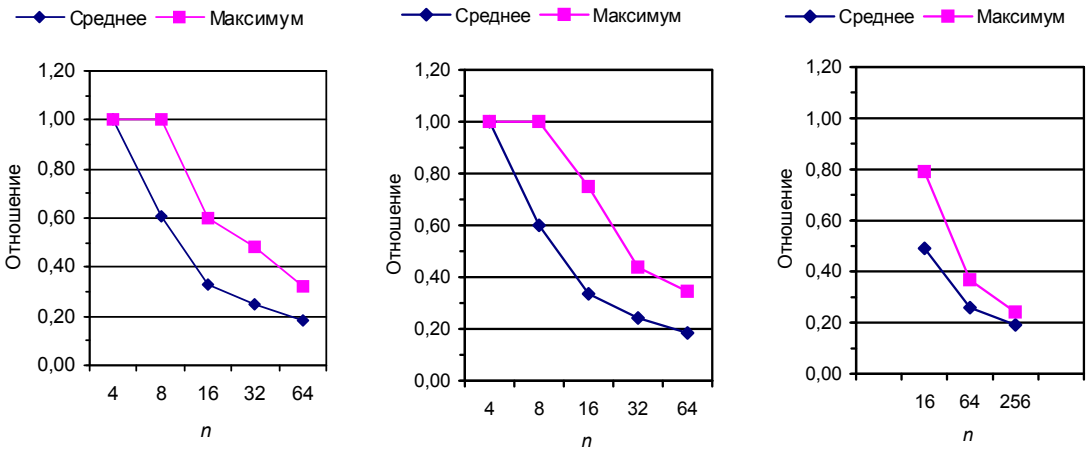
Граф программы — решётка

n	I_a	I_m	t_a	t_m	N_a	N_m	C_a	C_m
4	104	353	0,0064	0,047	0,54	3	4	4
16	60	166	0,0097	0,032	0,24	2	11,41	17
64	89	306	0,283	1,77	0,62	4	29,82	40
256	571	1000	15,8	99,7	1,28	15	92,32	123

Для всех случаев, указанных в табл. 1–4, получены полностью сбалансированные отображения ($\delta = 0$), за исключением отображения решётки с числом вершин $256 = 16 \times 16$, где $0 < \delta < 0{,}01$. Рассматривались решетки с $n = k \times k$ вершинами, $k = 4, 8, 16$ (при $k = 2$ решётка вырождается в кольцо).

На рис. 2 приведены графики отношений $r_a = C_a/C_{\max}$ и $r_m = C_m/C_{\max}$, где $C_{\max}$ — число рёбер графа программы, для линейки ($C_{\max} = n - 1$), кольца ($C_{\max} = n$) и квадратной решётки ($C_{\max} = \sqrt{n}(\sqrt{n} - 1)$) соответственно. Графики показывают, что несмотря на рост объёма информации, передаваемой между ядрами системы, с увеличением числа вершин графа программы относительная доля рёбер графа, приходящихся на межъядерные взаимодействия, существенно уменьшается.

На рис. 3–5 приведены примеры отображений решётки (рис. 3) и нерегулярных сеток (рис. 4 и 5). Одинаковыми знаками помечены вершины графа программы, отображённые на одно и то же ядро. Рисунки свидетельствуют о субоптимальности полученных отображений.



а *б* *в*

Рис. 2. Отношения r_a (среднее) и r_m (максимум): *а* — линейка; *б* — кольцо; *в* — рёшетка

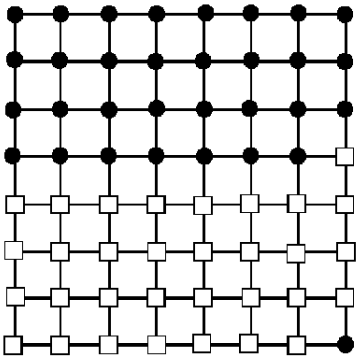


Рис. 3. Пример отображения решётки 8×8 вершин на систему из двух ядер

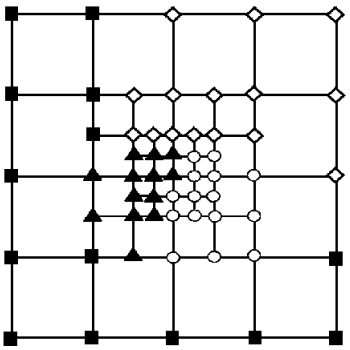


Рис. 4. Пример отображения прямоугольной нерегулярной сетки на систему из четырёх ядер ($n = 57$)

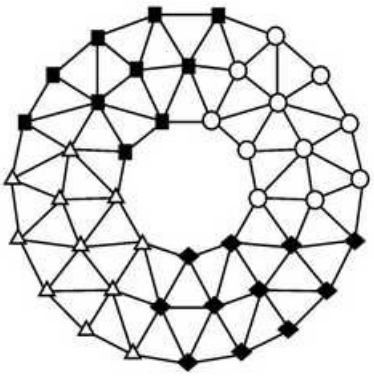


Рис. 5. Пример отображения триангуляционной нерегулярной сетки на систему из четырёх ядер ($n = 43$)

Эксперименты показали, что предложенный алгоритм отображения, основанный на использовании сети Вана [7, 8], позволяет получить для компьютера с числом ядер $m = 2, 4, 8$:

1) полный баланс вычислительной нагрузки (нагрузка распределяется равномерно) для типовых графов параллельных программ (пустой граф, линейка, решётка, кольцо) с числом вершин $n \in \{2, 4, 8, 16, 32, 64\}$;
2) субоптимальные отображения для нерегулярных сеток с несколькими десятками вершин (небаланс нагрузки не превышает 5 %);
3) существенное снижение доли рёбер графа программы, приходящихся на межъядерные обмены, при увеличении числа вершин графа программы.

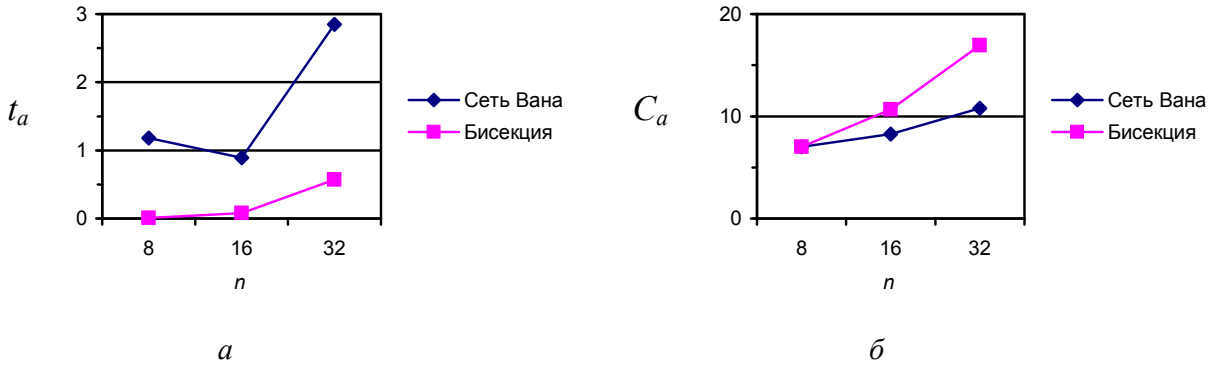
Табл. 5 показывает ($n = 32$), что при числе ядер $m = 2, 4$ времена работы алгоритма отображения t_a и t_m и количества рестартов алгоритма малы, но при увеличении числа ядер до $m = 8$ эти величины резко возрастают.

Таблица 5

Граф программы — линейка ($n = 32$)

m	I_a	I_m	t_a	t_m	N_a	N_m	C_a	C_m
2	14	81	0,0026	0,031	0,58	6	5,69	13
4	83	285	0,046	0,454	0,21	2	7,24	13
8	177	375	2,85	27,39	0,58	23	10,79	18

Проведены эксперименты по отображению графов программ для числа ядер $m = 8$ рекурсивной бисекцией, когда граф программы разбивается на два подграфа, которые, в свою очередь, рекурсивно разбиваются на подграфы вплоть до заданного числа ядер. Каждый шаг рекурсивной бисекции реализуется сетью Вана. Установлено, что рекурсивная бисекция существенно сокращает время отображения (рис. 6, *а*) при равномерном распределении вычислительной нагрузки, но затраты C_a и C_m на межъядерные обмены информацией в полученном отображении при этом возрастают (рис. 6, *б*).



а *б*

Рис. 6. Сравнение отображений линейки на компьютер с числом ядер сетью Вана и рекурсивной бисекцией: *а* — по среднему времени отображения; *б* — по средним затратам на межъядерные обмены в полученном отображении

Случай большого числа вершин (сотни и тысячи) можно свести к рассмотренным выше, используя методику Карписа — Кумара [8] огрубления графа программы.

Заключение

Рассмотрена задача отображения графа параллельной программы со взвешенными вершинами (процессами) и рёбрами (межпроцессорными обменами) на произвольный взвешенный граф распределённой вычислительной системы. Предложен алгоритм решения этой задачи, основанный на использовании сетей Хопфилда. Алгоритм

протестирован на частном случае рассмотренной задачи — отображении ряда графов параллельных программ (однородное множество независимых заданий — пустой граф, линейка, кольцо, решётка, неоднородная решетка) на многоядерный компьютер. Эксперименты показали, что для графов параллельных программ с несколькими десятками вершин предложенный алгоритм позволяет получить хорошо сбалансированные субоптимальные отображения: величина относительного небаланса нагрузки на ядро не превышает 5 %; подграф графа программы, отображённый на отдельное ядро, имеет близкое к минимуму количество внешних рёбер, соответствующих межъядерным обменам.

ЛИТЕРАТУРА

1. *Корнеев В. В.* Параллельные вычислительные системы. М.: Нолидж, 1999. 320 с.
2. *Bokhari S. H.* On the mapping problem // IEEE Trans. Comp. 1981. V. C-30. No. 3. P. 207–214.
3. *Тарков М. С.* Вложение структур параллельных программ в структуры живучих распределенных вычислительных систем // Автометрия. 2003. Т. 39. № 3. С. 84–96.
4. *Осовский С.* Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.
5. *Меламед И. И.* Нейронные сети и комбинаторная оптимизация // Автоматика и телемеханика. 1994. № 4. С. 3–40.
6. *Smith K. A.* Neural networks for combinatorial optimization: a review of more than a decade of research// INFORMS J. Computing. 1999. V. 11. No. 1. P. 15–34.
7. *Hung D. L. and Wang J.* Digital hardware realization of a recurrent neural network for solving the assignment problem // Neurocomputing. 2003. V. 51. P. 447–461.
8. *Karypis G. and Kumar V.* Multilevel k-way partitioning scheme for irregular graphs // J. Parallel and Distributed Computing. 1998. V. 48. P. 96–129.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/20/7

УДК 631.391:519.2

БЫСТРЫЙ АЛГОРИТМ ВОССТАНОВЛЕНИЯ ИСТИННОГО РЕШЕНИЯ ФИКСИРОВАННОГО ВЕСА СИСТЕМЫ ЛИНЕЙНЫХ БУЛЕВЫХ УРАВНЕНИЙ С ИСКАЖЁННОЙ ПРАВОЙ ЧАСТЬЮ

А. Н. Алексейчук, А. Ю. Грязнухин

Институт специальной связи и защиты информации Национального технического университета Украины «Киевский политехнический институт», г. Киев, Украина

E-mail: alex-crypto@mail.ru, x67mail@gmail.com

Рассматривается система линейных булевых уравнений с искажённой правой частью, истинное решение которой имеет заданный вес, не зависящий от числа неизвестных в системе. Предлагается вероятностный алгоритм нахождения этого решения, имеющий меньшую временную сложность по сравнению с методом максимума правдоподобия. В отличие от известных алгоритмов, обладающих указанным свойством, предложенный алгоритм использует только операции сравнения и (поразрядного и арифметического) сложения двоичных целых чисел, что позволяет применять его на практике в случае, когда другие алгоритмы оказываются менее эффективными.

Ключевые слова: *система булевых уравнений с искаженной правой частью, вероятностный алгоритм.*

Введение

Рассмотрим систему булевых уравнений

$$Ax = b = Ax_0 \oplus \xi, \tag{1}$$

где A — случайная равновероятная матрица размера $m \times k$ над полем из двух элементов; x_0 — неизвестный k-мерный вектор; $\xi = (\xi_1, \ldots, \xi_m)^{\mathrm{T}}$ — случайный вектор с независимыми координатами, распределёнными по закону

$$\Pr\{\xi_i = 1\} = 1 - \Pr\{\xi_i = 0\} = p,\ i = 1, 2, \ldots, m,\ p \in (0, 1/2). \tag{2}$$

Система уравнений (СУ) (1) называется *системой линейных булевых уравнений с искажённой правой частью*, а вектор x_0 — её *истинным решением* [1].

Предположим, что вес (количество ненулевых координат) вектора x_0 равен фиксированному числу $\rho \geqslant 3$. Требуется восстановить этот вектор по известным значениям A, b, p и ρ.

Как правило, данную задачу решают в предположении, что число неизвестных k в системе (1) принимает любые достаточно большие натуральные значения, а вероятность искажения p не зависит от k. При этом требуется разработать алгоритм, позволяющий для любых наперёд заданных $p \in (0, 1/2)$ и $\rho \in \{3, 4, \ldots\}$ находить истинное

решение этой системы уравнений с любой наперёд заданной надёжностью при всех достаточно больших значениях k и m [1–3].

Назовём алгоритм A восстановления истинных решений СУ (1) *состоятельным*, если существует функция $m_0 : \mathbb{N}\times(0,1/2)\times(0,1/2)\times\{3,4,\ldots\} \longrightarrow \mathbb{N}$, удовлетворяющая следующему условию: для любых $p, \delta \in (0,1/2)$, $\rho \in \{3,4,\ldots\}$ существует число $k_0 \in \mathbb{N}$, такое, что для любого натурального $k \geqslant k_0$ и произвольного вектора $x_0 \in \{0,1\}^k$ веса ρ вероятность правильного восстановления этого вектора с помощью алгоритма A по известной реализации случайной системы (1), состоящей из $m = m_0(k,p,\delta,\rho)$ уравнений, больше либо равна $1-\delta$.

В дальнейшем рассматриваются только состоятельные алгоритмы. Трудоёмкость (или временная сложность в худшем случае) каждого из них является функцией четырёх параметров: $T = T(k,p,\delta,\rho)$, где $m = m_0(k,p,\delta,\rho)$; $k \geqslant k_0$; $p,\delta \in (0,1/2)$; $\rho \in \{3,4,\ldots\}$. Обычно интересуются асимптотическим поведением этой функции при $k \to \infty$ для каждого набора значений (p,δ,ρ). Наибольший интерес представляют алгоритмы, трудоёмкость которых полиномиально зависит от k, $(1-2p)^{-1}$ и δ^{-1}.

Уточним понятие элементарной операции, необходимое для оценки трудоёмкости алгоритмов. Отметим, что в большинстве работ, посвящённых алгоритмам решения систем булевых уравнений с искажёнными правыми частями, тип используемых операций не оговаривается явно. Как правило, это операции над целыми числами или двоичными векторами произвольной длины. В настоящей работе *элементарной* называется любая двоичная операция (булева функция двух переменных), а также операция $i \mapsto i+1$, где i — произвольное целое число. Любые (не обязательно элементарные) операции называются *условными*.

Остановимся подробнее на известных алгоритмах восстановления истинных решений систем уравнений (1). Наиболее естественный из них состоит в применении метода максимума правдоподобия и сводится к перебору всех k-мерных векторов веса ρ. Известно [1, теорема 5.1], что для любых $p \in (0,1/2)$, $\rho \in \{3,4,\ldots\}$ и $m = (1+\theta_k)\log\binom{k}{\rho}/(1-h(p))$, где $h(p) = -p\log p - (1-p)\log(1-p)$, $\lim\limits_{k\to\infty}\theta_k > 0$, метод максимума правдоподобия позволяет восстанавливать истинное решение СУ (1) с вероятностью, стремящейся к 1 при $k \to \infty$. Нетрудно убедиться, что в этом случае трудоёмкость метода составляет $\mathrm{O}((1-2p)^{-2}k^{\rho}\log k)$ элементарных операций, где O зависит только от ρ.

В [2] предложен асимптотически более эффективный алгоритм, основная идея которого заключается в том, чтобы построить таблицы D и D', состоящие из векторов Ay и $Az \oplus b$ соответственно, где y и z пробегают все k-мерные булевы векторы веса $\lfloor\rho/2\rfloor$ и $\lceil\rho/2\rceil$ соответственно, и применить к этим таблицам алгоритм решения задачи о ближайшей окрестности (approximate nearest neighbor problem), описанный в [4]. Последний позволяет с некоторой высокой вероятностью найти в таблицах векторы Ay_0 и $Az_0 \oplus b$, расположенные на определённом (не слишком большом) расстоянии Хэмминга. В [2] показано, что для любой последовательности положительных чисел $\omega_1, \omega_2, \ldots$, сходящейся к $+\infty$, и произвольных p, $\delta \in (0,1/2)$, $\rho \in \{3,4,\ldots\}$ при всех достаточно больших k и $m = \lceil\rho(1-2p)^{-2}\log(k\delta^{-1})\omega_k\rceil$ истинное решение СУ (1) равно $y_0 \oplus z_0$ с вероятностью не менее $1-\delta$. При этом для нахождения этого решения требуется $\mathrm{O}((1-2p)^{-2}k^{\lceil\rho/2\rceil(1+4p^2+\alpha_k(p))}\log(k\delta^{-1})\omega_k)$ условных операций, где $\lim\limits_{k\to\infty}\alpha_k(p) = 0$ для любого $p \in (0,1/2)$, а O зависит только от ρ (отметим, что в формулировках этого результата, приведённых в [2] (см. теорему 5 и следствие 1), имеются неточности и опечатки).

Обратим внимание, что в [4] отсутствуют явные оценки надёжности алгоритма решения задачи о ближайшей окрестности, а в [2] ошибочно предполагается, что этот алгоритм всегда завершается успешно. Поэтому вопрос о том, с какой вероятностью алгоритм из [2] восстановит вектор x_0, выполняя указанное выше количество операций, остаётся открытым.

Отметим также, что алгоритм из [4] предназначен для решения задачи о ближайшей окрестности в евклидовом пространстве, а не в пространстве Хэмминга. В результате (при стандартном вложении второго пространства в первое) основная доля вычислений приходится на операции над вещественными числами. Последнее обстоятельство существенно усложняет программную реализацию алгоритма, который становится труднопримени́мым на практике, если число неизвестных k заключено в пределах от нескольких сотен до нескольких тысяч, а вероятность p является величиной порядка 10^{-3}. (Отметим, что к решению систем уравнений с такими параметрами приводит задача восстановления некоторых линейных кодов по наборам случайных кодовых слов, искажённых в двоичном симметричном канале связи [5].)

В [3] предложен другой алгоритм восстановления вектора x_0, основанный на быстром умножении некоторых целочисленных матриц, трудоёмкость которого для любых $p \in (0, 1/2)$, $\rho \in \{3, 4, \ldots\}$ и всех достаточно больших k, m (зависящих от p и ρ) ограничена сверху величиной $poly((1-2p)^{-1})k^{0,8\rho}$ условных операций, где $poly$ — некоторый полином, не зависящий от параметров. Этот алгоритм интересен тем, что в оценке его трудоёмкости отсутствует зависимость от p в показателе степени числа k. Однако при указанных выше значениях k и p алгоритм из [3] практически не превосходит по эффективности метод максимума правдоподобия. При этом для его применения требуется существенно больший объём памяти.

Целью настоящей работы является разработка алгоритма, имеющего меньшую трудоёмкость по сравнению с методом максимума правдоподобия, допускающего простую программную реализацию и позволяющего на практике восстанавливать истинные решения систем уравнений (1) от нескольких сотен или тысяч неизвестных, по крайней мере, при малых значениях p и ρ. Представленный ниже алгоритм базируется на той же идее, что и алгоритм из [2], но использует только операции сравнения и (поразрядного и арифметического) сложения двоичных целых чисел. Вместо решения задачи о ближайшей окрестности предложено проводить быстрый поиск пары близких векторов в таблицах D и D' по критерию совпадения их фрагментов в случайно выбранном множестве координат. Отметим, что эта идея используется в [6–8] и ряде других работ по теории информационного поиска. Получены оценки надёжности и трудоёмкости предложенного алгоритма. Приведены результаты вычислительных экспериментов, показывающие, что средняя трудоёмкость алгоритма заметно меньше её верхней границы в худшем случае.

1. Теоретические результаты

В дальнейшем, если не оговорено противное, символ O обозначает некоторую положительную постоянную, не зависящую от каких-либо параметров. В частности, для любых функций φ и ψ, определённых на некотором множестве X и принимающих вещественные значения, формула $\varphi(x) = \mathrm{O}(\psi(x))$, $x \in X$, означает, что существует такое положительное число C, не зависящее от x, что $|\varphi(x)| \leqslant C|\psi(x)|$ для всех $x \in X$ (см. [9, с. 12]). Если при этом множество X не указано явно, то по умолчанию оно совпадает с пересечением областей определения функций φ и ψ.

Введём следующие обозначения: V_k — множество двоичных векторов длины k; $\|x\|$ — вес вектора $x \in V_k$; $\mathrm{d}(x, y)$ — расстояние Хэмминга между векторами $x, y \in V_k$; $\overline{a, b} = \{u \in \mathbb{Z} : a \leqslant u \leqslant b\}, a, b \in \mathbb{Z}$. Положим

$$\rho_1 = \lfloor \rho/2 \rfloor, \rho_2 = \lceil \rho/2 \rceil, N = \binom{k}{\rho_1}, \widetilde{N} = \binom{k}{\rho_2},$$
$$V^{(1)} = \{y \in V_k : \|y\| = \rho_1\}, V^{(2)} = \{z \in V_k : \|z\| = \rho_2\}.$$

Для любого множества $M \subseteq \overline{1, m}$ и произвольной матрицы U, строки которой занумерованы числами $1, 2, \ldots, m$, обозначим U_M подматрицу матрицы U, содержащуюся в её строках с номерами из M.

Предлагаемый *алгоритм восстановления вектора x_0 по указанным выше исходным данным* A, b, p, ρ зависит от параметров $l \in \mathbb{N}$, $\varepsilon \in (p, 1/2)$, $t \in \mathbb{N}$, где $t < (1-\varepsilon)m$, и имеет следующий вид.

Положить Out = «НЕТ».

Для каждого $i \in \overline{1, l}$ выполнить следующие действия.

Процедура 1. Сгенерировать случайное равновероятное t-множество $\mathcal{M}_i \subseteq \overline{1, m}$; построить таблицу D_i, состоящую из упорядоченных пар $(y, A_{\mathcal{M}_i} y)$, где $y \in V^{(1)}$; с помощью одного из известных быстрых алгоритмов сортировки построить новую таблицу $D_i{}^*$, расположив пары $(y, A_{\mathcal{M}_i} y)$ в порядке неубывания t-разрядных целых чисел, соответствующих векторам $A_{\mathcal{M}_i} y$, $y \in V^{(1)}$.

Процедура 2. Для каждого $z \in V^{(2)}$

— положить $b(z) = Az \oplus b$;

— используя алгоритм бинарного поиска, проверить по таблице $D_i{}^*$, существует ли вектор $y \in V^{(1)}$, такой, что $A_{\mathcal{M}_i} y = b(z)_{\mathcal{M}_i}$; если да, то

  — найти указанный вектор y;

  — если $\mathrm{d}(Ay, b(z)) \leqslant m\varepsilon$, то положить Out $= y \oplus z$ и закончить работу.

Прокомментируем описанный алгоритм. Он состоит из l шагов, на каждом из которых выполняются процедуры 1 и 2. Результатом работы алгоритма является значение Out, равное некоторому k-мерному двоичному вектору, рассматриваемому в качестве оценки истинного решения СУ (1), либо слову «НЕТ», если эту СУ решить не удалось (хотя бы даже ошибочно).

Процедура 1 базируется на идее, предложенной в [2], и аналогична этапу предвычислений в атаках «баланс время — данные — память» [10, 11]. Множества $\mathcal{M}_i$, $i \in \overline{1, l}$, следует выбирать независимо друг от друга, случайно и равновероятно из совокупности всех подмножеств мощности t множества $\overline{1, m}$. Для их генерации можно использовать алгоритм, описанный в [12, с. 212]. Для сортировки таблицы D_i следует применять быстрые алгоритмы, трудоёмкость которых составляет $\mathrm{O}(N \log N)$ операций сравнения в худшем случае [13].

Процедура 2 состоит в поиске векторов $y \in V^{(1)}$ и $z \in V^{(2)}$, для которых расстояние Хэмминга между векторами Ay и $b(z)$ не превосходит $m\varepsilon$. Поиск ведётся до первого успеха: в случае нахождения указанных векторов алгоритм завершает работу. Для ускорения поиска используется следующий приём [6–8]: рассмотрим фрагмент $b(z)_{\mathcal{M}_i}$ вектора $b(z)$ и проверим, не встречается ли он среди вторых компонент пар векторов, записанных в таблице D_i. Алгоритм бинарного поиска (см., например, [12, с. 236]) выполняет эту проверку за время $\mathrm{O}(\log N)$ вместо $\mathrm{O}(N)$ операций сравнения, если таблица D_i отсортирована по второй компоненте. Отметим также, что алгоритм бинарного

поиска находит ровно один вектор $y \in V^{(1)}$ со свойством $A_{\mathcal{M}_i} y = b(z)_{\mathcal{M}_i}$, если такой существует. Поэтому описанный алгоритм решения СУ (1) может совершить ошибку в случае, когда вектор $b(z)_{\mathcal{M}_i}$ встречается более одного раза в наборе $(A_{\mathcal{M}_i} y : y \in V^{(1)})$, $i \in \overline{1,l}$.

Обозначим $P_{\text{ош}}$ вероятность ошибки описанного алгоритма (подчеркнём, что она определяется относительно совместного распределения независимых случайных элементов A, ξ, $\mathcal{M}_1, \ldots, \mathcal{M}_l$, распределённых по указанным выше законам).

Теорема 1. Для любых $k, l, m, t, \rho \in \mathbb{N}$, $p \in (0, 1/2)$ и $\varepsilon \in (p, 1/2)$, таких, что $3 \leqslant \rho \leqslant k-1$, $t < (1-\varepsilon)m$, описанный алгоритм выполняет

$$T = \mathrm{O}(l\widetilde{N}(t \log N + \rho m)) \tag{3}$$

элементарных операций и использует память размера

$$S = \mathrm{O}(tN \log N + km) \ \text{ бит.} \tag{4}$$

При этом справедливо неравенство

$$P_{\text{ош}} \leqslant 2^{-t} N + \exp\{-2m(\varepsilon - p)^2\} + \widetilde{N}^2 \exp\{-2m(1/2-\varepsilon)^2\} + \exp\left\{-l(1-\varepsilon-tm^{-1})^t\right\}. \tag{5}$$

Доказательство. На i-м шаге алгоритма ($i \in \overline{1,l}$) для построения таблицы D_i достаточно выполнить не более $Nt\rho_1$ элементарных операций, а для формирования таблицы D_i^* — $\mathrm{O}(Nt \log N)$ операций. Кроме того, для каждого $z \in V^{(2)}$ вычисление вектора $b(z)$, поиск в таблице D_i^* и (возможная) проверка условия $\mathrm{d}(Ay, b(z)) \leqslant m\varepsilon$ потребуют не более $t\rho_2$, $\mathrm{O}(t \log N)$ и $\mathrm{O}(\rho m)$ элементарных операций соответственно. Следовательно, $T = \mathrm{O}(l(Nt\rho_1 + Nt \log N + \widetilde{N}(t\rho_2 + t \log N + \rho m)))$, откуда вытекает справедливость формулы (3).

Далее, для хранения матрицы A и вектора b требуется $\mathrm{O}(km)$ бит памяти, а для хранения таблиц D_i и D_i^* — $\mathrm{O}(tN \log N)$ бит (включая дополнительную память, необходимую для быстрой сортировки). Следовательно, выполняется равенство (4).

Убедимся в справедливости формулы (5). Обозначим

$$\mathcal{M} = (\mathcal{M}_1, ..., \mathcal{M}_i), \mathcal{N}(\xi) = \{i \in \overline{1,l} : \xi_i = 0\}, S(x_0) = \{(y,z) \in V^{(1)} \times V^{(2)} : y \oplus z \neq x_0\}$$

и рассмотрим следующие события:

$$\Omega_1 = \bigcup_{(y,z) \in S(x_0)} \{d(Ay, Az \oplus b) \leqslant m\varepsilon\}, \Omega_2 = \{\|\xi\| \geqslant m\varepsilon\}, \Omega_3 = \bigcap_{i=1}^{l} \{\mathcal{M}_i \nsubseteq \mathcal{N}(\xi)\} \backslash \Omega_2,$$

$$\Omega_4 = \{\text{Out} = \text{«НЕТ»}\} \backslash (\Omega_1 \cup \Omega_2 \cup \Omega_3).$$

Заметим, что если алгоритм совершает ошибку, то либо наступает событие Ω_1, либо выполняется равенство Out = «НЕТ». Следовательно,

$$P_{\text{ош}} \leqslant \Pr(\Omega_1) + \Pr(\Omega_2) + \Pr(\Omega_3) + \Pr(\Omega_4). \tag{6}$$

Оценим вероятность события Ω_1. Поскольку A — случайная равновероятная матрица, а вектор ξ не зависит от A, то в силу равенства (1) для любого $(y,z) \in S(x_0)$ случайная величина $\mathrm{d}(Ay, Az \oplus b) = \|A(y \oplus z \oplus x_0) \oplus \xi\|$ распределена по закону Бернулли с параметрами $(m, 1/2)$. Отсюда на основании неравенства для вероятностей больших уклонений (см., например, [14, с. 31]) следует, что

$$\Pr\{\mathrm{d}(Ay, Az \oplus b) \leqslant m\varepsilon\} \leqslant \exp\{-2m(1/2 - \varepsilon)^2\}$$

и, значит,

$$\Pr(\Omega_1) \leqslant |V^{(1)}| \cdot |V^{(2)}| \exp\{-2m(1/2-\varepsilon)^2\} \leqslant \widetilde{N}^2 \exp\{-2m(1/2-\varepsilon)^2\}. \tag{7}$$

Аналогично, используя неравенство для вероятностей больших уклонений, получим, что

$$\Pr(\Omega_2) \leqslant \exp\{-2m(\varepsilon-p)^2\}. \tag{8}$$

Далее, в силу независимости случайных равновероятных t-множеств $\mathcal{M}_1, \ldots, \mathcal{M}_l$ и вектора ξ, а также неравенства $t \leqslant (1-\varepsilon)m$ вероятность события $\Omega_3 = \bigcap\limits_{i=l}^{l}\{\mathcal{M}_i \nsubseteq \nsubseteq \mathcal{N}(\xi)\} \cap \{|\mathcal{N}(\xi)| > (1-\varepsilon)m\}$ не превосходит величины

$$N = \left(1 - \frac{\binom{\lceil (1-\varepsilon)m \rceil}{t}}{\binom{m}{t}}\right)^l \leqslant \left(1 - \frac{((1-\varepsilon)m)_t}{(m)_t}\right)^l,$$

где $(u)_t = u(u-1)\ldots(u-t+1)$ для любого $u \geqslant t$. Используя неравенства $(u-t)^t \leqslant (u)_t \leqslant u^t$, $u > t \geqslant 1$, получим отсюда, что

$$\begin{aligned}\Pr(\Omega_3) &\leqslant \left(1 - \left(\frac{(1-\varepsilon)m - t}{m}\right)^t\right)^l \leqslant \exp\left\{-l\left(\frac{(1-\varepsilon)m-t}{m}\right)^t\right\} = \\ &= \exp\left\{-l(1-\varepsilon-tm^{-1})^t\right\}.\end{aligned} \tag{9}$$

Покажем, наконец, что

$$\Pr(\Omega_4) \leqslant 2^{-t}N. \tag{10}$$

Зафиксируем векторы $y_0 \in V^{(1)}$, $z_0 \in V^{(2)}$, такие, что $x_0 = y_0 \oplus z_0$. Обозначим $U_i(\xi, \mathcal{M})$ событие, состоящее в том, что i — наименьшее натуральное число от 1 до l со свойством $\mathcal{M}_i \subseteq \mathcal{N}(\xi)$; $V(A, \mathcal{M}_i)$ — событие, состоящее в том, что случайный вектор $A_{\mathcal{M}_i}y_0$ встречается в наборе $(A_{\mathcal{M}_i}y : y \in V^{(1)})$ не менее двух раз. Справедливо равенство

$$\Pr(\Omega_4) = \sum_{i=1}^{l} \Pr(\Omega_4 \cap U_i(\xi, \mathcal{M})). \tag{11}$$

Пусть происходит событие $\Omega_4 \cap U_i(\xi, \mathcal{M})$. Тогда $\xi_{\mathcal{M}_i} = 0$ и вектор $A_{\mathcal{M}_i}z_0 \oplus b_{\mathcal{M}_i} = A_{\mathcal{M}_i}y_0$ присутствует среди вторых компонент пар, записанных в таблице D_i. С другой стороны, поскольку алгоритм возвращает значение Out = «НЕТ», на i-м шаге в результате выполнения процедуры 2 определяется вектор $y \in V^{(1)}$, отличный от y_0. Следовательно, происходит событие $V(A, \mathcal{M}_i)$. Итак,

$$\Pr(\Omega_4 \cap U_i(\xi, \mathcal{M})) \leqslant \Pr(\Omega_4 \cap V(A, \mathcal{M}_i)), i \in \overline{1,l}. \tag{12}$$

Далее, для любого фиксированного t-множества $M \subseteq \overline{1,m}$ вероятность события $V(A, M) = V(A, \mathcal{M}_i) \cap \{\mathcal{M}_i = M\}$ не превосходит среднего числа появлений нулевого вектора в наборе $(A_M(y \oplus y_0) : y \in V^{(1)} \backslash \{y_0\})$, которое равно $2^{-t}(N-1)$. Отсюда на

основании формул (11), (12) и независимости случайных элементов A, ξ и $\mathcal{M}$ следует, что

$$\Pr(\Omega_4) \leqslant \sum_{i=l}^{l} \sum_{\substack{M \subseteq \overline{1,m}: \\ |M|=t}} \Pr(U_i(\xi, \mathcal{M}) \cap V(A, \mathcal{M}_i) \cap \{\mathcal{M}_i = M\}) =$$

$$= \sum_{i=l}^{l} \sum_{\substack{M \subseteq \overline{1,m}: \\ |M|=t}} \Pr(U_i(\xi, \mathcal{M}) \cap \{\mathcal{M}_i = M\}) \Pr(V(A, M)) \leqslant$$

$$\leqslant 2^{-t} N \sum_{i=l}^{l} \sum_{\substack{M \subseteq \overline{1,m}: \\ |M|=t}} \Pr(U_i(\xi, \mathcal{M}) \cap \{\mathcal{M}_i = M\}) = 2^{-t} N \sum_{i=l}^{l} \Pr(U_i(\xi, \mathcal{M})) \leqslant 2^{-t} N.$$

Итак, справедливо неравенство (10), что и требовалось доказать.

Из оценок (6)–(10) непосредственно следует неравенство (5). ■

Исследуем асимптотическое поведение трудоёмкости алгоритма при $k \to \infty$ для каждого набора значений (p, δ, ρ).

Теорема 2. Пусть δ, δ_1, δ_2, δ_3, δ_4 — положительные числа, такие, что

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 \leqslant \delta < 1/2. \tag{13}$$

Для любых $p \in (0, 1/2)$, $\rho \in \{3, 4, \ldots\}$ и $k > \rho$ положим

$$t = \max\left\{\left\lceil \log(N{\delta_1}^{-1}) \right\rceil, \lfloor (1/2 - p)^{-1} \rfloor + 1\right\}, \varepsilon = p + t^{-1}; \tag{14}$$

$$m = \max\left\{\left\lceil 1/2 \cdot t^2 \ln({\delta_2}^{-1}) \right\rceil, \lceil 1/2 \cdot (1/2 - \varepsilon)^{-2} \ln(\widetilde{N}^2 {\delta_3}^{-1}) \rceil, \lfloor t(1-\varepsilon)^{-1} \rfloor + 1\right\}; \tag{15}$$

$$l = \left\lceil \left(1 - \varepsilon - tm^{-1}\right)^{-t} \ln({\delta_4}^{-1}) \right\rceil. \tag{16}$$

Тогда предложенный алгоритм восстанавливает истинное решение СУ (1) с вероятностью не менее $1 - \delta$, а его трудоёмкость при $k \to \infty$ составляет

$$T = \mathrm{O}\left((1-2p)^{-2} k^{\lceil \rho/2 \rceil (1 + \log\left(\frac{1}{1-p}\right))} \log^2 k^{\lceil \rho/2 \rceil}\right) \tag{17}$$

элементарных операций, где O зависит только от δ_j ($j \in \overline{1,4}$).

Доказательство. Из соотношений (14), (15) следует, что параметры ε и t удовлетворяют ограничениям, указанным в условии теоремы 1. Неравенство $P_{\text{ош}} \leqslant \delta$ вытекает непосредственно из формул (5) и (13)–(16).

Убедимся в справедливости равенства (17) (на протяжении оставшейся части доказательства символ O обозначает некоторую постоянную, зависящую разве что от параметров $\delta_j, j \in \overline{1,4}$).

Обозначим $\mu = \varepsilon + tm^{-1}$, $C = 1 + 2/\ln({\delta_2}^{-1})$ и зафиксируем число $k_0 \in \mathbb{N}$, такое, что $1/2 \cdot \log k_0 > \max\{C, 1 + (1/2 - p)^{-1}\}$. Из определения параметров N и t вытекает, что для любого $k \geqslant k_0$ справедливы следующие соотношения:

$$2(1 + (1/2 - p)^{-1}) < \log N \leqslant t = \lceil \log(N{\delta_1}^{-1}) \rceil \leqslant \log(2N{\delta_1}^{-1}). \tag{18}$$

Кроме того, согласно формуле (15), $tm^{-1} \leqslant 2/(t \ln({\delta_2}^{-1}))$, откуда следует, что для любого $k \geqslant k_0$

$$\mu = p + t^{-1} + tm^{-1} \leqslant p + t^{-1}\left(1 + \frac{2}{\ln({\delta_2}^{-1})}\right) \leqslant p + \frac{C}{\log N}.$$

Оценим сверху значение $(1-\mu)^{-t}$ при $k \geqslant k_0$. Заметим, что поскольку $p < 1/2$ и $1/2 \cdot \log N > C$, то $1-\mu \geqslant 1-p-C/\log N \geqslant (1-p)\,(1-2C/\log N) > 0$. Отсюда, используя верхнюю оценку параметра t из формулы (18), получим

$$(1-\mu)^{-t} \leqslant (1-p)^{-t}\left(1-\frac{2C}{\log N}\right)^{-t} = 2^{t\log(1-p)^{-1}}2^{t\log(1-2C/\log N)^{-1}} \leqslant$$
$$\leqslant (2N{\delta_1}^{-1})^{\log(1-p)^{-1}}\left(2N{\delta_1}^{-1}\right)^{\log(1-2C/\log N)^{-1}} \leqslant$$
$$\leqslant N^{\log(1-p)^{-1}}(2{\delta_1}^{-1})(2N{\delta_1}^{-1})^{\log(1-2C/\log N)^{-1}}.$$

Наконец, применяя известное неравенство $\ln(1-x) \geqslant -x(1-x)^{-1}$, $x \in (0,1)$, получим

$$\ln\left((2N{\delta_1}^{-1})^{\log(1-2C/\log N)^{-1}}\right) = \ln\left(\left(1-\frac{2C}{\log N}\right)^{-\log(2N{\delta_1}^{-1})}\right) \leqslant$$
$$\leqslant \log(2N{\delta_1}^{-1})\frac{2C}{\log N}\left(1-\frac{2C}{\log N}\right)^{-1} = \mathrm{O}(1), \quad k \to \infty.$$

Итак,

$$l = \mathrm{O}\left((1-\mu)^{-t}\right) = \mathrm{O}\left(N^{\log(1-p)^{-1}}\right), \quad k \to \infty. \tag{19}$$

Далее, на основании соотношений (18) для любого $k \geqslant k_0$ справедливы неравенства $t(1/2-p) > 2(1+(1/2-p)^{-1})(1/2-p) > 2$, из которых следует, что

$$1/2-\varepsilon = 1/2-p-t^{-1} = (1/2-p)(1-t^{-1}(1/2-p)^{-1}) > 1/2\cdot(1/2-p).$$

Отсюда на основании формулы (15) получаем

$$m = \mathrm{O}((1/2-p)^{-2}\log^2\widetilde{N}), \quad k \to \infty. \tag{20}$$

Кроме того, согласно формуле (14),

$$t = \log N + O(1), k \to \infty. \tag{21}$$

Подставляя выражения в правых частях равенств (19)–(21) в формулу (3) и принимая во внимание соотношения $N \leqslant \widetilde{N} \leqslant k^{\lceil\rho/2\rceil}$, $\rho\widetilde{N} \leqslant 1/2\cdot k^{\lceil\rho/2\rceil}$, $\rho \in \overline{3,k-1}$, получаем

$$T = \mathrm{O}(l\widetilde{N}(t\log N + \rho m)) = \mathrm{O}\left(\rho\widetilde{N}^{1+\log(1-p)^{-1}}(1/2-p)^{-2}\log^2\widetilde{N}\right) =$$
$$= \mathrm{O}\left((1-2p)^{-2}k^{\lceil\rho/2\rceil\left(1+\log(1-p)^{-1}\right)}\log^2 k^{\lceil\rho/2\rceil}\right), \quad k \to \infty.$$

Итак, равенство (17), а вместе с ним и теорема полностью доказаны. ■

Замечание 1. Как видно из доказательства теоремы 2, при выполнении равенств (14)–(16) и $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 0{,}25\delta$ трудоёмкость описанного алгоритма ограничена сверху полиномиальной функцией от δ^{-1} — величины, обратной к верхней границе вероятности ошибки алгоритма. Вопрос о том, обладают ли указанным свойством алгоритмы, изложенные в [2, 3], в этих публикациях не обсуждается.

Замечание 2. Трудоёмкость описанного алгоритма можно уменьшить, если при выполнении процедуры 1 хранить таблицу D_i в «распределённом» виде, записывая по адресу $A_{\mathcal{M}_i}y$ номер вектора $y \in V^{(1)}$ (в случае многократного обращения по одному и тому же адресу сохраняется последняя запись). При выполнении процедуры 2 для нахождения вектора $y \in V^{(1)}$ со свойством $A_{\mathcal{M}_i}y = b(z)_{\mathcal{M}_i}$ достаточно обратиться в память по адресу $b(z)_{\mathcal{M}_i}$, $z \in V^{(2)}$, $i \in \overline{1,l}$. Нетрудно видеть, что вероятность ошибки модифицированного таким образом алгоритма оценивается сверху по формуле (5). При этом его трудоёмкость составляет

$$\widetilde{T} = \mathrm{O}(\widetilde{N}l\rho m) \tag{22}$$

элементарных операций, а объём используемой памяти —

$$\widetilde{S} = \mathrm{O}(2^t t + km) \ \text{ бит.} \tag{23}$$

2. Вычислительные эксперименты

Описанный алгоритм (в модифицированном виде; см. замечание 2) реализован программно и применён к решению ряда систем уравнений с искажёнными правыми частями. Вычислительные эксперименты проводились на ЭВМ с процессором Intel Core i5-2450 (2,5 ГГц) и объёмом оперативной памяти 6 Гб RAM (DDR3) на базе Windows 7. При этом использовалась среда разработки Microsoft Visual C++ Studio 2008.

В табл. 1 и 2 показаны типичные результаты, полученные в двух сериях экспериментов. Слева в таблицах приведены теоретические значения параметров t, ε, l и m, $\log\widetilde{T}$, $\log\widetilde{S}$, рассчитанные при $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 0{,}25\delta$ по формулам (14)–(16) и (22), (23). Справа указаны значения параметров t', ε', m', выбранные для проведения экспериментов, а также результаты, полученные в ходе их выполнения.

Таблица 1

Сравнение теоретических и экспериментальных результатов
($k = 200$, $\rho = 4$, $\delta = 0{,}1$)

p	Теория						Эксперимент					
	t	ε	l	m	$\log\widetilde{T}$	$\log\widetilde{S}$	t'	ε'	$l'_{\mathrm{ср}}$	$l'_{\max}$	m'	$T_{\mathrm{ср}}$, с
0,005	20	0,055	21	738	30,20	24,33	20	0,055	1,15	3	600	0,0056
0,008	20	0,058	22	738	30,27	24,33	20	0,058	1,20	6	600	0,0082
0,010	20	0,060	23	738	30,33	24,33	20	0,060	1,27	6	600	0,0089
0,020	20	0,070	29	738	30,67	24,33	20	0,070	1,59	12	600	0,0215
0,050	20	0,100	57	738	31,64	24,33	20	0,100	2,66	20	600	0,0604

Для каждой пары значений (p, m'), приведённых в табл. 1 и 2, 225 раз выполнялась следующая процедура: независимо друг от друга, случайно равновероятно генерировались $m' \times k$ -матрица A и вектор x_0 веса ρ, по которым формировалась СУ вида (1) с искажениями в правой части, распределёнными по закону (2). Затем к полученной СУ применялся описанный выше (модифицированный) алгоритм с параметрами l, ε' и t', значения которых приведены в таблицах. В результате выполнения алгоритма для каждой из 225 систем уравнений определялось фактическое число шагов (выбираемых множеств $\mathcal{M}$, $i \in \overline{1,l}$) до нахождения оценки истинного решения этой системы.

Таблица 2

Сравнение теоретических и экспериментальных результатов
($k = 2000$, $\rho = 4$, $\delta = 0{,}1$)

p	Теория						Эксперимент					
	t	ε	l	m	$\log \widetilde{T}$	$\log \widetilde{S}$	t'	ε'	$l'_{\text{ср}}$	$l'_{\max}$	m'	$T_{\text{ср}}$, с
0,005	27	0,042	21	1345	37,72	31,76	26	0,042	1,17	4	1000	19,516
0,008	27	0,045	23	1345	37,85	31,76	26	0,045	1,20	4	1000	21,312
0,010	27	0,047	25	1345	37,97	31,76	26	0,047	1,39	7	1000	30,572
0,020	27	0,057	33	1345	38,39	31,76	26	0,057	1,56	5	1000	43,866
0,050	27	0,087	79	1345	39,63	31,76	26	0,087	4,15	31	1000	153,721

Параметры $l'_{\text{ср}}$ и $l'_{\max}$ в таблицах равны соответственно среднему арифметическому и наибольшему из этих чисел. В последней колонке табл. 1 и 2 приведено среднее время выполнения алгоритма в секундах (при этом надёжность восстановления истинных решений сгененированных СУ составляет 100 %).

Как видно из таблиц, значения $l'_{\text{ср}}$ и $l'_{\max}$ заметно меньше теоретической верхней границы l количества шагов, выполняемых до нахождения оценки истинного решения СУ. Например, согласно табл. 2, истинное решение веса 4 системы из 1000 уравнений от 2000 неизвестных с вероятностью искажения в правой части $p = 0{,}050$ находится в среднем не более чем на пятом (а в худшем случае — на тридцать первом) шаге со 100 %-й надёжностью, в то время как теоретическое значение числа шагов $l = 79$ (при большем количестве уравнений $m = 1345$ и нижней границе надёжности алгоритма 90 %). Среднее время восстановления решения такой СУ составляет около 2,5 мин.

Экспериментально установлено, что параметры алгоритма могут быть выбраны в широком диапазоне в зависимости от ограничений относительно трудоёмкости и количества уравнений в системе. При этом на практике значения указанных параметров заметно меньше теоретических (при той же нижней границе надёжности).

В табл. 3–5 приведены результаты сравнения предложенного алгоритма с методом максимума правдоподобия. Данные в табл. 3 получены следующим образом. Сначала экспериментальным путём для каждого значения p, указанного в таблице, были получены оценки $m_1(p)$ и $m_2(p)$ наименьшего числа уравнений в (случайно сгенерированной) системе (1), при котором вероятность правильного восстановления её истинного решения с помощью метода максимума правдоподобия и соответственно предложенного алгоритма составляет не менее 90 %. Затем для каждого p 225 раз выполнялась процедура, в ходе которой к случайно сгенерированной СУ (1), состоящей из $m_2(p)$ уравнений, применялся предложенный алгоритм. Потом из этой СУ удалялись последние $m_2(p) - m_1(p)$ уравнений и полученная система решалась методом максимума правдоподобия. Данные в табл. 4 и 5 получены аналогично, с тем отличием, что для каждого исходного значения p вместо 225 запусков процедуры выполнялось только 10. Последнее обстоятельство обусловлено заметным увеличением времени решения СУ методом максимума правдоподобия.

Таблица 3

Сравнение алгоритмов решения систем с искажёнными правыми частями ($k = 128$, $\rho = 4$)

p	Метод максимума правдоподобия		Предложенный алгоритм	
	Среднее время выполнения, с	Количество уравнений, $m_1(p)$	Среднее время выполнения, с	Количество уравнений, $m_2(p)$
0,005	3,6795	40	0,0008	56
0,008	3,6923	40	0,0027	56
0,010	3,8418	40	0,0072	56
0,020	3,9712	48	0,0113	160
0,050	5,8443	72	0,0453	200

Таблица 4

Сравнение алгоритмов решения систем с искажёнными правыми частями ($k = 128$, $\rho = 5$)

p	Метод максимума правдоподобия		Предложенный алгоритм	
	Среднее время выполнения, с	Количество уравнений, $m_1(p)$	Среднее время выполнения, с	Количество уравнений, $m_2(p)$
0,005	94,48	40	0,0732	56
0,008	99,73	40	0,0899	56
0,010	105,96	40	0,0995	56
0,020	106,16	48	0,3466	72
0,050	136,60	72	1,4910	160

Таблица 5

Сравнение алгоритмов решения систем с искажёнными правыми частями ($k = 128$, $\rho = 6$)

p	Метод максимума правдоподобия		Предложенный алгоритм	
	Среднее время выполнения, с	Количество уравнений, $m_1(p)$	Среднее время выполнения, с	Количество уравнений, $m_2(p)$
0,005	2056,94	40	0,4013	56
0,008	2154,86	40	0,4551	64
0,010	2138,73	40	0,4278	64
0,020	2197,63	48	0,7418	120
0,050	3691,22	72	2,7181	200

Как видно из таблиц, предложенный алгоритм в сотни (а в ряде случаев — в тысячи) раз превосходит по эффективности метод максимума правдоподобия, причём с ростом значения ρ это различие становится всё более заметным.

В целом, предложенный алгоритм представляется более простым и эффективным с точки зрения программной реализации по сравнению с алгоритмами, описанными в [2, 3], и может быть применён на практике для восстановления истинных решений малого веса слабоискажённых систем линейных булевых уравнений от нескольких сотен или тысяч неизвестных.

Авторы благодарны рецензенту за критические замечания, учёт которых позволил существенно улучшить качество работы, избежать неточностей в формулировках и усилить первоначальную версию теоремы 2.

ЛИТЕРАТУРА

1. *Балакин Г. В.* Введение в теорию случайных систем уравнений // Труды по дискретной математике. М.: ТВП, 1997. Т. 1. С. 1–18.
2. *Grigorescu E., Reysin L., and Vempala S.* On noise-tolerant learning of sparse parities and related problems // Proc. 22nd Intern. Conf. of Algorithmic Learning Theory. Berlin, Heidelberg: Springer Verlag, 2011. P. 413–424.
3. *Valiant G.* Finding correlations in subquadratic time, with applications to learning parities and juntas // Proc. 53rd Annual IEEE Symp. on the Foundations of Computer Science, New Brunswick, New Jersey, October 2–23, 2012. P. 11–20.
4. *Andoni A. and Indyk P.* Near-optimal hashing algorithms for approximate nearest neighbor in high dimension // Proc. 47th Symp. on Foundations of Computer Science, Berkeley, California, October 21–24, 2006. P. 459–468.
5. *Алексейчук А. Н., Грязнухин А. Ю.* Метод восстановления систематических линейных кодов по наборам искаженных кодовых слов // Прикладная радиоэлектроника. 2013. Т. 12. № 2. С. 128–132.
6. *Greene D., Parnas M., and Yao F.* Multi-index hashing for information retrieval // Proc. 35th Annual IEEE Symp. on the Foundations of Computer Science, Santa Fe, New Mexico, November 20–22, 1994. P. 722–731.
7. *Dolev D., Harari Y., and Parnas M.* Finding the neighborhood of a query in a dictionary // Proc. 2nd Israel Symp. on Theory and Computing Systems, Natanya, Israel, June 7–9, 1993. P. 33–42.
8. *Indyk P. and Motwani R.* Approximate nearest neighbors: towards removing the curse of dimensionality // Proc. Symp. on Theory of Computing. New York, NY, USA: ACM, 1998. P. 604–613.
9. *Де Брёйн Н. Г.* Асимптотические методы в анализе: пер. с англ. М.: ИЛ, 1961. 247 с.
10. *Babbage S. H.* Improved “exhaustive search” attacks on stream ciphers // European Convention on Security and Detection, Brighton, May 16–18, 1995. P. 161–166.
11. *Golić J.* Cryptanalysis of alleged A5 stream cipher // Advances in Cryptology. Proc. EUROCRYPT’97. Springer Verlag, 1997. P. 239–255.
12. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика: пер. с англ. М.: Мир, 1980. 476 с.
13. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов: пер. с англ. М.: Мир, 1979. 535 с.
14. *Ширяев А. Н.* Вероятность: учеб. пособие для вузов. М.: Наука, 1989. 640 с.

2013 Вычислительные методы в дискретной математике №2(20)

DOI 10.17223/20710410/20/8

УДК 519.1

ФОРМУЛА ДЛЯ ЧИСЛА СОЧЕТАНИЙ С ПОВТОРЕНИЯМИ ПРИ ОГРАНИЧЕНИЯХ И ЕЁ ПРИМЕНЕНИЕ

В. В. Гоцуленко

Институт технической теплофизики НАН Украины, г. Киев, Украина

E-mail: gosul@ukr.net

Получены различные обобщения понятия числа сочетаний с повторениями. Найдены формулы для вычисления введённых комбинаторных чисел и рассмотрены различные задачи, которые решаются с их применением.

Ключевые слова: *сочетания с повторениями при ограничениях, диофантовы уравнения, формальный полином, производящие функции, мультимножества.*

Введение

Всё разнообразие комбинаторных формул может быть фактически выведено из нескольких основных утверждений, касающихся конечных множеств [1, 2]. Тем не менее во многих отношениях полезно вводить различные комбинаторные числа, характеризующие те или иные количественные соотношения на множествах произвольной природы. Классические понятия числа сочетаний, размещений и перестановок в случае возможности дублирования элементов в выборках обобщаются в различных направлениях. Так, в частности, возникают сочетания, размещения и перестановки с повторениями [1]. Существует множество комбинаторных задач, которые можно эффективно решать, не обращаясь всякий раз к основным принципам комбинаторики, например к правилу произведения, а используя различные обобщения основных комбинаторных чисел.

Непосредственное прикладное применение комбинаторных чисел связано с подсчётом количества подмультимножеств заданного мультимножества [3], обладающих определёнными свойствами. При этом данная задача может рассматриваться и как универсальный способ построения и классификации комбинаторных чисел в зависимости от задаваемых свойств подмультимножеств.

Первичная спецификация мультимножества приводит к задаче подсчета m-элементных подмультимножеств заданного мультимножества с фиксированными весами его элементов. Например, в [4] получена формула для решения данной задачи. В [5] на основе данной формулы получено выражение, более удобное для проведения вычислений. В этой же работе получена формула для числа упорядоченных подмультимножеств. Другой подход для определения количества m-элементных подмультимножеств заданного мультимножества через их вторичные спецификации рассматривается в [6].

В данной работе получено несколько новых возможных способов обобщения понятия числа сочетаний с повторениями и рассмотрены некоторые комбинаторные задачи, решаемые с их помощью. В частности, полученные результаты имеют непосредственное применение к задаче о подсчёте количества подмультимножеств заданного мультимножества.

1. Определение числа сочетаний с повторениями при ограничениях

Формулировка достаточно общей комбинаторной задачи, приводящей к сочетаниям с повторениями, может быть следующей: имеется большое число предметов, например бесконечное, n различных типов. Необходимо определить, сколько из них можно составить различных m-элементных наборов, не принимая во внимание порядок элементов. В качестве точной формализации понятия k-элементного набора можно рассматривать n-компонентный вектор $(x_1, x_2, \ldots, x_n)$, где x_k означает количество элементов k-го типа, $k = 1, \ldots, n$, в соответствующем m-элементном наборе.

Следовательно, исходная задача эквивалентна задаче определения количества неотрицательных решений линейного диофантова уравнения

$$x_1 + x_2 + \ldots + x_n = m, \tag{1}$$

где $x_k \geqslant 0$, $k = 1, \ldots, n$. Для числа сочетаний с повторениями в литературе используются различные обозначения, например $\overline{C}_n^m$ или f_n^m; будем придерживаться последнего обозначения. Решение задачи (1) дается следующей формулой [1]:

$$f_n^m = C_{n-1+m}^{n-1},$$

где C_n^m — число сочетаний без повторений из n элементов по m элементов.

Если в исходной задаче количество элементов каждого типа не является бесконечным, а ограничено некоторым числом, для каждого типа своим, то приходим к понятию числа сочетаний с повторениями при ограничениях. Иными словами, рассмотрим следующую комбинаторную задачу. Имеется n ящиков с конфетами. Разрешается взять ровно m конфет из этих ящиков, причём из k-го ящика разрешается взять не более m_k конфет. Необходимо найти, сколькими способами это можно сделать. Формулировка данной задачи в терминах диофантовых уравнений следующая: необходимо найти количество неотрицательных решений линейного диофантова уравнения (1) при наличии ограничений

$$0 \leqslant x_k \leqslant m_k, \;\; k = 1, \ldots, n. \tag{2}$$

Обозначим данное число символом $f_n^m\,[m_1, m_2, \ldots, m_n]$ и будем его называть числом сочетаний с повторениями из n типов элементов по m элементов при (простых) ограничениях, определяемых неравенствами (2).

Замечание 1. Несложно проверить, что решение задачи (1), (2), т. е. формула для $f_n^m\,[m_1, m_2, \ldots, m_n]$, имеет смысл лишь при $m_k \geqslant 0$ для $k = 1, \ldots, n$ и $m_1 + m_2 + {} + \ldots + m_n \geqslant m$.

Задача (1), (2) также может быть сформулирована как задача подсчёта m-элементных подмультимножеств заданного мультимножества с весами элементов m_1, m_2, …, m_n. Применяя метод производящих функций, далее мы получим новую формулу для решения задачи (1), (2) и некоторых её обобщений.

Отметим, что конфеты в ящиках могут быть упакованы не насыпью, а находиться в коробках или блоках по r_j штук, $j = 1, \ldots, N$. Таким образом, имеется N типов коробок, где в коробке j-го типа находится r_j конфет.

Замечание 2. В отношении последней задачи возможны два варианта. В первом случае разрешается из каждого ящика брать не более одной коробки конфет. Решение задачи в этом случае будем называть числом сочетаний с повторениями при блочных ограничениях и обозначать символом $f_n^m \left[\dfrac{m_1, m_2, \ldots, m_n}{r_1, r_2, \ldots, r_N}\right]$. Во втором случае ограничений на количество коробок нет. Необходимо лишь, чтобы суммарное количество

конфет, взятых из k-го ящика, не превышало m_k, $k = 1, \ldots, n$. Решение второй задачи будем называть числом сочетаний второго типа с повторениями при блочных ограничениях и обозначать символом $F_n^m \left[\dfrac{m_1, m_2, \ldots, m_n}{r_1, r_2, \ldots, r_N} \right]$.

Интерпретация формулы $f_n^m \left[\dfrac{m_1, m_2, \ldots, m_n}{r_1, r_2, \ldots, r_N} \right]$ в терминах числа решений диофантова уравнения (1) приводит к следующим ограничениям для допустимых решений:

$$0 \leqslant x_k \leqslant m_k,\ x_k \in \{r_1, r_2, \ldots, r_N\},\ k = 1, \ldots, n,$$

где r_j, $j = 1, \ldots, N$, — произвольные целые неотрицательные числа, удовлетворяющие условию $\min\limits_{1 \leqslant j \leqslant N} \{r_j\} \leqslant \min\limits_{1 \leqslant k \leqslant n} \{m_k\}$.

Введём в рассмотрение следующие множества целых чисел ($k = 1, \ldots, n$):

$$I(m_k; r_1, r_2, \ldots, r_N) = \left\{ r : r = \sum_{p=1}^{N} \alpha_{kp} r_p \leqslant m_k,\ \alpha_{kp} \in \mathbb{Z}^+ = \mathbb{N} \cup \{0\},\ p = 1, \ldots, N \right\}.$$

Тогда для определения комбинаторного числа $F_n^m \left[\dfrac{m_1, m_2, \ldots, m_n}{r_1, r_2, \ldots, r_N} \right]$ через целочисленные решения уравнения (1) приходим к следующим ограничениям:

$$0 \leqslant x_k \leqslant m_k,\ x_k \in I(m_k; r_1, r_2, \ldots, r_N),\ k = 1, \ldots, n.$$

2. Формулы для числа сочетаний с повторениями при ограничениях

Для вывода формул числа сочетаний с повторениями при ограничениях воспользуемся следующей конструкцией метода производящих функций [7]. Рассмотрим формальный полином

$$P(t) = \left(1 + t + t^2 + \ldots + t^{m_1}\right)\left(1 + t + t^2 + \ldots + t^{m_2}\right) \ldots \left(1 + t + t^2 + \ldots + t^{m_n}\right) = \\ = \sum_{\substack{j=0, x_1 + x_2 + \ldots + x_n = j, \\ 0 \leqslant x_k \leqslant m_k, k = 1, \ldots, n}}^{m_1 + m_2 + \ldots + m_n} t^{x_1} t^{x_2} \ldots t^{x_n}. \tag{3}$$

Представим $P(t)$ в стандартной форме по возрастающим показателям степеней:

$$P(t) = 1 + p_1 t + p_2 t^2 + \ldots + p_R t^R,\ R = m_1 + m_2 + \ldots + m_n.$$

Тогда из тождества (3) следует, что $f_n^m [m_1, m_2, \ldots, m_n] = p_m$. Для вычисления данного коэффициента положим в тождестве (3) $t = \exp\{i\varphi\}$, где $i = \sqrt{-1}$ — мнимая единица. Умножим обе части полученного соотношения на $\exp\{-im\varphi\}$ и проинтегрируем его по переменной φ на отрезке $[-\pi, \pi]$. Учитывая, что система функций $\{\exp\{ik\varphi\}\}_{k=0}^{\infty}$ является ортогональной в пространстве $L_2[-\pi, \pi]$, получим следующее представление:

$$f_n^m [m_1, m_2, \ldots, m_n] = \frac{1}{2\pi} \int\limits_{-\pi}^{\pi} \exp(-im\varphi) \prod_{k=1}^{n} \frac{\exp\{i(m_k + 1)\varphi\} - 1}{\exp\{i\varphi\} - 1} d\varphi.$$

Аналогично, рассматривая тождество

$$\left(\sum_{j : r_j \leqslant m_1} t^j \right) \left(\sum_{j : r_j \leqslant m_2} t^j \right) \ldots \left(\sum_{j : r_j \leqslant m_n} t^j \right) = \sum_{\substack{0 \leqslant x_k \leqslant m_k,\ k = 1, \ldots, n, \\ x_k \in \{r_1, r_2, \ldots, r_N\}}} t^{x_1} t^{x_2} \ldots t^{x_n},$$

приходим к представлению

$$f_n^m\left[\frac{m_1,m_2,\ldots,m_n}{r_1,r_2,\ldots,r_N}\right]=\frac{1}{2\pi}\int\limits_{-\pi}^{\pi}\prod_{p=1}^{n}\sum_{j:0\leqslant r_j\leqslant m_p}\exp\left\{i\varphi\left(r_j-m\right)\right\}d\varphi. \tag{4}$$

Наконец, используя тождество

$$\left(\sum_{j:r_j\in I(m_1;r_1,r_2,\ldots,r_N)}t^j\right)\left(\sum_{j:r_j\in I(m_2;r_1,r_2,\ldots,r_N)}t^j\right)\ldots\left(\sum_{j:r_j\in I(m_n;r_1,r_2,\ldots,r_N)}t^j\right)=$$
$$=\sum_{\substack{0\leqslant x_k\leqslant m_k,\ k=1,\ldots,n,\\ x_k\in I(m_k;r_1,r_2,\ldots,r_N)}}t^{x_1}t^{x_2}\ldots t^{x_n},$$

получаем выражение для числа сочетаний второго типа с повторениями при блочных ограничениях:

$$F_n^m\left[\frac{m_1,m_2,\ldots,m_n}{r_1,r_2,\ldots,r_N}\right]=\frac{1}{2\pi}\int\limits_{-\pi}^{\pi}\prod_{p=1}^{n}\sum_{j:r_j\in I(m_p;r_1,r_2,\ldots,r_N)}\exp\left\{i\varphi\left(r_j-m\right)\right\}d\varphi. \tag{5}$$

3. Некоторые следствия и обобщения

Отметим, что если элементы в последовательности $\{r_1,r_2,\ldots,r_N\}$ рекуррентно связаны, например образуют геометрическую прогрессию, то подынтегральные выражения в соотношениях (4), (5) допускают дальнейшее упрощение. Например, если $r_j/r_{j-1}=a=\text{const}$ для $j=2,\ldots,N$, то, учитывая, что

$$\sum_{j:r_j\leqslant m_k}t^j=t^{r_1}\frac{t^{\psi(k)}-1}{t-1},\ \text{где}\ \psi\left(k\right)=\left[1+\log_a\left(\frac{m_k}{r_1}\right)\right],\ [z]-\text{целая часть числа } z,$$

соотношение (4) примет в этом случае более простую форму

$$f_n^m\left[\frac{m_1,m_2,\ldots,m_n}{r_1,r_2,\ldots,r_N}\right]=\frac{1}{2\pi}\int\limits_{-\pi}^{\pi}\exp\left\{-im\varphi\right\}A\left(i\varphi\right)d\varphi,$$

где

$$A\left(t\right)=\frac{t^{nr_1}}{\left(t-1\right)^n}\prod_{k=1}^{n}\left(t^{\psi(k)}-1\right).$$

Выше мы рассмотрели задачи о количестве выборок конфет различного состава для одного ребенка. В качестве небольшого обобщения рассмотрим задачу, когда конфеты необходимо распределить между несколькими детьми. Пусть имеется n ящиков с конфетами, причём в разных ящиках конфеты разных видов. Пусть также в j-м ящике находится r_j конфет. Конфеты из всех ящиков необходимо распределить между m детьми так, чтобы первому досталось ровно k_1 конфет, второму — k_2 конфет и т. д., последнему — k_m конфет. Необходимо определить, сколькими способами это можно сделать. Интерпретация данной задачи в терминах решений диофантовых уравнений приводит к следующей формулировке.

Рассматривается система линейных диофантовых уравнений

$$\begin{cases} x_{11}+x_{12}+\ldots+x_{1n}=k_1,\\ x_{21}+x_{22}+\ldots+x_{2n}=k_2,\\ \ldots\\ x_{m1}+x_{m2}+\ldots+x_{mn}=k_m, \end{cases}\qquad \begin{cases} x_{11}+x_{21}+\ldots+x_{m1}=r_1,\\ x_{12}+x_{22}+\ldots+x_{m2}=r_2,\\ \ldots\\ x_{1n}+x_{2n}+\ldots+x_{mn}=r_n, \end{cases} \tag{6}$$

где

$$\sum_{j=1}^{n} r_j = \sum_{i=1}^{m} k_i;\ k_i \geqslant 0,\ i = 1, \ldots, m;\ r_j \geqslant 0,\ j = 1, \ldots, n. \tag{7}$$

Необходимо определить количество её неотрицательных решений. Решение задачи (6), (7) обозначим символом $f_{m;n}^{k_1,\ldots,k_m;r_1,\ldots,r_n}$, который также может рассматриваться как некоторое расширение понятия числа сочетаний с повторениями. Для определения данного комбинаторного числа рассмотрим следующую промежуточную подзадачу. Обозначим через $\mathbf{A} = ||a_{ij} : i = 1, \ldots, m; j = 1, \ldots, n||$ произвольную матрицу с целыми неотрицательными элементами. Через $\mathbf{b} = ||b_i : i = 1, \ldots, m||$ обозначим произвольный вектор-столбец с целыми неотрицательными элементами. Рассматривается задача определения количества неотрицательных решений системы линейных диофантовых уравнений

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i,\ i = 1, \ldots, m. \tag{8}$$

Обозначим решение рассматриваемой задачи через $\mathcal{N}(\mathbf{A}, \mathbf{b})$. Данное число равно коэффициенту при $t_1^{b_1} t_2^{b_2} \ldots t_m^{b_m}$ в разложении по возрастающим показателям степеней производящей функции

$$\prod_{j=1}^{n} \sum_{k=0}^{K_j} t_1^{ka_{1j}} t_2^{ka_{2j}} \ldots t_m^{ka_{mj}}, \tag{9}$$

где $K_j = 1 + \max\left\{ \left[\frac{b_j}{a_{ij}}\right] + \operatorname{sgn}\left(\frac{b_j}{a_{ij}} - \left[\frac{b_j}{a_{ij}}\right]\right) :\ \forall i = 1, \ldots, m\ (a_{ij} > 0) \right\}$, $j = 1, \ldots, n$.

Поступая по аналогии, как и ранее, положим в (9) $t_k = \exp\{i\varphi_k\}$, $k = 1, \ldots, m$. Далее, интегрируя полученное выражение на кубе $[-\pi, \pi]^m$ по переменным φ_k, учитывая ортогональность системы функций $\left\{ \exp\left\{ -i \sum_{k=1}^{m} q_k \varphi_k \right\} : q_k \in \mathbb{Z}^+, k = 1, \ldots, m \right\}$ в пространстве $L_2[-\pi, \pi]^m$, приходим к следующему результату:

$$\mathcal{N}(\mathbf{A}, \mathbf{b}) = \frac{1}{(2\pi)^m} \int_{-\pi}^{\pi} \ldots \int_{-\pi}^{\pi} \exp\left\{ -i \sum_{k=1}^{m} b_k \varphi_k \right\} \prod_{j=1}^{n} \sum_{k=0}^{K_j} \exp\left\{ ik \sum_{i=1}^{m} a_{ij} \varphi_i \right\} d\varphi_1 \ldots d\varphi_m.$$

Для применения полученной формулы к определению числа целочисленных неотрицательных решений системы диофантовых уравнений (6), (7) перенумеруем неизвестные x_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, с помощью одного индекса k построчно слева направо и сверху вниз. Это соответствие для $k = 1, \ldots, mn$ задаётся правилом: $\nu_{n,m}(k) = (i, j)$, если $k = (i-1)n + j$.

Следовательно, задача (6), (7) допускает представление в виде (8), если положить

$$\mathbf{A} = ||a_{ij} : i = 1, \ldots, m+n, j = 1, \ldots, mn||,\ \mathbf{b} = ||b_i : i = 1, \ldots, m+n||,$$

где для $i = 1, \ldots, m+n$, $j = 1, \ldots, mn$

$$a_{ij} = \begin{cases} 1, \text{ если } i \in \{1, 2, \ldots, m\} \text{ и } \nu_{n,m}(j) \in \{(i, q) : 1 \leqslant q \leqslant n\}, \\ 1, \text{ если } i = m + p, 1 \leqslant p \leqslant n \text{ и } \nu_{n,m}(j) \in \{(q, p) : 1 \leqslant q \leqslant m\}, \\ 0 \text{ в остальных случаях;} \end{cases} \tag{10}$$

$$b_i = \begin{cases} k_i, \text{ если } 1 \leqslant i \leqslant m, \\ r_j, \text{ если } i = m + j, 1 \leqslant j \leqslant n. \end{cases} \tag{11}$$

Таким образом, справедлива формула $f_{m;n}^{k_1,\ldots,k_m;r_1,\ldots,r_n} = \mathcal{N}(\mathbf{A}, \mathbf{b})$, где матрица $\mathbf{A}$ и вектор $\mathbf{b}$ определяются с помощью соотношений (10), (11).

Пример. Рассмотрим применение полученных выше формул для решения задачи (6), (7). Пусть $m = 3$ и $n = 2$. Необходимо найти количество всех целых неотрицательных решений системы уравнений

$$\begin{cases} x_{11} + x_{12} = k_1, \\ x_{21} + x_{22} = k_2, \\ x_{31} + x_{32} = k_3, \end{cases} \quad \begin{cases} x_{11} + x_{21} + x_{31} = r_1, \\ x_{11} + x_{21} + x_{31} = r_2. \end{cases}$$

Пусть также $k_1 = 3$, $k_2 = 5$, $k_3 = 2$, $r_1 = 4$ и $r_2 = 6$. Очевидно, что условие (7) при этом выполнено. Матрица $\mathbf{A}$ и вектор $\mathbf{b}$ имеют вид

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 5 \\ 2 \\ 4 \\ 6 \end{bmatrix}.$$

Таким образом, количество всех целых неотрицательных решений рассматриваемой системы уравнений даётся формулой

$$f_{3;2}^{3,5,2;4,6} = \frac{1}{(2\pi)^5} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} F(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5)\, d\varphi_1 d\varphi_2 d\varphi_3 d\varphi_4 d\varphi_5 = 11,$$

где

$$F(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) = e^{-i(3\varphi_1 + 5\varphi_2 2\varphi_2 + 4\varphi_4 + 6\varphi_5)} \prod_{j=1}^{6} F_j(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5),$$

$$\begin{aligned}
F_1(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_1+\varphi_4)} + e^{2i(\varphi_1+\varphi_4)} + e^{3i(\varphi_1+\varphi_4)} + e^{4i(\varphi_1+\varphi_4)} + e^{5i(\varphi_1+\varphi_4)} + e^{6i(\varphi_1+\varphi_4)}, \\
F_2(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_1+\varphi_5)} + e^{2i(\varphi_1+\varphi_5)} + e^{3i(\varphi_1+\varphi_5)} + e^{4i(\varphi_1+\varphi_5)} + e^{5i(\varphi_1+\varphi_5)} + e^{6i(\varphi_1+\varphi_5)}, \\
F_3(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_2+\varphi_4)} + e^{2i(\varphi_2+\varphi_4)} + e^{3i(\varphi_2+\varphi_4)} + e^{4i(\varphi_2+\varphi_4)} + e^{5i(\varphi_2+\varphi_4)} + e^{6i(\varphi_2+\varphi_4)}, \\
F_4(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_2+\varphi_5)} + e^{2i(\varphi_2+\varphi_5)} + e^{3i(\varphi_2+\varphi_5)} + e^{4i(\varphi_2+\varphi_5)} + e^{5i(\varphi_2+\varphi_5)} + e^{6i(\varphi_2+\varphi_5)}, \\
F_5(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_3+\varphi_4)} + e^{2i(\varphi_3+\varphi_4)} + e^{3i(\varphi_3+\varphi_4)} + e^{4i(\varphi_3+\varphi_4)} + e^{5i(\varphi_3+\varphi_4)} + e^{6i(\varphi_3+\varphi_4)}, \\
F_6(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5) &= 1 + e^{i(\varphi_3+\varphi_5)} + e^{2i(\varphi_3+\varphi_5)} + e^{3i(\varphi_3+\varphi_5)} + e^{4i(\varphi_3+\varphi_5)} + e^{5i(\varphi_3+\varphi_5)} + e^{6i(\varphi_3+\varphi_5)}.
\end{aligned}$$

Заключение

Получены некоторые обобщения понятия числа сочетаний с повторениями. Найдены интегральные формулы для рассматриваемых комбинаторных чисел, применение которых может рассматриваться как альтернатива использованию различных переборных алгоритмов, реализуемых на ЭВМ с помощью языков программирования. Однако использование данных формул приводит к вычислению определённых интегралов; вопросы, связанные с эффективностью реализации операции интегрирования, требуют дальнейшего исследования.

ЛИТЕРАТУРА

1. *Виленкин Н. Я.* Комбинаторика. М.: Наука, 1969. 323 с.
2. *Новиков Ф. А.* Дискретная математика для программистов. СПб.: Питер, 2009. 384 с.
3. *Петровский А. Б.* Пространства множеств и мультимножеств. М.: УРСС, 2003. 248 с.
4. *MacMahon P. A.* Combinatory analysis. Cambridge: The University Press, 1915. 296 p.
5. *Juric Z. and Siljak H.* A new formula for the number of combinations and permutations of multisets // Appl. Math. Sci. 2011. V. 5. No. 18. P. 875–881.
6. *Заторский Р. А.* Подсчет m-подмультимножеств через их вторичные спецификации // Комбинаторный анализ. Вып. 7. М.: МГУ, 1986. С. 136–145.
7. *Полиа Г., Сеге Г.* Задачи и теоремы из анализа. М.: Наука, 1978. 391 с.

2013 Вычислительные методы в дискретной математике №2(20)

DOI 10.17223/20710410/20/9

УДК 519.7

ЭФФЕКТИВНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА МЕТОДОМ ВЕТВЕЙ И ГРАНИЦ

Ю. Л. Костюк

Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: kostyuk_y_l@sibmail.com

Для известного алгоритма точного решения задачи коммивояжёра методом ветвей и границ (алгоритма Литтла) предлагается следующая модификация. На каждой промежуточной стадии выполнения алгоритма вычисляется более точная оценка снизу для всех альтернативных маршрутов, которые можно построить на основе текущего частичного решения. Благодаря этому, отсечение бесперспективных вариантов выполняется, как правило, намного эффективнее, особенно для случайной несимметричной матрицы расстояний. Рассмотрены реализации модифицированного алгоритма с поиском вглубь и вширь, а также с поиском вглубь, когда ищется приближенный маршрут с произвольно задаваемой погрешностью. Для функции трудоёмкости $U(n) = a \cdot c^n$, измеряющей количество обрабатываемых матриц расстояний (узлов дерева решений) в графе с n вершинами, с помощью вычислительного эксперимента получены оценки констант a и c для различных реализаций алгоритма. При этом время обработки каждого узла увеличивается в 1,5–2 раза при существенном уменьшении общего времени выполнения алгоритма.

Ключевые слова: *задача коммивояжёра, метод ветвей и границ, вычислительный эксперимент.*

Введение

Как известно, задача коммивояжёра (ЗК) является NP-трудной задачей, поэтому точные алгоритмы её решения не могут иметь временную сложность меньше экспоненциальной. Решением ЗК является один из $(n-1)!$ циклов обхода по всем n вершинам полного взвешенного графа, сумма длин рёбер в котором минимальна. Такой граф обычно задаётся матрицей неотрицательных расстояний (длин рёбер графа для каждой пары вершин) размером $n \times n$, в которой диагональные элементы полагаются равными очень большой величине, теоретически бесконечности. В самом общем случае такая матрица асимметрична, а для её элементов не выполняется неравенство треугольника.

Одним из известных ранних алгоритмов точного решения ЗК для общего случая является алгоритм Литтла [1, 2], основанный на методе ветвей и границ, который строит дерево решений для перебора вариантов маршрута (циклов обхода) с отсечением. Отсекаются такие частично построенные маршруты, у которых оценка снизу длины маршрута больше или равна длине ранее построенного полного наилучшего маршрута. При построении оценки снизу на каждом этапе работы алгоритма матрица расстояний подвергается такому преобразованию с трудоемкостью $O(n^2)$, чтобы в каждой её строке и каждом столбце появился хотя бы один нуль. Более точную оценку снизу можно получать, решая задачу о назначениях на матрице расстояний за время $O(n^3)$,

при этом улучшается эффективность отсечений в дереве решений. В данной работе предлагается такое дополнительное преобразование матрицы расстояний, которое улучшает оценку снизу за время $O(n^2)$.

1. Алгоритм Литтла

Для вычисления оценки снизу длины частично построенного маршрута в алгоритме Литтла применяется следующее преобразование матрицы. Сначала из каждой строки вычитается её наименьший элемент, в результате в каждой строке образуется один (или больше) нулей. Затем из каждого столбца вычитается его наименьший элемент. Это преобразование основано на том факте, что если из любого столбца или строки матрицы вычесть константу, то стоимость оптимального маршрута уменьшается на величину этой константы, а сам маршрут остается прежним. Сумма всех вычтенных при этом величин и будет оценкой снизу для всех вариантов маршрута, строящегося по данной матрице.

Следующее действие — выбор некоторого ребра графа, такого, что все возможные варианты маршрута разбиваются на две группы: те, которые включают выбранное ребро, и те, которые его не включают, и для каждой группы создается своя матрица расстояний. Эти матрицы подвергаются аналогичному преобразованию с выбором ребра и т. д. Таким образом, строящееся при этом дерево решений получается двоичным, каждому его узлу соответствует своя матрица расстояний.

Выбор ребра на очередном шаге производится таким образом, чтобы оптимальный вариант маршрута содержал выбранное ребро с наибольшей вероятностью. Для этого просматриваются строки матрицы и среди них выделяется та, в которой второе минимальное ребро имеет наибольший вес. Затем аналогичным образом просматриваются столбцы и выделяется тот столбец, в котором второе минимальное ребро имеет наибольший вес. Окончательно выбирается такое ребро с нулевым весом, для которого вес второго минимального ребра в строке или столбце максимальный. Выбранное ребро включается в маршрут для вариантов маршрута 1-й группы и исключается из всех вариантов маршрута 2-й группы. В результате оценка снизу для всех вариантов маршрута 2-й группы увеличивается на вес второго минимального ребра в строке или столбце.

Рассмотрим более подробно реализацию алгоритма Литтла. На вход алгоритма подается квадратная матрица расстояний $\{C_{ij}\}$ размером $n \times n$, все $C_{ij} > 0$, причём диагональные элементы $C_{ii} = \infty$. Строки и столбцы матрицы помечаются отрезками маршрута, включёнными в строящийся вариант маршрута. В самом начале все эти отрезки суть отдельные вершины. Отрезок маршрута, начинающийся из вершины k и заканчивающийся в вершине i, задаётся парой чисел (k, i). После того как выбрано ребро (i, j), находящееся на пересечении строки, помеченной отрезком (k, i), и столбца, помеченного отрезком (j, l), для вариантов маршрута 1-й группы эти два отрезка объединяются. Для этого из матрицы расстояний удаляется строка (k, i) и столбец (j, l), в результате чего число строк и столбцов матрицы уменьшается на 1. Строка, помеченная отрезком, заканчивающимся в вершине l, и столбец, помеченный отрезком, начинающимся с вершины k, помечаются отрезком (k, l). Кроме того, элемент матрицы на их пересечении заменяется величиной ∞. Это делается для того, чтобы предотвратить в дальнейшем выбор ребра графа (l, k), замыкающего локальный цикл, не включающий все вершины.

Для вариантов маршрута 2-й группы выбранное ребро (i, j) в матрице заменяется величиной ∞, что предотвращает в последующем выбор этого ребра для маршрута. При этом размер матрицы не изменяется.

Когда размер матрицы уменьшится до 2×2, если при этом оценка снизу окажется меньше стоимости ранее найденного наилучшего маршрута, то строится окончательный замкнутый маршрут путем включения в него как ребра (i, j), так и ребра (l, k). Стоимость нового маршрута совпадает с его оценкой снизу, и он запоминается как наилучший маршрут среди всех ранее просмотренных.

Алгоритм Литтла просматривает строящееся таким образом дерево решений вглубь, в котором сначала делается переход к первой группе вариантов маршрута, а после их просмотра — ко второй группе. Согласно обзору [3], возможен также упорядоченный обход вширь дерева решений, когда на каждом этапе выбирается тот узел дерева решений, которому соответствует матрица с минимальной оценкой снизу. Согласно тому же обзору [3], возможна более точная оценка длины маршрута, вычисленная как решение задачи о назначениях по заданной матрице. Однако если каждое преобразование матрицы в алгоритме Литтла имеет трудоёмкость порядка $\mathrm{O}(n^2)$, то получить решение задачи о назначениях можно лишь за время $\mathrm{O}(n^3)$, используя для этого венгерский алгоритм [4]. Общая же трудоёмкость определяется произведением

1) трудоёмкости преобразования матрицы и получения оценки снизу длины маршрута;

2) количеством просматриваемых узлов дерева решений.

Так как дерево решений двоичное, а его высота обычно ненамного превышает количество n вершин графа расстояний, то количество узлов в нём имеет порядок $\mathrm{O}(2^n)$. Реальная же трудоёмкость алгоритма определяется эффективностью отсечений при просмотре дерева решений, её порядок — $\mathrm{O}(n^2 \cdot c^n)$ для алгоритма Литтла и $\mathrm{O}(n^3 \cdot c^n)$ для случая, когда оценка снизу в узлах дерева решений вычисляется как решение задачи о назначениях, при этом константа c имеет меньшую величину. Эту константу можно оценить с помощью вычислительного эксперимента.

2. Модифицированный алгоритм

Модификация алгоритма Литтла состоит в том, что при обработке алгоритмом очередного узла в дереве решений матрица расстояний подвергается дополнительному преобразованию. При вычислении оценки снизу стоимости вариантов маршрутов, строящихся из текущего узла, в матрице расстояний в каждой строке и каждом столбце появляется не менее чем по одному нулю.

Дополнительное преобразование состоит в том, что в матрице отмечается такая группа строк, в которых имеется ровно по одному нулевому элементу и все эти нули находятся в одном и том же столбце. Если такая группа, состоящая хотя бы из двух строк, найдена, то минимальный ненулевой элемент (пусть его величина равна a) в этих строках вычитается из всех строк группы и добавляется к тому столбцу, в котором находятся нули. В результате ранее полученные нули в этой группе строк сохраняются и добавляется, по крайней мере, один новый нуль. Оптимальный маршрут при этом не изменяется, а его стоимость уменьшается на величину $a(p-1)$, где p — количество строк в группе. На такую же величину увеличивается оценка снизу стоимости вариантов маршрутов, строящихся из текущего узла. Матрица просматривается до тех пор, пока не будут обработаны все такие группы строк. Далее аналогичным образом в матрице просматриваются столбцы, которые, если будут найдены подобные группы, подвергаются аналогичному преобразованию.

Такое преобразование матрицы является частным случаем преобразования, используемого в алгоритме решения задачи о назначении венгерским методом [4]. Заметим, что здесь не требуется полностью решать задачу о назначении для всей матрицы расстояний, поэтому дополнительное преобразование матрицы имеет порядок $O(n^2)$ и увеличивает время обработки одного узла дерева решений не более чем в 1,5–2 раза. Благодаря такому дополнительному преобразованию, для многих узлов дерева решений увеличивается оценка снизу стоимости вариантов маршрутов, вследствие чего отсечение ветвей дерева происходит раньше. Поэтому количество обрабатываемых узлов чаще всего будет существенно меньше, чем в известном алгоритме. В качестве иллюстрации рассмотрим решение ЗК для следующей матрицы расстояний размера 5×5, которая является частью матрицы 7×7 примера, приведённого в [2]:

	1	2	3	4	5
1	∞	3	93	13	33
2	4	∞	77	42	21
3	45	17	∞	36	16
4	39	90	80	∞	56
5	28	46	88	33	∞

Здесь дерево решений будем просматривать вглубь. В корневом узле дерева решений матрица подвергается следующей обработке. После вычитания из строк и столбцов минимальных элементов получим оценку снизу, равную 136, и матрицу

	1	2	3	4	5
1	∞	0	49	5	30
2	0	∞	32	33	17
3	29	1	∞	15	0
4	0	51	0	∞	17
5	0	18	19	0	∞

Дополнительное преобразование не изменяет эту матрицу. Ребро графа, которое здесь следует выбрать, — $(4, 3)$. В 3-м столбце второй минимальный элемент равен 19, поэтому для правого узла дерева решений оценка снизу будет не менее чем $136 + 19 = 155$. Матрица для левого узла дерева решений получается путём удаления 4-й строки и 3-го столбца, а также замене ребра $(3, 4)$ на ∞:

	1	2	4–3	5
1	∞	0	5	30
2	0	∞	33	17
4–3	29	1	∞	0
5	0	18	0	∞

Преобразование не изменяет эту матрицу. Выбирается ребро графа $(2, 1)$. Оценка снизу для правого узла: $136 + 17 = 153$. Матрица для левого узла после удаления 2-й строки и 1-го столбца и замены ребра $(1, 2)$ на ∞:

	2–1	4–3	5
2–1	∞	5	30
4–3	1	∞	0
5	18	0	∞

После вычитания из 1-й строки числа 5 и из 1-го столбца числа 1 оценка снизу 142, а матрица равна

	2–1	4–3	5
2–1	∞	0	25
4–3	0	∞	0
5	17	0	∞

Здесь в 1-й и 3-й строках имеется по одному нулю во 2-м столбце, в этих строках второй минимальный элемент равен 17, поэтому из 1-й и 3-й строк вычитается 17, а ко 2-му столбцу добавляется 17. Новая оценка снизу равна $142 + 17 = 159$, а преобразованная матрица имеет вид

	2–1	4–3	5
2–1	∞	0	8
4–3	0	∞	0
5	0	0	∞

Выбирается ребро графа $(1, 4)$, в результате два отрезка маршрута 2–1 и 4–3 объединяются в отрезок 2–1–4–3. Оценка снизу для правого узла: $159 + 8 = 167$. Матрица для левого узла после удаления 1-й строки и 2-го столбца и замены ребра $(3, 2)$ на ∞:

	2–3	5
2–3	∞	0
5	0	∞

Преобразование не изменяет эту матрицу. Так как её размер 2×2, то в маршрут добавляются рёбра $(3, 5)$ и $(5, 2)$. Наилучшее решение на этом этапе 1–4–3–5–2, его стоимость 159.

Возврат по дереву решений к матрице 3×3 даёт оценку правой ветви 167, поэтому производится ещё один возврат, к матрице 4×4, которая даёт оценку правой ветви 153, что меньше, чем 159. Матрица в правой ветви дерева решений:

	1	2	4–3	5
1	∞	0	5	30
2	∞	∞	33	17
4–3	29	1	∞	0
5	0	18	0	∞

После вычитания из 2-й строки числа 17 оценка снизу равна 153, а матрица

	1	2	4–3	5
1	∞	0	5	30
2	∞	∞	16	0
4–3	29	1	∞	0
5	0	18	0	∞

Здесь во 2-й и 3-й строках имеется по одному нулю в 4-м столбце, в этих строках второй минимальный элемент равен 1, поэтому из 2-й и 3-й строк вычитается 1, а к 4-му столбцу добавляется 1. Кроме того, в 1-м и 3-м столбцах имеется по одному нулю в 4-й строке, в этих столбцах второй минимальный элемент равен 5, поэтому из 1-го и 3-го столбца вычитается 5, а к 5-й строке добавляется 5. В результате новая оценка снизу равна $153+1+5=159$, что равно ранее полученной стоимости. Поэтому делается возврат по дереву решений к матрице 5×5, у неё для правой ветви оценка 155, а матрица после вычитания числа 19 из 3-го столбца равна

	1	2	3	4	5
1	∞	0	30	5	30
2	0	∞	13	33	17
3	29	1	∞	15	0
4	0	51	∞	∞	17
5	0	18	19	0	∞

Здесь во 2-й и 4-й строках имеется по одному нулю в 1-м столбце, в этих строках второй минимальный элемент равен 13, поэтому из 2-й и 3-й строк вычитается 13, а к 1-му столбцу добавляется 13. Новая оценка снизу равна $155+13=168$, что больше ранее полученной стоимости, поэтому делается возврат к корню дерева решений и алгоритм завершает свою работу.

На рис. 1 изображено получившееся дерево решений, в котором прямоугольники обозначают вершины, подвергнутые обработке.

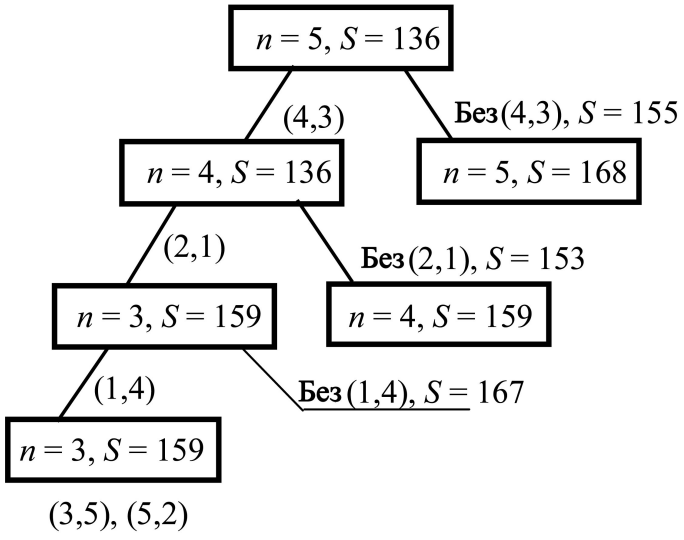


Рис. 1. Дерево решений для примера ЗК с матрицей 5×5

Таким образом, для получения оптимального маршрута 1–4–3–5–2 стоимостью 159 всего было построено и обработано 6 узлов дерева решений. Для сравнения, алгоритм Литтла получил такой же маршрут, построив и обработав 13 узлов дерева решений.

3. Реализация алгоритмов

В алгоритмах используются следующие глобальные переменные:

– стоимость $S_{\min}$ наилучшего на текущий момент построенного маршрута;

– размер n_0 исходной матрицы расстояний;

– массив P_{opt} размером n_0, в котором записан наилучший на текущий момент построенный маршрут.

Удобнее всего реализовать объектную модель, в которой каждый узел дерева решений представляется экземпляром класса со следующей структурой:

– указатель на другой узел;

– размер матрицы в узле n;

– 1-я оценка снизу S_0, равная сумме вычтенных из строк и столбцов матрицы величин;

– 2-я оценка снизу S, более точная, чем S_0 ($S \geqslant S_0$);

– матрица расстояний в узле M, её размер $n \times n$;

– два массива размером n, в которых записаны начальные и конечные вершины отрезков частично построенного маршрута;

– четыре массива размером n, в которых записаны номера нулевых элементов и вторых наименьших элементов в строках и столбцах;

– массив P размером n_0, в котором записан частично построенный маршрут, для каждого входящего в маршрут ребра (i, j): $P[i] = j$.

В алгоритме узлы дерева решений организованы в виде линейного списка, представляющего собой обход частично построенного дерева решений по висячим узлам слева направо. Первый узел в списке обрабатывается на очередном этапе, при этом возможны следующие варианты работы алгоритма:

1) маршрут достроен до конца, его вес меньше ранее построенного маршрута, новый вариант маршрута запоминается, а узел удаляется из списка;

2) оценка снизу всех маршрутов, которые могут быть построены из текущего узла, не лучше веса ранее построенного маршрута, узел удаляется из списка;

3) оценка снизу всех маршрутов, которые могут быть построены из текущего узла, меньше веса ранее построенного маршрута, тогда выбирается лучшее ребро для маршрута, создается новый первый узел с включением этого ребра, а в текущем узле это ребро удаляется. При этом проверяется оценка снизу для будущих маршрутов из текущего узла, и если она окажется не лучше веса ранее построенного маршрута, то текущий узел удаляется из списка.

Далее приведён общий вид алгоритма Литтла, реализующего поиск вглубь.

1. Задание исходного значения n_0, а также присваивание $S_{\min} = \infty$.
2. Создание начального элемента списка — корня дерева решений, в котором:
   2.1. Указателю на другой узел присваивается пустая ссылка.
   2.2. Размер матрицы в узле $n = n_0$.
   2.3. Обеим оценкам снизу S и S_0 присваиваются нули.
   2.4. Матрице расстояний в узле M задаются исходные значения.
   2.5. Каждому элементу массивов, содержащих начальные и конечные вершины отрезков маршрута, задаются значения, равные их номерам.
3. Цикл, пока список узлов дерева решений не пуст:

3.1. Если $S < S_{\min}$, то для первого узла в списке:
  3.1.1. Вычитание из строк матрицы минимальных значений с коррекцией S_0 и S.
  3.1.2. Вычитание из столбцов матрицы минимальных значений с коррекцией S_0 и S.
  3.1.3. Формирование массивов с номерами нулевых элементов и вторых наименьших элементов в строках.
  3.1.4. Формирование массивов с номерами нулевых элементов и вторых наименьших элементов в столбцах.
3.2. Если $S < S_{\min}$, то для первого узла в списке:
  3.2.1. Выбор наилучшего ребра для включения в маршрут.
  3.2.2. Если $n = 2$, то:
    3.2.2.1. Выбор второго ребра для маршрута.
    3.2.2.2. Запоминание нового маршрута в P_{opt}.
    3.2.2.3. Присваивание $S_{\min} = S_0$.
    3.2.2.4. Удаление первого узла из списка.
  3.2.2. Иначе:
    3.2.2.5. Замена выбранного ребра в матрице величиной ∞ с коррекцией S.
    3.2.2.6. Создание нового узла с размером матрицы в узле $n - 1$ и со ссылкой на предыдущий первый узел.
    3.2.2.7. Копирование матрицы в новый узел (без той строки и того столбца, где было выбранное ребро).
    3.2.2.8. Копирование с коррекцией массивов с записанными начальными и конечными вершинами отрезков маршрута.
    3.2.2.9. Копирование массива P с добавлением выбранного ребра.
    3.2.2.10. Копирование S_0 и S.
3.2. Иначе:
  3.2.3. Удаление первого узла из списка.

Модифицированный алгоритм содержит следующее дополнительное преобразование матрицы, которое необходимо вставить после п. 3.1.4:

3.1.5. Если $S < S_{\min}$, то для первого узла в списке:
  3.1.5.1. Цикл, пока не просмотрены все строки
    3.1.5.1.1. Если найдена строка с одним нулем, то:
      3.1.5.1.1.1. Цикл с поиском других строк с нулем в том же столбце.
      3.1.5.1.1.2. Если найдено две или больше строки, то коррекция строк и столбца, коррекция S_0.
  3.1.5.2. Цикл, пока не просмотрены все столбцы
    3.1.5.2.1. Если найден столбец с одним нулем, то:
      3.1.5.2.1.1. Цикл с поиском других столбцов с нулем в той же строке.
      3.1.5.2.1.2. если найдено два или больше столбца, то коррекция столбцов и строк, коррекция S_0.
  3.1.5.3. коррекция S.

Алгоритм с просмотром дерева решений вглубь можно использовать для получения не только точного решения ЗК, но и приближённого с произвольно заданной степенью погрешности, при этом он будет выполняться во много раз быстрее. Для этого в него надо внести следующие изменения. Везде, где встречается проверка «если $S < S_{\min}$», её надо заменить на «если $S(1 + \varepsilon) < S_{\min}$», где ε — величина погрешности. При этом

будет гарантировано, что стоимость найденного маршрута не более чем в $(1+\varepsilon)$ раз больше стоимости оптимального маршрута. Для рассмотренного алгоритма такие замены следует произвести в п.п. 3.1, 3.2, 3.1.5.

Алгоритм с просмотром дерева решений вширь дополнительно содержит следующие действия по упорядочению узлов дерева решений:

3.3. Если 2-й узел в списке существует, то для него:
  3.3.1. Поиск i-го узла в списке, у которого $S_i \geqslant S$.
  3.3.2. Вставка 2-го узла в список перед i-м узлом.
3.4. Если 1-й узел в списке существует, то для него:
  3.3.1. Поиск i-го узла в списке, у которого $S_i \geqslant S$.
  3.3.2. Вставка 1-го узла в список перед i-м узлом.

4. Вычислительный эксперимент

Рассмотренные алгоритмы проверены на следующих модельных данных. Элементы матрицы расстояний M_{ij} генерируются как равномерно распределённые случайные числа из диапазона от 0 до 1000, причём элементы M_{ij} и M_{ji} независимы друг от друга. Размеры матрицы n изменяются в диапазоне от 30 до 100, для каждого n строится от 3000 (для $n = 30$) до 400 (для $n = 100$) вариантов матриц, и для каждого экземпляра матрицы маршруты вычисляются (в общем случае) следующими пятью алгоритмами:

1) алгоритмом Литтла с просмотром вглубь;

2) модифицированным алгоритмом с просмотром вглубь;

3) модифицированным алгоритмом с просмотром вглубь, вычисляющим приближённое решение с $\varepsilon = 0{,}05$;

4) модифицированным алгоритмом с просмотром вширь;

5) модифицированным алгоритмом с просмотром вширь, но с начальным приближением, полученным алгоритмом 3.

Все алгоритмы написаны на языке Паскаль в системе Delphi, вычисления производились на компьютере с процессором Atlon, имеющим кэш-память по 1 Мбайт на каждое процессорное ядро и работающим на частоте 2,9 ГГц. В табл. 1 представлены среднее количество обрабатываемых узлов дерева решений и среднее время вычисления маршрута, полученные по одним и тем же сгенерированным экземплярам матриц для всех пяти алгоритмов. Отсутствуют только те результаты вычислений, для получения которых потребовалось бы слишком много времени.

Кроме того, подсчитана величина разброса (как среднеквадратичное отклонение) для количества узлов дерева решений и затраченного времени. Величина разброса оказалась очень большой для всех случаев, для максимальных размерностей она в 1,5–2 раза превышает соответствующие средние значения. Это является следствием того, что ЗК является NP-трудной, поэтому для наихудших (с точки зрения алгоритмов) экземпляров матрицы количество обработанных узлов может быть больше среднего количества в десятки раз. Из табл. 1 видно, что алгоритму с просмотром вширь требуется обработать несколько меньше узлов дерева решений, чем аналогичному алгоритму с просмотром вглубь, однако на уменьшении времени работы это сказывается лишь при n, меньших 50. При больших n время работы резко увеличивается, при $n = 80$ оно более чем в 10 раз превышает время работы алгоритма с просмотром вглубь. При этом не помогает задание хорошего начального приближения перед выполнением алгоритма с просмотром вширь. Это происходит из-за того, что в алгоритме с просмотром вширь максимальный размер списка узлов дерева решений растёт экспоненциально с ростом n, и когда размер используемой алгоритмом памяти начинает

Таблица 1

Результаты эксперимента

Размер матрицы n	Алгоритм 1		Алгоритм 2		Алгоритм 3		Алгоритм 4		Алгоритм 5	
	Сред. кол-во узлов	Сред. время, с	Сред. кол-во узлов	Сред. время, с	Сред. кол-во узлов	Сред. время, с	Сред. кол-во узлов	Сред. время, с	Сред. кол-во узлов	Сред. время, с
30	999	0,0156	163	0,0039	60	0,0019	118	0,0035	118	0,0034
40	5807	0,134	431	0,0148	115	0,0045	315	0,0131	315	0,0131
50	31506	1,051	980	0,0472	191	0,0102	724	0,0475	724	0,0450
60	189022	8,241	2421	0,160	390	0,029	1847	0,241	1847	0,186
70	–	–	5277	0,459	728	0,071	4076	1,454	4076	1,149
80	–	–	13797	1,481	1485	0,175	11100	18,982	11100	11,159
90	–	–	33504	4,562	2589	0,391	–	–	–	–
100	–	–	89043	14,679	8478	1,602	–	–	–	–

существенно превышать размер кэш-памяти в процессоре, происходит интенсивный обмен данными между кэш-памятью и основной памятью компьютера. В то же время максимальный размер списка узлов дерева решений для алгоритма с просмотром вглубь не превышает n, и весь список обычно помещается в кэш-память. Трудоёмкость по количеству обрабатываемых узлов дерева решений всех алгоритмов приближённо имеет порядок $\mathrm{O}(c^n)$, а трудоёмкость по времени — $\mathrm{O}(n^2 \cdot c^n)$, где константа c имеет разную величину для разных алгоритмов. Оценим величину константы c по методу наименьших квадратов. Количество обрабатываемых узлов U как функция от n:

$$U(n) = a \cdot c^n, \tag{1}$$

где a и c — константы.

Чтобы избавиться от константы a, вместо функции $U(n)$ будем рассматривать отношение двух значений этой функции от разных значений n:

$$U(n_2)/U(n_1) = c^{n_2 - n_1}. \tag{2}$$

После логарифмирования равенства (2) можно записать сумму квадратов разностей измеренных и вычисленных $U(n)$, причём отношение будем записывать для пар соседних измерений:

$$\sum_i \left(\ln \frac{U_i}{U_{i-1}} - (n_i - n_{i-1}) \ln c \right)^2 \to \min_{\ln c}. \tag{3}$$

После дифференцирования (3) по параметру $\ln c$ и приравнивания производной нулю получим выражение для оценки c:

$$c = \exp \left(\sum_i \ln \frac{U_i}{U_{i-1}} (n_i - n_{i-1}) / \sum_i (n_i - n_{i-1})^2 \right). \tag{4}$$

Для оценивания константы a прологарифмируем (1) и запишем сумму квадратов разностей на основе измеренных и вычисленных $U(n)$:

$$\sum_i \left(\ln U_i - \ln a - n_i \cdot \ln c\right)^2 \to \min_{\ln a}. \tag{5}$$

После дифференцирования (5) по параметру $\ln a$ и приравнивания производной нулю получим выражение для оценки a:

$$a = \exp\left(\frac{1}{K}\sum_{i=1}^{K}(\ln U_i - n_i \cdot \ln c)\right). \tag{6}$$

Вычисленные по формулам (4) и (6) оценки параметров a и c для количества обрабатываемых узлов дерева решений (1) у рассмотренных алгоритмов сведены в табл. 2.

Таблица 2

Оценки параметров a и c

Параметр	Алгоритм 1	Алгоритм 2	Алгоритм 3	Алгоритм 4	Алгоритм 5
a	5,238	10,665	5,811	7,725	7,725
c	1,191	1,094	1,073	1,095	1,095

Одной из особенностей многих точных алгоритмов решения NP-трудных задач является то, что после нахождения оптимального решения алгоритм ещё долго перебирает другие варианты решения, пытаясь улучшить найденное. Поэтому представляет интерес выяснить, какова доля времени, затрачиваемого на поиск оптимального маршрута, по сравнению с общим временем вычислений. Оценки среднего значения этого параметра для алгоритмов 1, 2 и 3 приведены в табл. 3. Алгоритмы 4 и 5 реализуют поиск вширь таким образом, что самый первый из найденных вариантов является оптимальным маршрутом, поэтому для них эта доля времени близка к единице.

Таблица 3

Доля времени, затрачиваемого на поиск оптимального маршрута

Размер матрицы n	Алгоритм 1	Алгоритм 2	Алгоритм 3, $\varepsilon = 0{,}05$
30	0,51	0,58	0,72
40	0,39	0,46	0,56
50	0,37	0,35	0,36
60	0,38	0,37	0,32
70	–	0,31	0,33
80	–	0,26	0,29
90	–	0,26	0,22
100	–	0,29	0,23

Для алгоритма 3, вычисляющего приближенное решение ЗК с гарантированной точностью ε, представляет интерес, какова при этом средняя погрешность полученного решения и во сколько раз время вычислений меньше, чем у алгоритма 2. Оценки этих величин представлены в табл. 4 для $\varepsilon = 0{,}05$, 0,1 и 0,2.

Таблица 4

Характеристики алгоритма 3

Размер матрицы n	$\varepsilon = 0{,}05$		$\varepsilon = 0{,}1$		$\varepsilon = 0{,}2$	
	Средняя погрешность	Доля времени вычислений	Средняя погрешность	Доля времени вычислений	Средняя погрешность	Доля времени вычислений
30	0,012	0,50	0,022	0,39	0,041	0,31
40	0,013	0,31	0,027	0,21	0,045	0,15
50	0,016	0,22	0,030	0,12	0,051	0,08
60	0,016	0,18	0,032	0,09	0,055	0,05
70	0,018	0,15	0,033	0,053	0,053	0,026
80	0,018	0,12	0,033	0,049	0,052	0,015
90	0,018	0,085	0,035	0,050	0,052	0,014
100	0,018	0,108	0,036	0,038	0,052	0,014

Следует отметить, что результаты проведённого вычислительного эксперимента справедливы лишь для случая несимметричной матрицы расстояний с равномерным распределением длин рёбер. Для других распределений рассмотренные алгоритмы имеют, как правило, намного большую трудоёмкость. Так, для известной ЗК с 48 городами [5], с симметричной матрицей, алгоритм 1 получил результат за 3,5 ч работы, а алгоритм 2 — за 1 ч 45 мин. Интересно, что при этом получен маршрут длиной 11461, что меньше, чем в [5], где длина маршрута указана равной 11470. Алгоритмы 1 и 2 вычислили следующий оптимальный маршрут:
1 2 48 15 43 21 33 30 23 9 10 40 36 34 6 8 47 7 38 14 18 12 22 13 28 32 25 3 5 29 26 41 24 35 17 31 20 11 16 42 4 46 45 39 44 27 37 19.

Программа, реализующая алгоритм 2, включая модуль с описанием класса для решения ЗК, доступна в сети Интернет по ссылке [6].

Заключение

Как показал вычислительный эксперимент, предложенная модификация алгоритма Литтла для случайной матрицы расстояний позволяет существенно сократить количество обрабатываемых узлов дерева решений при поиске оптимального маршрута в задаче коммивояжёра. При этом время обработки каждого узла имеет тот же порядок трудоёмкости, $O(n^2)$, но с увеличенным в 1,5–2 раза множителем. В целом, трудоёмкость обоих алгоритмов имеет порядок $O(n^2 \cdot c^n)$, но у модифицированного алгоритма константа c существенно меньше, что позволяет при примерно одинаковом времени решать задачу для $n = 100$ против $n = 60$. Сравнение реализаций модифицированного алгоритма с поиском вглубь и вширь показало незначительное преимущество поиска вширь только для небольших размерностей ($n < 50$), когда объём требуемой памяти для размещения матриц в узлах дерева решений не слишком велик. Для $n \geqslant 50$ время

работы алгоритма с поиском вширь резко увеличивается из-за ограниченного объёма кэш-памяти.

ЛИТЕРАТУРА

1. *Little J. D. C., Murty K. G., Sweeney D. W., and Karel C.* An algorithm for the Traveling Salesman Problem // Operations Research. 1963. No. 11. P. 972–989.
2. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика: пер. с англ. М.: Мир, 1980. 478 с.
3. *Меламед И.Й., Сергеев С. И., Сигал И. Х.* Задача коммивояжёра. Точные алгоритмы // Автоматика и телемеханика. 1989. № 10. С. 3–29.
4. *Данциг Дж.* Линейное программирование, его применения и обобщения: пер. с англ. М.: Прогресс, 1966.
5. *Хелд М., Карп Р. М.* Применения динамического программирования к задачам упорядочивания // Кибернетический сборник. Вып. 9, старая серия. М.: Мир, 1964. С. 202–218.
6. *Костюк Ю. Л.* Модифицированный алгоритм решения задачи коммивояжёра методом ветвей и границ. Программа и модуль с описанием класса: язык Паскаль в системе Delphi // Электронный ресурс: `www.inf.tsu.ru/Decanat/Staff.nsf/people/KostjukJuL`, 2013.

2013 Вычислительные методы в дискретной математике №2(20)
DOI 10.17223/20710410/20/10
УДК 511

МОДИФИКАЦИЯ МЕТОДА ЛАГАРИАСА — ОДЛЫЖКО ДЛЯ РЕШЕНИЯ ОБОБЩЁННОЙ ЗАДАЧИ О РЮКЗАКЕ И СИСТЕМ ЗАДАЧ О РЮКЗАКАХ

Д. М. Мурин

Ярославский государственный университет им. П. Г. Демидова, г. Ярославль, Россия

E-mail: nirum87@mail.ru

В работе 1985 г. Дж. Лагариас и А. Одлыжко предложили метод решения вычислительной задачи о рюкзаке, основанный на её сведении к задаче поиска короткого вектора в решётке специального вида. Метод Лагариаса — Одлыжко позволяет решать «практически все» задачи о рюкзаках, обладающие малой плотностью. В настоящей работе метод Лагариаса — Одлыжко решения задачи о рюкзаке модифицируется применительно к случаям обобщённой задачи о рюкзаке и систем задач о рюкзаках. Система задач о рюкзаках вводится в настоящей работе как конечное множество индивидуальных задач о рюкзаках, имеющих общее решение. Определяются множества значений плотности обобщённых задач о рюкзаках и систем задач о рюкзаках, такие, что модифицированные методы позволяют решать «практически все» обобщённые задачи о рюкзаках и системы задач о рюкзаках, обладающие плотностью из этих множеств.

Ключевые слова: *метод Лагариаса — Одлыжко, задача о рюкзаке, обобщённая задача о рюкзаке, системы задач о рюкзаках.*

Введение

Задача о рюкзаке (распознавания) NP-полна [1]. Однако на настоящий момент известны два метода решения вычислительных задач о рюкзаках, обладающих малой плотностью, один из которых предложен Дж. Лагариасом и А. Одлыжко [2], а второй — Э. Брикеллом [3]. Дж. Лагариас и А. Одлыжко рассматривали *вычислительную задачу о рюкзаке* в следующей формулировке.

Условие: заданы вектор $A = (a_1, a_2, \ldots, a_r) \in \mathbb{N}^r$ и число $S \in \mathbb{N}$, причём известно, что уравнение

$$\sum_{i=1}^{r} a_i x_i = S, \tag{1}$$

где $x_1, \ldots, x_r$ — неизвестные, имеет решение в числах 0 и 1.

Вопрос: найти решение уравнения (1) в числах 0 и 1.

В работе [2] рассматривается понятие *плотности* задачи о рюкзаке.

Определение 1. *Плотностью* задачи о рюкзаке с вектором $A = (a_1, a_2, \ldots, a_r) \in \mathbb{N}^r$ называется число

$$d_{\text{з.р}} = \frac{r}{\log_2 \left(\max\limits_{1 \leqslant i \leqslant r} a_i \right)}.$$

Метод Лагариаса — Одлыжко основан на сведении задачи о рюкзаке к задаче поиска короткого вектора в решётке специального вида. Предположим, что имеется алгоритм «Оракул», который за полиномиальное время выдаёт один из кратчайших (самых коротких) ненулевых векторов решёток некоторых специальных видов (виды

решёток описаны далее). Несмотря на то, что задача нахождения кратчайшего ненулевого вектора решётки в общем случае является NP-трудной [4], ответ на вопрос о существовании алгоритма «Оракул» неизвестен, при реализации «приближением» к этому алгоритму служит алгоритм построения LLL-приведённого базиса решётки.

Дж. Лагариас и А. Одлыжко рассматривали решётку $\Lambda_1 = \{z_1 b_1 + \ldots + z_{r+1} b_{r+1} : z_1, \ldots, z_{r+1} \in \mathbb{Z}\}$, образованную следующими линейно независимыми векторами размерности $r+1$:

$$\begin{gathered} b_1 = (1, 0, \ldots, 0, N \cdot a_1), \\ b_2 = (0, 1, \ldots, 0, N \cdot a_2), \\ \vdots \\ b_r = (0, 0, \ldots, 1, N \cdot a_r), \\ b_{r+1} = (0, 0, \ldots, 0, N \cdot S), \end{gathered}$$

где $N \geqslant \left[\sqrt{r/2}\right] + 1$ — некоторое достаточно большое натуральное число, а вектор $(a_1, \ldots, a_r, S) \in \mathbb{N}^{r+1}$ задаёт вычислительную задачу о рюкзаке. В работе [2] показано, что справедлива следующая теорема.

Теорема 1. Пусть $a_1, \ldots, a_r$ — произвольные натуральные числа, $\tilde{e} = (e_1, \ldots, e_r) \in \in \{0,1\}^r$ — произвольный вектор и $S = \sum\limits_{i=1}^{r} e_i a_i$. Тогда вероятность ошибки при решении задачи о рюкзаке, заданной значениями a_i, $1 \leqslant i \leqslant r$, и S, при условии, что плотность этой задачи $d_{\text{з.р}} < 0{,}6463\ldots$, с помощью алгоритма «Оракул», применённого к решётке Λ_1, стремится к 0 при $r \to \infty$. (Здесь и далее под ошибкой понимается получение с помощью «Оракула» вектора, отличного от $\tilde{e}$ и $-\tilde{e}$.)

Этот результат улучшен в работе [5] путём модификации исследуемой решётки. Определим решётку $\Lambda_2 = \{z_1 b_1^* + \ldots + z_{r+1} b_{r+1}^* : z_1, \ldots, z_{r+1} \in \mathbb{Z}\}$, где $b_i^* = b_i$ при $1 \leqslant i \leqslant r$ и $b_{r+1}^* = (1/2, 1/2, \ldots, 1/2, N \cdot S)$, $N \geqslant \left[\sqrt{r}/2\right] + 1$ — некоторое достаточно большое натуральное число. Заметим, что в данном случае векторы $b_1^*, \ldots, b_{r+1}^*$ не обязательно линейно независимы.

Справедлива следующая теорема [5].

Теорема 2. Пусть $a_1, \ldots, a_r$ — произвольные натуральные числа, $\tilde{e} = (e_1, \ldots, e_r) \in \in \{0,1\}^r$ — произвольный вектор и $S = \sum\limits_{i=1}^{r} e_i a_i$. Тогда вероятность ошибки при решении задачи о рюкзаке, заданной значениями a_i, $1 \leqslant i \leqslant r$, и S, при условии, что плотность этой задачи $d_{\text{з.р}} < 0{,}9408\ldots$, с помощью алгоритма «Оракул», применённого к решётке Λ_2, стремится к 0 при $r \to \infty$.

В настоящей работе предлагаются методы решения вычислительной обобщённой задачи о рюкзаке и систем задач о рюкзаках, основанные на методе Лагариаса — Одлыжко решения вычислительной задачи о рюкзаке.

Вычислительная обобщённая задача о рюкзаке рассматривается в следующей формулировке.

Условие: заданы вектор $A = (a_1, a_2, \ldots, a_r) \in \mathbb{N}^r$, число $S \in \mathbb{N}$ и натуральное число $m \geqslant 2$, причём известно, что уравнение

$$\sum_{i=1}^{r} a_i x_i = S, \tag{2}$$

где $x_1, \ldots, x_r$ — неизвестные, имеет решение в числах $0, 1, \ldots, m-1$.

Вопрос: найти решение уравнения (2) в числах $0, 1, \ldots, m-1$.

Замечание. Числа $0, 1, \ldots, m-1$ можно рассматривать как элементы кольца вычетов $\mathbb{Z}/m\mathbb{Z}$, но поскольку в обобщённой задаче о рюкзаке рассматривается обычная сумма $\sum\limits_{i=1}^{r} a_i x_i$ натуральных чисел, то оперирование значениями $0, 1, \ldots, m-1$ представляется более естественным.

Определение 2. *Плотностью* обобщённой задачи о рюкзаке с вектором $A = (a_1, a_2, \ldots, a_r) \in \mathbb{N}^r$ назовём число

$$d_{\text{о.з.р}} = \frac{r}{\log_m \left(\max\limits_{1 \leqslant i \leqslant r} a_i \right)}.$$

Системой задач о рюкзаках назовём конечное множество индивидуальных задач о рюкзаках, имеющих общее решение. Таким образом, система задач о рюкзаках является естественным обобщением индивидуальной задачи о рюкзаке. Кроме того, с точки зрения информационной безопасности система задач о рюкзаках тесно связана со случаем широковещательной рассылки сообщения, например в криптосистеме Меркля — Хеллмана. Дадим строгую формулировку *системы обобщённых задач о рюкзаках.*

Условие: заданы k векторов $A_1 = (a_{11}, \ldots, a_{r1})$, $\ldots$, $A_k = (a_{1k}, \ldots, a_{rk})$ размерности r, k натуральных чисел $S_1, \ldots, S_k$ и натуральное число $m \geqslant 2$, причём известно, что система уравнений

$$\sum_{i=1}^{r} a_{ij} x_i = S_j, \quad 1 \leqslant j \leqslant k, \tag{3}$$

где $x_1, \ldots, x_r$ — неизвестные, имеет решение в числах $0, 1, \ldots, m-1$.

Вопрос: найти решение системы уравнений (3) в числах $0, 1, \ldots, m-1$.

Определение 3. *Плотностью* системы обобщённых задач о рюкзаках с векторами $A_1 = (a_{11}, \ldots, a_{r1})$, $\ldots$, $A_k = (a_{1k}, \ldots, a_{rk})$ назовём число

$$d_{\text{с.о.з.р}} = \frac{r}{\log_m \left(\max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij} \right)}.$$

Обозначим через $S_r(R)$ число точек целочисленной решётки, попавших внутрь или на поверхность r-мерной сферы радиуса $\sqrt{R}$ с центром в начале координат (в r-мерном пространстве). В работе [2], по существу, доказана следующая теорема (непосредственно в работе рассматривался случай $m = 2$), необходимая нам для получения основных результатов.

Теорема 3. Для всех $\alpha > 0$, натуральных чисел $r \geqslant 1$ и любого натурального числа $m \geqslant 2$

$$S_r(\alpha r) \leqslant e^{\gamma} = m^{\gamma \log_m e},$$

где $\gamma = \min\limits_{x \geqslant 0} \delta(\alpha, x) \cdot r$; $\delta(\alpha, x) = \alpha x + \ln \theta(e^{-x})$; $\theta(z) = 1 + 2 \sum\limits_{i=1}^{\infty} z^{i^2}$.

1. Модификация метода Лагариаса — Одлыжко для случая обобщённой задачи о рюкзаке

Рассмотрим решётку Λ_3, образованную следующими векторами $b_1, \ldots, b_{r+1}$ размерности $r+1$:

$$\begin{gathered} b_1 = (1, 0, \ldots, 0, N \cdot a_1), \\ b_2 = (0, 1, \ldots, 0, N \cdot a_2), \\ \vdots \\ b_r = (0, 0, \ldots, 1, N \cdot a_r), \\ b_{r+1} = (0, 0, \ldots, 0, N \cdot S), \end{gathered}$$

где $N \geqslant \left[(m-1)\sqrt{r/2}\right] + 1$ — некоторое достаточно большое натуральное число.

Определение 4. Назовем обобщённую задачу о рюкзаке *приведённой*, если выполнены неравенства

$$\frac{1}{r}\sum_{i=1}^{r} a_i \leqslant S \leqslant (m-1)\sum_{i=1}^{r} a_i - \frac{1}{r}\sum_{i=1}^{r} a_i.$$

Если обобщённая задача о рюкзаке не является приведённой, то имеется возможность исключить из рассмотрения либо самый большой (если $\frac{1}{r}\sum\limits_{i=1}^{r} a_i \geqslant S$), либо самый маленький (если $S \geqslant (m-1)\sum\limits_{i=1}^{r} a_i - \frac{1}{r}\sum\limits_{i=1}^{r} a_i$) элемент вектора. Для приведённой обобщённой задачи о рюкзаке такой возможности нет.

Для любого натурального числа $m \geqslant 2$ справедлива следующая теорема.

Теорема 4. Пусть $a_1, \ldots, a_r$ — произвольные натуральные числа, $\tilde{e} = (e_1, \ldots, e_r) \in \{0, 1, \ldots, m-1\}^r$ — произвольный вектор и $S = \sum\limits_{i=1}^{r} e_i a_i$. Тогда вероятность ошибки при решении приведённой обобщённой задачи о рюкзаке, заданной значениями a_i, $1 \leqslant i \leqslant r$, и S, при условии, что плотность этой задачи $d_{\text{о.з.р}} < \dfrac{\ln m}{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x)}$, с помощью алгоритма «Оракул», применённого к решетке Λ_3, стремится к 0 при $r \to \infty$.

Доказательство. Решётка Λ_3 содержит вектор $e = (e_1, e_2, \ldots, e_r, 0)$, где $e_1, \ldots, e_r$ — решение ассоциированной обобщённой задачи о рюкзаке ($e = \sum\limits_{i=1}^{r} e_i b_i - b_{r+1}$).

Пусть $t = \sum\limits_{i=1}^{r} a_i$. Можно считать, что $\sum\limits_{i=1}^{r} e_i \leqslant (m-1)r/2$, иначе можно перейти к «дополняющей» задаче со следующими параметрами:

$$\begin{gathered} S^{\text{new}} = (m-1)t - S, \\ b_{r+1}^{\text{new}} = (0, 0, \ldots, 0, N((m-1)t - S)), \\ e^{\text{new}} = (m-1-e_1, m-1-e_2, \ldots, m-1-e_r, 0). \end{gathered}$$

Решение «дополняющей» задачи эквивалентно решению исходной задачи, при этом $\sum\limits_{i=1}^{r}(m-1-e_i) \leqslant (m-1)r/2$ (заметим также, что $S^{\text{new}} = (m-1)t - S \geqslant (m-1)t - -(m-1)t + t/r = t/r$).

Поскольку $\sum\limits_{i=1}^{r} e_i \leqslant (m-1)r/2$, $e_i \in \{0, 1, \ldots, m-1\}$ и $h^2 + (m-1-h)^2 \leqslant (m-1)^2$ при $0 \leqslant h \leqslant m-1$, то $||e||^2 = \sum\limits_{i=1}^{r} e_i^2 \leqslant (m-1)^2 r/2$.

Поэтому вектор e является «*достаточно коротким*» вектором решётки и его можно искать при помощи «Оракула». Однако может оказаться, что «Оракул» выдаст вектор, не совпадающий с векторами $-e, e$. В этом случае вектор, предоставленный «Оракулом», является элементом следующего множества ошибочных векторов:

$$X_3 = \{x = (x_1, \ldots, x_{r+1}) : x \in \Lambda_3, ||x|| \leqslant ||e||, x \notin \{-e, 0, e\}\}.$$

Выбор числа $N \geqslant \left[(m-1)\sqrt{r/2}\right] + 1$ обеспечивает для векторов $x \in X_3$ выполнение условия $x_{r+1} = 0$ (иначе если $x \in X_3$ и $x_{r+1} \neq 0$, то $||x|| \geqslant |x_{r+1}| \geqslant N > (m-1)\sqrt{r/2} \geqslant$ $\geqslant ||e||$, то есть $x \notin X_3$ — противоречие).

Рассмотрим произвольный вектор $x \in X_3$ и определим число $y \in \mathbb{Z}$ следующим образом:

$$yS = \sum_{i=1}^{r} x_i a_i,$$

тогда

$$|y|S = \left|\sum_{i=1}^{r} x_i a_i\right| \leqslant ||x|| \sum_{i=1}^{r} a_i \leqslant (m-1)t\sqrt{r/2}$$

и, поскольку $t \leqslant Sr$, получим $|y| \leqslant (m-1)r\sqrt{r/2}$.

Пусть P — вероятность того, что множество X_3 не является пустым, тогда справедливо неравенство

$$P \leqslant \Pr\left(\sum_{i=1}^{r} x_i a_i = yS \mid x \in X_3, |y| \leqslant (m-1)r\sqrt{r/2}\right) \cdot |X_3| \cdot \left|\left\{y : |y| \leqslant (m-1)r\sqrt{r/2}\right\}\right|.$$

Определим вектор $z = (z_1, z_2, \ldots, z_r)$, где $z_i = x_i - ye_i$, $1 \leqslant i \leqslant r$. Случай $z = 0$ не рассматривается, поскольку при этом $||x|| = |y|||e||$, и так как $||x|| \leqslant ||e||$, то $|y| \leqslant 1$ и $x \in \{-e, 0, e\}$, а значит, $x \notin X_3$.

Пусть $B = \max\limits_{1 \leqslant i \leqslant r} a_i$, без ограничения общности будем считать, что $z_1 \neq 0$. Определим

$$z^* = -\sum_{i=2}^{r} \frac{a_i z_i}{z_1},$$

тогда

$$\Pr\left(\sum_{i=1}^{r} z_i a_i = 0\right) = \Pr(a_1 = z^*) = \sum_{j=1}^{B} \Pr(a_1 = z^* \mid z^* = j)\Pr(z^* = j) =$$
$$= \sum_{j=1}^{B} \Pr(a_1 = j)\Pr(z^* = j) = \sum_{j=1}^{B} \frac{1}{B}\Pr(z^* = j) \leqslant \frac{1}{B}.$$

Предыдущая оценка получена в предположении, что $z_1 \neq 0$, но на самом деле каждый элемент z_i при $1 \leqslant i \leqslant r$ может быть не равным нулю, поэтому

$$\Pr\left(\sum_{i=1}^{r} z_i a_i = 0\right) \leqslant \frac{r}{B}.$$

С учётом результата теоремы 3 при $\alpha = (m-1)^2/2$ получим

$$|X_3| \leqslant \left|\left\{\tilde{x} \in \mathbb{Z}^r : ||\tilde{x}|| \leqslant (m-1)\sqrt{r/2}\right\}\right| \leqslant m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) r \log_m e}.$$

По свойствам модуля

$$\left|\left\{y : |y| \leqslant (m-1)r\sqrt{r/2}\right\}\right| \leqslant 2(m-1)r\sqrt{r/2} + 1.$$

Таким образом,

$$P \leqslant r\left(2(m-1)r\sqrt{r/2} + 1\right) \frac{m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) r \log_m e}}{B},$$

и если $B = m^{C_m r}$, то при $C_m > \min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) \log_m e$ получим $\lim\limits_{r \to \infty} P = 0$.

Если плотность обобщённой задачи о рюкзаке

$$d_{\text{о.з.р}} = \frac{r}{\log_m \left(\max\limits_{1 \leqslant i \leqslant r} a_i\right)} < \frac{\ln m}{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x)},$$

то $B = \max\limits_{1 \leqslant i \leqslant r} a_i > m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) r \log_m e}$. ■

2. Модификация метода Лагариаса — Одлыжко для случая систем обобщённых задач о рюкзаках

Рассмотрим решётку Λ_4, образованную следующими векторами $b_1, \ldots, b_{r+1}$ размерности $r + k$:

$$\begin{gathered}
b_1 = (1, 0, \ldots, 0, N \cdot a_{11}, N \cdot a_{12}, \ldots, N \cdot a_{1k}), \\
b_2 = (0, 1, \ldots, 0, N \cdot a_{21}, N \cdot a_{22}, \ldots, N \cdot a_{2k}), \\
\vdots \\
b_r = (0, 0, \ldots, 1, N \cdot a_{r1}, N \cdot a_{r2}, \ldots, N \cdot a_{rk}), \\
b_{r+1} = (0, 0, \ldots, 0, N \cdot S_1, N \cdot S_2, \ldots, N \cdot S_k),
\end{gathered}$$

где $N \geqslant [(m-1)\sqrt{r/2}] + 1$ — некоторое достаточно большое натуральное число.

Определение 5. Назовём систему обобщённых задач о рюкзаках *приведённой*, если для всех $1 \leqslant j \leqslant k$ выполнены неравенства

$$\frac{1}{r}\sum_{i=1}^{r} a_{ij} \leqslant S_j \leqslant (m-1)\sum_{i=1}^{r} a_{ij} - \frac{1}{r}\sum_{i=1}^{r} a_{ij}.$$

Для любого натурального числа $m \geqslant 2$ справедлива следующая теорема.

Теорема 5. Пусть a_{ij}, $1 \leqslant i \leqslant r$, $1 \leqslant j \leqslant k$, — произвольные натуральные числа, причём векторы $A_1 = (a_{11}, \ldots, a_{r1}), \ldots, A_k = (a_{1k}, \ldots, a_{rk})$ являются линейно независимыми. Пусть $\tilde{e} = (e_1, \ldots, e_r) \in \{0, 1, \ldots, m-1\}^r$ — произвольный вектор и $S_j = \sum\limits_{i=1}^{r} e_i a_{ij}$ для всех $1 \leqslant j \leqslant k$. Тогда вероятность ошибки при решении приведённой системы обобщённых задач о рюкзаках, заданной значениями a_{ij}, $1 \leqslant i \leqslant r$, $1 \leqslant j \leqslant k$, и S_j, $1 \leqslant j \leqslant k$, при условии, что плотность этой системы $d_{\text{с.о.з.р}} < \dfrac{k \ln m}{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x)}$, с помощью алгоритма «Оракул», применённого к решётке Λ_4, стремится к 0 при $r \to \infty$.

Доказательство. Решётке Λ_4 принадлежит вектор $e = (e_1, \ldots, e_r, 0, \ldots, 0)$, где $e_1, \ldots, e_r$ — решение ассоциированной системы обобщённых задач о рюкзаках. Отметим, что, как и ранее, можно считать, что $||e||^2 \leqslant (m-1)^2 r/2$.

Множеством ошибочных векторов в этом случае является

$$X_4 = \{x = (x_1, \ldots, x_{r+1}) : x \in \Lambda_4, ||x|| \leqslant ||e||, x \notin \{-e, 0, e\}\}.$$

Выбор числа $N \geqslant \left[(m-1)\sqrt{r/2}\right] + 1$ обеспечивает выполнение для векторов $x \in X_4$ условия $x_{r+j} = 0$ для всех $1 \leqslant j \leqslant k$ (иначе если $x \in X_4$ и $x_{r+j} \neq 0$ для некоторого $1 \leqslant j \leqslant k$, то $||x|| \geqslant |x_{r+j}| \geqslant N > (m-1)\sqrt{r/2} \geqslant ||e||$, то есть $x \notin X_4$ — противоречие).

Пусть $t_j = \sum\limits_{i=1}^{r} a_{ij}$ для всех $1 \leqslant j \leqslant k$. Рассмотрим произвольный вектор $x \in X_4$ и определим число $y \in \mathbb{Z}$ следующим образом:

$$yS_j = \sum_{i=1}^{r} x_i a_{ij} \text{ для всех } 1 \leqslant j \leqslant k,$$

тогда для всех $1 \leqslant j \leqslant k$

$$|y| S_j = \left| \sum_{i=1}^{r} x_i a_{ij} \right| \leqslant ||x|| \sum_{i=1}^{r} a_{ij} \leqslant (m-1) t_j \sqrt{r/2},$$

и поскольку $t_j \leqslant S_j r$, получим

$$|y| \leqslant (m-1) r \sqrt{r/2};$$

более точную оценку может дать неравенство

$$|y| \leqslant (m-1) \min_{1 \leqslant j \leqslant k} \{t_j / S_j\} \sqrt{r/2}.$$

Пусть P — вероятность того, что множество X_4 не является пустым, тогда справедливо неравенство

$$P \leqslant \Pr\left(\sum_{i=1}^{r} x_i a_{i1} = yS_1, \ldots, \sum_{i=1}^{r} x_i a_{ik} = yS_k \mid x \in X_4, |y| \leqslant (m-1) r \sqrt{r/2} \right) \cdot |X_4| \times$$
$$\times |\{y : |y| \leqslant (m-1) r \sqrt{r/2}\}| = \prod_{j=1}^{k} \Pr\left(\sum_{i=1}^{r} x_i a_{ij} = yS_j \mid x \in X_4, |y| \leqslant (m-1) r \sqrt{r/2} \right) \times$$
$$\times |X_4| \cdot |\{y : |y| \leqslant (m-1) r \sqrt{r/2}\}|.$$

Пусть $B = \max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij}$. Как и прежде, можно показать, что для всех $1 \leqslant j \leqslant k$

$$\Pr\left(\sum_{i=1}^{r} x_i a_{ij} = yS_j \right) \leqslant \frac{r}{B},$$
$$\left| \left\{ y : |y| \leqslant (m-1) r \sqrt{r/2} \right\} \right| \leqslant 2(m-1) r \sqrt{r/2} + 1,$$
$$|X_4| \leqslant |\{\tilde{x} \in \mathbb{Z}^r : ||\tilde{x}|| \leqslant (m-1)\sqrt{r/2}\}| \leqslant m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) r \log_m e}.$$

Таким образом, в этом случае

$$P \leqslant r^k \left(2(m-1) r \sqrt{r/2} + 1 \right) \frac{m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) r \log_m e}}{B^k},$$

и если $B = m^{C_m r/k}$, то при $C_m > \min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) \log_m e$ получим $\lim\limits_{r \to \infty} P = 0$.

Если плотность системы обобщённых задач о рюкзаках

$$d_{\text{с.о.з.р}} = \frac{r}{\log_m \left(\max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij} \right)} < \frac{k \ln m}{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x)},$$

то $B = \max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij} > m^{\min\limits_{x \geqslant 0} \delta((m-1)^2/2, x) \frac{r}{k} \log_m e}$. ■

При $m = 2$ будем называть систему обобщённых задач о рюкзаках системой задач о рюкзаках. Далее в теореме 6 определяется множество значений плотности систем задач о рюкзаках, такое, что алгоритм «Оракул» позволяет решать «практически все» системы задач о рюкзаках, обладающих плотностью из этого множества.

Рассмотрим решётку Λ_5, образованную следующими векторами $b_1, \ldots, b_{r+1}$ размерности $r + k$:

$$\begin{gathered}
b_1 = (1, 0, \ldots, 0, N \cdot a_{11}, N \cdot a_{12}, \ldots, N \cdot a_{1k}), \\
b_2 = (0, 1, \ldots, 0, N \cdot a_{21}, N \cdot a_{22}, \ldots, N \cdot a_{2k}), \\
\vdots \\
b_r = (0, 0, \ldots, 1, N \cdot a_{r1}, N \cdot a_{r2}, \ldots, N \cdot a_{rk}), \\
b_{r+1} = \left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}, N \cdot S_1, N \cdot S_2, \ldots, N \cdot S_k \right),
\end{gathered}$$

где $N \geqslant [\sqrt{r}/2] + 1$ — некоторое достаточно большое натуральное число.

Теорема 6. Пусть a_{ij}, $1 \leqslant i \leqslant r$, $1 \leqslant j \leqslant k$, — произвольные натуральные числа, причём векторы $A_1 = (a_{11}, \ldots, a_{r1}), \ldots, A_k = (a_{1k}, \ldots, a_{rk})$ являются линейно независимыми. Пусть $\tilde{e} = (e_1, \ldots, e_r) \in \{0, 1\}^r$ — произвольный вектор и $S_j = \sum\limits_{i=1}^{r} e_i a_{ij}$ для всех $1 \leqslant j \leqslant k$. Тогда вероятность ошибки при решении системы задач о рюкзаках, заданной значениями a_{ij}, $1 \leqslant i \leqslant r$, $1 \leqslant j \leqslant k$, и S_j, $1 \leqslant j \leqslant k$, при условии, что плотность этой системы $d_{\text{с.з.р}} < k \cdot 0{,}9408\ldots$, с помощью алгоритма «Оракул», применённого к решётке Λ_5, стремится к 0 при $r \to \infty$.

Доказательство. Решётке Λ_5 принадлежит вектор $e^* = (e_1^*, \ldots, e_r^*, 0, \ldots, 0)$, где $e_i^* = e_i - 1/2$, а $e_1, \ldots, e_r$ — решение ассоциированной системы задач о рюкзаках ($e^* = \sum\limits_{i=1}^{r} e_i b_i - b_{r+1}$). Заметим, что $||e^*||^2 = r/4$.

Множеством ошибочных векторов в этом случае является

$$X_5 = \{x = (x_1, \ldots, x_{r+1}) : x \in \Lambda_5, ||x|| \leqslant ||e^*||, x \notin \{-e^*, 0, e^*\}\}.$$

Выбор числа $N \geqslant [\sqrt{r}/2] + 1$ обеспечивает выполнение для векторов $x \in X_5$ условия $x_{r+j} = 0$ для всех $1 \leqslant j \leqslant k$ (иначе если $x \in X_5$ и $x_{r+j} \neq 0$ для некоторого $1 \leqslant j \leqslant k$, то $||x|| \geqslant |x_{r+j}| \geqslant N > \sqrt{r}/2 \geqslant ||e^*||$, то есть $x \notin X_5$ — противоречие).

Поскольку для векторов множества X_5, являющихся, по определению, векторами решётки Λ_5, выполнено

$$x = \sum_{i=1}^{r} y_i b_i + y b_{r+1}$$

для некоторых целых y и y_i, $1 \leqslant i \leqslant r$, то $x_i = y_i + y/2$ для всех $1 \leqslant i \leqslant r$ и $x_{r+j} = N\left(\sum\limits_{i=1}^{r} a_{ij}y_i + yS_j\right)$ для всех $1 \leqslant j \leqslant k$.

Пусть $t_j = \sum\limits_{i=1}^{r} a_{ij}$ для всех $1 \leqslant j \leqslant k$. Оценим значение $|y|$. Так как $x_{r+j} = 0$ для всех $1 \leqslant j \leqslant k$, то $\sum\limits_{i=1}^{r} a_{ij}y_i = -yS_j$ для всех $1 \leqslant j \leqslant k$, и, подставляя значения $y_i = x_i - y/2$, получим $\sum\limits_{i=1}^{r}(x_i - y/2)a_{ij} = -yS_j$ и $\sum\limits_{i=1}^{r} x_i a_{ij} = \dfrac{1}{2}y\sum\limits_{i=1}^{r} a_{ij} - yS_j = \dfrac{1}{2}yt_j - yS_j = \dfrac{1}{2}y(t_j - 2S_j)$, поэтому для всех $1 \leqslant j \leqslant k$ выполнено

$$|y(t_j - 2S_j)| = 2\left|\sum_{i=1}^{r} x_i a_{ij}\right| \leqslant 2||x||\sum_{i=1}^{r} a_{ij} \leqslant \max_{1\leqslant i\leqslant r} a_{ij} r\sqrt{r}.$$

Пусть числа $1 \leqslant I \leqslant r$ и $1 \leqslant J \leqslant k$ являются наименьшими индексами, такими, что $a_{IJ} = \max\limits_{1\leqslant i\leqslant r, 1\leqslant j\leqslant k} a_{ij}$, тогда можно считать, что $|t_J - 2S_J| \geqslant a_{IJ}/2$. Иначе (если $|t_J - 2S_J| < a_{IJ}/2$) возможны только два случая.

В первом случае $e_I = 1$, и, исключив из рассмотрения для всех $1 \leqslant j \leqslant k$ элементы a_{Ij}, перейдём к системе задач о рюкзаках, в которой для всех $1 \leqslant j \leqslant k$

$$S_j^{\text{new}} = S_j - a_{Ij}, \qquad t_j^{\text{new}} = t_j - a_{Ij},$$
$$|t_J^{\text{new}} - 2S_J^{\text{new}}| = |t_J - a_{IJ} - 2S_J + 2a_{IJ}| = |t_J - 2S_J + a_{IJ}| \geqslant \frac{1}{2}a_{IJ}.$$

Во втором случае $e_I = 0$, и, исключив из рассмотрения для всех $1 \leqslant j \leqslant k$ элементы a_{Ij}, перейдём к системе задач о рюкзаках, в которой для всех $1 \leqslant j \leqslant k$

$$S_j^{\text{new}} = S_j, \qquad t_j^{\text{new}} = t_j - a_{Ij},$$
$$|t_J^{\text{new}} - 2S_J^{\text{new}}| = |t_J - 2S_J - a_{IJ}| \geqslant \frac{1}{2}a_{IJ}.$$

Теперь, поскольку $|t_J - 2S_J| \geqslant \dfrac{1}{2}a_{IJ}$,

$$a_{IJ}r\sqrt{r} \geqslant |y(t_J - 2S_J)| = |y||t_J - 2S_J| \geqslant \frac{1}{2}a_{IJ}|y|.$$

Таким образом, $|y| \leqslant 2r\sqrt{r}$.

Пусть P — вероятность того, что множество X_5 не является пустым, тогда справедливо следующее неравенство:

$$P \leqslant \Pr\left(\sum_{i=1}^{r} x_i a_{i1} = \frac{1}{2}y(t_1 - 2S_1), \ldots, \sum_{i=1}^{r} x_i a_{ik} = \frac{1}{2}y(t_k - 2S_k) \mid x \in X_5, |y| \leqslant 2r\sqrt{r}\right) \times$$
$$\times|\{x : ||x|| \leqslant ||e^*||\}| \cdot |\{y : |y| \leqslant 2r\sqrt{r}\}| =$$
$$= \prod_{j=1}^{k} \Pr\left(\sum_{i=1}^{r} x_i a_{ij} = \frac{1}{2}y(t_j - 2S_j) \mid x \in X_5, |y| \leqslant 2r\sqrt{r}\right) \times$$
$$\times|\{x : ||x|| \leqslant ||e^*||\}| \cdot |\{y : |y| \leqslant 2r\sqrt{r}\}|.$$

По построению векторов $b_1, \ldots, b_{r+1}$ ввиду того, что $x = \sum\limits_{i=1}^{r} y_i b_i + yb_{r+1}$ и $x_{r+1} = 0$, имеет место

$$|\{x : ||x|| \leqslant \sqrt{r}/2\}| \leqslant |\{\tilde{x} \in \mathbb{Z}^r : ||\tilde{x}|| \leqslant \sqrt{r}/2\}| + |\{\tilde{x} \in \mathbb{Z}^r : ||\tilde{x} - (1/2, \ldots, 1/2)|| \leqslant \sqrt{r}/2\}|.$$

В последней сумме первое слагаемое соответствует случаям с чётными y, а второе — с нечётными.

Далее, $|\{\tilde{x} : \tilde{x} \in \mathbb{Z}^r, ||\tilde{x} - (1/2, \ldots, 1/2)|| \leqslant \sqrt{r}/2\}| = 2^r$. С учётом результата теоремы 3, при $\alpha = 1/4$ получим

$$|\{\tilde{x} \in \mathbb{Z}^r : ||\tilde{x}|| \leqslant \sqrt{r}/2\}| \leqslant 2^{\min\limits_{x \geqslant 0} \delta(1/4, x) r \log_2 e}.$$

Численно устанавливается, что $\min\limits_{x \geqslant 0} \delta(1/4, x) \log_2 e = 1{,}0628\ldots$ Следовательно, можно утверждать, что

$$|\{x : ||x|| \leqslant \sqrt{r}/2\}| \leqslant 2^{C_1 r} + 2^r,$$

где $C_1 = 1{,}0628\ldots$

Пусть $B = \max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij}$. Как и прежде, можно показать, что для всех $1 \leqslant j \leqslant k$

$$\Pr\left(\sum_{i=1}^{r} x_i a_{ij} = \frac{1}{2} y(t_j - 2S_j)\right) \leqslant \frac{r}{B}, \quad |\{y : |y| \leqslant 2r\sqrt{r}\}| \leqslant 4r\sqrt{r} + 1.$$

Таким образом, в этом случае

$$P \leqslant r^k (4r\sqrt{r} + 1) \frac{2^{C_1 r} + 2^r}{B^k},$$

и если $B = 2^{Cr/k}$, то при $C > C_1$ получим, что $\lim\limits_{r \to \infty} P = 0$.

Если плотность системы задачи о рюкзаках

$$d_{\text{с.з.р}} = \frac{r}{\log_2 \left(\max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij} \right)} < k \cdot 0{,}9408\ldots,$$

то $B = \max\limits_{1 \leqslant i \leqslant r, 1 \leqslant j \leqslant k} a_{ij} > 2^{C_1 r/k}$. ■

ЛИТЕРАТУРА

1. *Karp R. M.* Reducibility among combinatorial problems // Complexity of Computer Computations. Plenum Press, 1972. P. 85–103.
2. *Odlyzko A. M. and Lagarias J. C.* Solving low-density subset sum problems // J. Association for Computing Machinery. 1985. V. 32. No. 1. P. 229–246.
3. *Brickell E. F.* Solving low-density knapsacks // Advances in Cryptology. Proc. Crypto'83. Plenun Press, 1984. P. 25–37.
4. *Boas P.* Another NP-complete problem and the complexity of computing short vectors in a lattice // Tech. rep. 8104, University of Amsterdam, Department of Mathematics, Netherlands, 1981. 10 p.
5. *Coster M. J., Joux A., LaMacchia B. A., et al.* Improved low-density subset sum algorithms // Computational Complexity. 1992. No. 2. P. 111–128.

2013 Вычислительные методы в дискретной математике №2(20)

DOI 10.17223/20710410/20/11

УДК 510.25+510.52+519.688

ЭФФЕКТИВНОЕ ПО ВРЕМЕНИ И ПАМЯТИ ВЫЧИСЛЕНИЕ ЛОГАРИФМИЧЕСКОЙ ФУНКЦИИ ВЕЩЕСТВЕННОГО АРГУМЕНТА НА МАШИНЕ ШЁНХАГЕ

С. В. Яхонтов

Санкт-Петербургский государственный университет, г. Санкт-Петербург, Россия

E-mail: SergeyV.Yakhontov@gmail.com

Строится алгоритм *FLE* быстрого вычисления логарифмической функции $\ln(1+x)$ вещественного аргумента на полуинтервале $\left[2^{-5}, 1-2^{-5}\right)$ на машине Шёнхаге с оракульной функцией и даётся верхняя оценка его временной и емкостной сложности. Алгоритм *FLE* строится на основе разложения в ряд Тейлора по аналогии с алгоритмом быстрого вычисления экспоненты *FEE*, при этом дополнительно строится модифицированный алгоритм двоичного деления *ModifBinSplit* для гипергеометрических рядов. Для алгоритмов *ModifBinSplit* и *FLE* показывается квазилинейность по времени и линейность по памяти при вычислении на машине Шёнхаге, то есть принадлежность классу Sch(FQLIN-TIME//LIN-SPACE). Для расчёта логарифмической функции на произвольном промежутке используется мультипликативная редукция интервала.

Ключевые слова: *логарифмическая функция, алгоритмические вещественные функции, квазилинейная временная сложность, линейная ёмкостная сложность.*

Введение

Данная работа является продолжением работ [1 – 3], посвящённых построению алгоритмических аналогов констант и элементарных функций с ограниченной сложностью вычисления двоично-рациональных приближений при вычислении на машине Шёнхаге [4].

Приводится мера временной и емкостной сложности [1 – 4] вычислений на машине Шёнхаге и даётся верхняя оценка временной и емкостной сложности предлагаемого алгоритма вычисления логарифмической функции $\ln(1+x)$ вещественного аргумента на полуинтервале $[2^{-5}, 1-2^{-5})$ на машине Шёнхаге.

Основные сведения о двоично-рациональных числах и алгоритмических числах и функциях можно найти в [5]. Посредством Sch(FQLIN-TIME//LINS-PACE) будем обозначать, как и в [1 – 3], класс алгоритмов, квазилинейных по времени и линейных по памяти при вычислении на машине Шёнхаге. Под квазилинейностью понимается ограниченность сверху функцией вида $\mathrm{O}(n\log(n)^k)$ при некотором k.

Основной предмет нашего интереса — это алгоритмы для расчёта элементарных функций, основанные на разложениях в ряды, так как такие алгоритмы важны в практической информатике в силу своей относительной простоты реализации.

Известно, что многие вещественные элементарные функции, с одной стороны, вычислимы с помощью разложения в ряд полиномиальными по времени и линейными по памяти алгоритмами [3] и, с другой — квазилинейными по времени и квазилинейными по памяти алгоритмами [6 – 8]. В данных работах рассматривается сложность

алгоритмов при реализации на машине Тьюринга [3] и в рамках модели битовых вычислений [6 – 8], но оценки вычислительной сложности, приведённые в них, верны и для машины Шёнхаге.

Возникает вопрос: нельзя ли рассчитывать элементарные функции алгоритмами, основанными на разложениях в ряды, одновременно квазилинейными по времени и линейными по памяти на машине Шёнхаге?

В [2, 3] даётся положительный ответ на данный вопрос для экспоненциальной функции комплексного аргумента и некоторых других элементарных функций, которые выражаются через экспоненциальную функцию комплексного аргумента с помощью простых соотношений. Квазилинейные по времени и линейные по памяти алгоритмы из [2, 3] для вычисления приближённых значений экспоненциальной функции на машине Шёнхаге основаны на модифицированном методе двоичного деления для гипергеометрических рядов [9] и модифицированном методе быстрого вычисления экспоненты [6, 7].

Метод двоичного деления (англ. binary splitting) [9], предназначенный для вычисления гипергеометрических рядов, является рекурсивным вариантом метода FEE [6 – 8, 10 – 13] вычисления гипергеометрических рядов: в методе из [9] дерево вычислений обходится сверху вниз с помощью рекурсии, в отличие от итеративного метода FEE, в котором, в частности, дерево вычислений обходится снизу вверх. Будем использовать рекурсивный метод из [9]; это связано с удобством его реализации на машине Шёнхаге, в которой есть рекурсивные вызовы. К тому же при использовании рекурсии как сам алгоритм, так и оценки его вычислительной сложности получаются проще.

В данной работе подход, использованный в [2, 3] для экспоненциальной функции комплексного аргумента, применяется для логарифмической функции вещественного аргумента: алгоритм класса Sch(FQLIN-TIME//LIN-SPACE) вычисления логарифмической функции на машине Шёнхаге основан на комбинации алгоритма расчёта гипергеометрических рядов из [1 – 3] и алгоритма быстрого вычисления логарифма, который строится для логарифмической функции по схеме метода быстрого вычисления экспоненты.

Везде далее под функцией $\log(k)$ будем понимать логарифм по основанию 2; через $M_{\mathrm{Sch}}(n)$ обозначим верхнюю оценку временной сложности алгоритма Шёнхаге — Штрассена [4] умножения целых чисел на машине Шёнхаге.

1. Описание вычислительной модели (машины Шёнхаге)

Данная машина, введенная в [4], оперирует символами из алфавита $\Sigma = \{0, 1, \ldots, 2^\lambda - 1\}$ и с последовательностями таких символов, где λ — некоторая константа (данную константу можно взять, например, равной 32). Машина состоит из одномерных массивов T_0, …, T_τ для чтения и записи символов из Σ, регистров A, B, C, M для арифметических операций над символами (которые интерпретируются в данном случае как натуральные числа) и управляющего устройства CPU. Массивы потенциально бесконечны в обе стороны; для каждого массива есть указатель p_i на текущий символ, записанный в массиве. Запись $\langle p + j \rangle$ означает символ, на который ссылается указатель $p + j$. Есть также дополнительный регистр Y — указатель на текущую выполняемую инструкцию, и дополнительный массив S, потенциально бесконечный в одну сторону, который служит в качестве стека рекурсивных вызовов. Битовый регистр E служит как регистр переполнения при арифметических операциях.

Программа для машины Шёнхаге состоит из нескольких модулей-процедур на языке TPAL, который аналогичен языку ассемблера для RISC-процессоров. В TPAL имеются команды загрузки в регистр символа, записанного в массиве, чтения из регистра и запись в массив, арифметические операции, команды увеличения/уменьшения содержимого регистров, команды сдвига, вызов и возврат из процедуры, переход по метке и условный переход. Целые числа, которыми оперирует машина, кодируются как последовательности символов в алфавите Σ:

$$a = a_0 + a_1 2^\lambda + a_2 (2^\lambda)^2 + \ldots + a_{k-1}(2^\lambda)^{k-1}, \quad 0 \leqslant a_i \leqslant 2^\lambda - 1.$$

Бит знака размещается в символе, следующем за старшим символом a_{k-1}.

При вызове процедур параметры записываются в массивах, локальные переменные процедур и возвращаемые значения — также в массивах. При возврате из процедуры память, занятая под параметры и локальные переменные, освобождается.

Временная вычислительная сложность алгоритма на машине Шёнхаге определяется как количество выполняемых на ней инструкций на языке TPAL. При этом затраты на арифметические операции над символами и на вызов процедуры учитываются как некоторое константное число шагов [4]. Память, используемую программой при вычислении в массиве, определим как максимум из количества битов по всем участвующим в вычислениях элементам массива.

Поскольку конструктивные функции — это функции, вычисляющие приближения своих значений по приближениям аргумента, определим оракульную машину Шёнхаге. Данная машина имеет оракульную функцию, которая вычисляет приближения аргумента; по таким приближениям машина вычисляет приближения функции. Условимся, что запрос к оракулу в виде записи точности вычисления записывается в массиве T_0, в котором также дается приближение аргумента. При оценке временной вычислительной сложности на оракульной машине Шёнхаге запрос к оракулу учитывается как одна операция.

Определение 1 [2, 3]. *Емкостную вычислительную сложность алгоритма* при расчёте на оракульной машине Шёнхаге определим как сумму длин используемой памяти по всем массивам, сложенную с максимумом объема памяти, занятой стеком.

Отметим, что машина Шёнхаге существенно отличается от машины Тьюринга, РАМ и РАСП [14] возможностью использования рекурсивных процедур. Поэтому из верхних оценок вычислительной сложности (временной и емкостной) для машины Шёнхаге непосредственно не следуют какие-либо верхние оценки сложности для реализации тех же алгоритмов на машинах Тьюринга, РАМ или РАСП.

В то же время машина Шёнхаге может рассматриваться как паскалевидная функция с теми же верхними оценками сложности. Из основного вывода в [15] и результатов данной работы получаем, что имеется алгоритм вычисления логарифмической функции вещественного аргумента, принадлежащий FP.

2. Алгоритмические вещественные числа и функции

В данной работе основу представления конструктивных объектов составляет понятие алгоритмической последовательности φ, сходящейся по Коши [5], при этом в качестве вычислительной модели берётся машина Шёнхаге. Такая последовательность определяется на множестве всех натуральных чисел $\mathbb{N}$, включая 0, а областью аппроксимирующих значений является всюду плотное в $\mathbb{R}$ естественное подмножество

множества рациональных чисел. Для последовательности, сходящейся по Коши и задающей вещественное число x, требуют, чтобы выполнялось $|\varphi(n) - x| \leqslant 2^{-n}$ для любого натурального n.

В качестве множества аппроксимирующих значений берётся множество двоично-рациональных чисел $\mathbb{D}$ [5]. Рациональное число d называется двоично-рациональным, если $d = m/2^n$ для некоторого целого m и натурального n. Двоично-рациональные числа имеют конечное двоичное представление: строка s, равная $\pm u_p u_{p-1} \dots u_0 . v_1 v_2 \dots v_r$, обозначает число

$$d = \pm \left(\sum_{i=0}^{p} u_i 2^i + m \sum_{j=1}^{r} v_j 2^{-j} \right).$$

Длина представления двоично-рационального числа определяется как количество символов в строке s, равное, с учётом знака и двоичной точки, $p+r+3$, и обозначается $l(s)$. Под точностью представления $\mathrm{prec}(s)$ понимается число битов справа от двоичной точки, то есть r. С точки зрения изучения вычислительной сложности двоично-рациональные числа удобны тем, что для любого n двоично-рациональные числа с точностью n равномерно распределены на вещественной прямой [5].

Последовательность $\varphi : \mathbb{N} \to \mathbb{D}$ двоично-рационально сходится к вещественному числу x, если для любого $n \in \mathbb{N}$ выполняется $\mathrm{prec}(\varphi(n)) = n + 1$ и $|\varphi(n) - x| \leqslant 2^{-n}$. Множество всех функций, двоично-рационально сходящихся к вещественному числу x, обозначается CF_x. Вещественное число x называется CF-алгоритмическим [5], если CF_x содержит вычислимую функцию φ.

Вещественная функция f, заданная на отрезке $[a, b]$ (или на любом другом промежутке), называется алгоритмической функцией [5] на этом отрезке, если существует машина Шёнхаге M с оракульной функцией, такая, что для любого $x \in [a, b]$ и любой вычислимой функции $\varphi \in CF_x$ функция ψ, вычисляемая M с оракульной функцией φ, принадлежит $CF_{f(x)}$. Фактически, это означает, что для любой вычислимой функции $\varphi \in CF_x$ и любого $n \in \mathbb{N}$ машина M последовательно вычисляет $m \in \mathbb{N}$ и $d \in \mathbb{D}$, такие, что $|\varphi(m) - x| \leqslant 2^{-m}$, $|d - f(x)| \leqslant 2^{-n}$.

Сложность расчёта двоично-рациональных приближений алгоритмических чисел и функций определяется в [5] на основе длины двоичного представления точности вычисления. Память ленты запроса и ленты ответа оракульной функции при оценке емкостной вычислительной сложности алгоритма не учитывается. Обращение к оракульной функции $\varphi \in CF_x$ аргумента x алгоритмической функции осуществляется следующим образом:

— на ленту запроса оракульной функции записывается точность вычисления аргумента 2^{-m} в виде 0^m (унарная запись);
— рассчитывается значение $\varphi(m)$ оракульной функции и результат записывается на ленту ответа;
— значение $\varphi(m)$ считывается с ленты ответа в промежуточную память.

При описании алгоритмов данные шаги в явном виде здесь приводиться не будут; просто подразумеваем, что процесс вычислений содержит обращения к оракульной функции.

Определение 2 [1–3]. Число $x \in \mathbb{R}$ назовём Sch(FQLIN-TIME//LIN-SPACE)-*алгоритмическим вещественным числом*, если существует функция $\varphi \in CF_x$, вычислимая в пределах Sch(FQLIN-TIME//LIN-SPACE).

Определение 3 [2, 3]. Вещественную функцию f, заданную на отрезке $[a, b]$, назовём Sch(FQLIN-TIME//LIN-SPACE)-*алгоритмической вещественной функцией на отрезке* $[a, b]$, если для любого $x \in [a, b]$ существует Sch(FQLIN-TIME//LIN-SPACE)-вычислимая функция ψ из $CF_{f(x)}$.

Множества Sch(FQLIN-TIME//LIN-SPACE) алгоритмических вещественных чисел и функций будем обозначать Sch(FQLIN-TIME//LIN-SPACE)$_{CF}$ и Sch(FQ-LIN-TIME//LIN-SPACE)$_{C[a,b]}$ соответственно. Здесь индекс $C[a, b]$ обусловлен тем, что алгоритмические функции являются непрерывными на всей области определения [5]. Построение алгоритмического аналога вещественной функции f на отрезке $[a, b]$ означает описание алгоритма, вычисляющего двоично-рациональные приближения с произвольной точностью значений $f(x)$ для $x \in [a, b]$.

Аналогичные определения можно ввести и для любых промежутков из области определения функции.

Под приближением t с точностью 2^{-n} будем понимать двоично-рациональное число t^*, такое, что $|t^* - t| \leqslant 2^{-n}$. Для модуля t^* выполняются неравенства

$$|t^*| - |t| \leqslant |t^* - t| \leqslant 2^{-n}, \quad \text{то есть} \quad |t^*| \leqslant |t| + 2^{-n},$$
$$|t| - |t^*| \leqslant |t - t^*| \leqslant 2^{-n}, \quad \text{то есть} \quad |t^*| \geqslant |t| - 2^{-n}.$$

Далее, пусть двоично-рациональное число t^{**} таково, что $|t^{**} - t| \leqslant 2^{-(n+1)}$. Отбросим все биты t^{**} после двоичной точки, начиная с $(n+2)$-го, а новую величину обозначим t^*. Тогда $\mathrm{prec}(t^*) = n + 1$ и $|\varepsilon| = |t^* - t^{**}| < 2^{-(n+1)}$. Отсюда получаем

$$|t^* - t| \leqslant |t^* - t^{**}| + |t^{**} - t| < 2^{-(n+1)} + 2^{-(n+1)} = 2^{-n}.$$

То есть чтобы вычислить t с точностью 2^{-n}, можно рассчитать t с точностью $2^{-(n+1)}$, а затем отбросить биты полученной величины t^{**} после двоичной точки, начиная с $(n + 2)$-го.

3. Метод двоичного деления

Данный метод предназначен для вычисления значений гипергеометрических рядов частного вида с рациональными коэффициентами

$$S = \sum_{i=0}^{\infty} \frac{a(i)}{b(i)} \prod_{j=0}^{i} \frac{p(j)}{q(j)}, \tag{1}$$

где a, b, p, q — полиномы с целыми коэффициентами. Линейно сходящиеся гипергеометрические ряды используются для расчёта многих констант математического анализа и элементарных функций в рациональных точках. Ряд (1) линейно сходится, если его частичная сумма

$$S(\mu(k)) = \sum_{i=0}^{\mu(k)} \frac{a(i)}{b(i)} \prod_{j=0}^{i} \frac{p(j)}{q(j)}, \tag{2}$$

где $\mu(k)$ — линейная функция от k, отличается от точного значения не более чем на 2^{-k}:

$$|S - S(\mu(k))| \leqslant 2^{-k}.$$

В классическом варианте метод двоичного деления состоит в следующем. Обозначим $k_1 = \mu(k)$. Рассмотрим частичную сумму (2) для некоторых чисел i_1 и i_2,

$0 \leqslant i_1 \leqslant k_1$, $0 \leqslant i_2 \leqslant k_1$, $i_1 \leqslant i_2$:

$$S(i_1, i_2) = \sum_{i=i_1}^{i_2} \frac{a(i)p(i_1)\dots p(i)}{b(i)q(i_1)\dots q(i)}.$$

Будем определять величины

$$P(i_1, i_2) = p(i_1)\dots p(i_2), \quad Q(i_1, i_2) = q(i_1)\dots q(i_2), \quad B(i_1, i_2) = b(i_1)\dots b(i_2),$$
$$T(i_1, i_2) = B(i_1, i_2)Q(i_1, i_2)S(i_1, i_2).$$

Если $i_1 = i_2$, то они вычисляются напрямую. Иначе ряд делится на две части, левую и правую, и $P(i_1, i_2)$, $Q(i_1, i_2)$, $B(i_1, i_2)$ рассчитываются для каждой из частей рекурсивно. Затем полученные величины комбинируются:

$$P(i_1, i_2) = P_l P_r, \quad Q(i_1, i_2) = Q_l Q_r, \quad B(i_1, i_2) = B_l B_r, \quad T(i_1, i_2) = B_r Q_r T_l + B_l P_l T_r.$$

Алгоритм начинает свою работу с $i_1 = 0$, $i_2 = k_1$. После вычисления $T(0, k_1)$, $B(0, k_1)$, $Q(0, k_1)$ осуществляется деление $T(0, k_1)$ на $B(0, k_1)Q(0, k_1)$, чтобы получить результат с заданной точностью. Выпишем алгоритм двоичного деления в явном виде (алгоритм 1).

Алгоритм 1. *BinSplit*. Приближённое значение частичной суммы (2) с точностью 2^{-k}

Вход: Запись 0^k точности вычисления 2^{-k}
Выход: Кортеж $[P(i_1, i_2), Q(i_1, i_2), B(i_1, i_2), T(i_1, i_2)]$
1: $k_1 := \mu(k)$;
2: $[P, Q, B, T] := BinSplitRecurs(0, k_1)$;
3: осуществляем деление $r := \dfrac{T}{BQ}$ с точностью 2^{-k};
4: возвращаем результат r.

В алгоритме 1 используется подалгоритм *BinSplitRecurs* (алгоритм 2), рассчитывающий рекурсивно P, Q, B, T.

Алгоритм 2. *BinSplitRecurs*. Вычисление величин P, Q, B, T

Вход: Границы отрезка i_1, i_2
Выход: Кортеж $[P(i_1, i_2), Q(i_1, i_2), B(i_1, i_2), T(i_1, i_2)]$
1: **Если** $i_1 = i_2$, **то**
2: $\quad T := a(i_1)p(i_1)$ и возвращаем кортеж $[p(i_1), q(i_1), b(i_1), T]$;
3: $i_{\text{mid}} := \left\lfloor \dfrac{i_1 + i_2}{2} \right\rfloor$;
4: вычисляем $[P_l, Q_l, B_l, T_l] := BinSplitRecurs(i_1, i_{\text{mid}})$;
5: вычисляем $[P_r, Q_r, B_r, T_r] := BinSplitRecurs(i_{\text{mid}}, i_2)$;
6: $T := B_r Q_r T_l + B_l P_l T_r$ и возвращаем кортеж $[P_l P_r, Q_l Q_r, B_l B_r, T]$.

Длины чисел $T(0, k_1)$ и $B(0, k_1)Q(0, k_1)$ пропорциональны $k \log(k)$, то есть алгоритм двоичного деления является квазилинейным по памяти; его временная сложность — $\mathrm{O}(M_{\text{Sch}}(k) \log(k)^2)$ [6].

4. Метод быстрого вычисления $\exp(x)$

Рассмотрим ряд Тейлора вещественной экспоненциальной функции [16]

$$\exp(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + r_n(x). \tag{3}$$

Будем вычислять значение этого ряда с точностью 2^{-n} в двоично-рациональной точке x_m, $|x_m - x_0| \leqslant 2^{-m}$, $-1/4 < x_m < 1/4$. Пусть k — наименьшее число, такое, что $n+1 \leqslant 2^k$, и $m = 2^{k+1}$. Представим $x_m = \pm 0{,}00\alpha_3\alpha_4 \ldots \alpha_m\alpha_{m+1}$ в виде

$$\begin{aligned} x_m &= \pm 0{,}00\alpha_3\alpha_4 + \pm 0{,}0000\alpha_5\alpha_6\alpha_7\alpha_8 + \ldots + \\ &\quad + \pm 0{,}00\ldots 0a_{m-2^k+1}a_{m-2^k+2}\ldots\alpha_m\alpha_{m+1} = \\ &= \frac{\beta_2}{2^4} + \frac{\beta_3}{2^8} + \frac{\beta_4}{2^{16}} + \ldots + \frac{\beta_{k+1}}{2^m} = \gamma_2 + \gamma_3 + \ldots + \gamma_{k+1}, \end{aligned}$$

где $\beta_2 = \pm\alpha_3\alpha_4$; $\beta_3 = \pm\alpha_5\alpha_6\alpha_7\alpha_8$; …; $\beta_{k+1} = \pm a_{m-2^k+1}a_{m-2^k+2}\ldots\alpha_m\alpha_{m+1}$; $\gamma_\varsigma = \beta_\varsigma 2^{-2^\varsigma}$, $2 \leqslant \varsigma \leqslant k+1$; β_ς — $2^{\varsigma-1}$-значное целое число. Значение $\exp(x_m)$ запишем как произведение

$$\exp(x_m) = \exp(\gamma_2)\exp(\gamma_3)\cdot\ldots\cdot\exp(\gamma_{k+1}).$$

В методе быстрого вычисления экспоненты (методе Карацубы; англ. FEE [7]) величины $\exp(\gamma_\varsigma)$ далее рассчитываются с помощью ряда Тейлора (3):

$$\exp(\gamma_\varsigma) = 1 + \frac{\beta_\varsigma}{1!2^{2^\varsigma}} + \frac{\beta_\varsigma^2}{2!2^{2^\varsigma 2}} + \ldots + \frac{\beta_\varsigma^r}{r!2^{2^\varsigma r}} + R_\varsigma(r) = \xi_\varsigma + R_\varsigma(r). \tag{4}$$

Здесь $r = m2^{-\varsigma+1}$. Так как для остаточного члена выполняется неравенство [7]

$$|R_\varsigma(r)| < 2\frac{|\beta_\varsigma|^{r+1}}{(r+1)!2^{2^\varsigma(r+1)}},$$

то $|R_\varsigma(r)| < 2^{-m}$. Величины ξ_ς получаются из формул

$$\xi_\varsigma = \frac{a_\varsigma}{b_\varsigma}, \quad a_\varsigma = \xi_\varsigma b_\varsigma, \quad b_\varsigma = r!2^{2^\varsigma r},$$

в которых целые a_ς вычисляются с помощью некоторого последовательного процесса группировки членов ряда (4). Отметим, что в формуле для b_ς присутствует факториал r, а r меняется от $m2^{-1}$ до $m2^{-k}$. Следовательно, здесь фигурирует величина, пропорциональная $n!$, и поэтому длина промежуточных данных вычислений имеет порядок $n\log(n)$.

5. Модификация метода двоичного деления для ряда Тейлора логарифмической функции

Рассмотрим ряд Тейлора логарифмической функции [16]:

$$\ln(1+x) = \sum_{i=1}^{\infty} (-1)^{(i-1)}\frac{x^i}{i} = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots + (-1)^{(n-1)}\frac{x^n}{n} + r_n(x). \tag{5}$$

Этот ряд сходится для значений x на полуинтервале $(-1, 1]$. Остаточный член ряда (5) для $|x| < 1$ в форме Коши имеет вид [16]

$$|r_n(x)| \leqslant \frac{|x|^{n+1}}{1-|x|}\left(\frac{1-\theta}{1+\theta x}\right)^n, \text{ где } 0 < \theta < 1.$$

При $x \geqslant 0$ имеем $1 + \theta x > 1 - \theta$, и последний множитель меньше единицы. Поэтому для остаточного члена ряда на полуинтервале $[0, 1)$ получаем следующую оценку:

$$|r_n(x)| \leqslant \frac{|x|^{n+1}}{1 - |x|}.$$

Пусть $m = 2^{k+1}$, $m \geqslant 2^5$, где k — натуральное число. Будем вычислять ряд (5) с точностью 2^{-m} в точке $\gamma^{(\varsigma)} = \beta^{(\varsigma)} 2^{-\xi}$, где $\xi = 2^\varsigma - 1$:

$$\ln\left(1 + \gamma^{(\varsigma)}\right) = \frac{\beta^{(\varsigma)}}{2^\xi} - \frac{\left(\beta^{(\varsigma)}\right)^2}{2 \cdot 2^{\xi 2}} + \ldots \pm \frac{\left(\beta^{(\varsigma)}\right)^r}{r \cdot 2^{\xi r}} + R^{(\varsigma)}(r), \tag{6}$$

где ς и r — натуральные числа, такие, что $\varsigma \geqslant 1$, $r = m2^{(-\varsigma+2)}$, $\beta^{(\varsigma)}$ — $2^{(\varsigma-1)}$-значное целое число. Ряд (6) линейно сходится относительно m, так как для остаточного члена на полуинтервале $[0, 1)$ выполняется следующая оценка:

$$|R^{(\varsigma)}(r)| \leqslant 2\frac{\left|\beta^{(\varsigma)}\right|^{r+1}}{2^{\xi(r+1)}} < 2\frac{2^{2^{(\varsigma-1)}(r+1)}}{2^{\xi(r+1)}} = \frac{2}{2^{\xi m 2^{(-\varsigma+2)}}} = \frac{2}{2^{2m}} \leqslant 2^{-(m+1)},$$

то есть сходимость ряда (6) не зависит от ς.

Модифицируем метод двоичного деления для частичной суммы ряда (6) так, чтобы вычисления находились в пределах класса Sch(FQLIN-TIME//LIN-SPACE).

Возьмём $k_1 = \log(r)$, $r_1 = \lceil r/k_1 \rceil$ (k_1 — натуральное число) и запишем частичную сумму ряда (6) в виде

$$P\left(\gamma^{(\varsigma)}\right) = \sigma_1 + \tau_2[\sigma_2 + \tau_3[\sigma_3 + \ldots + \tau_{k_1-1}[\sigma_{k_1-1} + \tau_{k_1}\sigma_{k_1}]]], \tag{7}$$

где

$$\begin{aligned}
\sigma_1 &= \frac{\beta^{(\varsigma)}}{2^\xi} - \frac{\left(\beta^{(\varsigma)}\right)^2}{2 \cdot 2^{\xi 2}} + \ldots \pm \frac{\left(\beta^{(\varsigma)}\right)^{r_1-1}}{(r_1-1)2^{\xi(r_1-1)}},\\
\tau_2 &= \frac{\left(\beta^{(\varsigma)}\right)^{r_1}}{2^{\xi r_1}},\\
\sigma_2 &= \pm\frac{1}{r_1} \mp \frac{\beta^{(\varsigma)}}{(r_1+1)2^\xi} \pm \frac{\left(\beta^{(\varsigma)}\right)^2}{(r_1+2)2^{\xi 2}} \mp \ldots \pm \frac{\left(\beta^{(\varsigma)}\right)^{r_1-1}}{(2r_1-1)2^{\xi(r_1-1)}},\\
\tau_3 &= \tau_2 = \frac{\left(\beta^{(\varsigma)}\right)^{r_1}}{2^{\xi r_1}},\\
\sigma_3 &= \pm\frac{1}{2r_1} \mp \frac{\beta^{(\varsigma)}}{(2r_1+1)2^\xi} \pm \frac{\left(\beta^{(\varsigma)}\right)^2}{(2r_1+2)2^{\xi 2}} \mp \ldots \pm \frac{\left(\beta^{(\varsigma)}\right)^{r_1-1}}{(3r_1-1)2^{\xi(r_1-1)}},\\
&\ldots
\end{aligned}$$

Величины σ_t будем вычислять классическим методом двоичного деления для суммы (2), где

$$\begin{aligned}
&\mu(k) = r_1 - 1;\\
&a(i) = \pm 1, \quad b(i) = \begin{cases} (1+i), & t = 0,\\ ((t-1)r_1 + i), & t > 0; \end{cases}\\
&p(j) = \begin{cases} 1, & j = 0,\\ \beta^{(\varsigma)}, & j \neq 0; \end{cases} \quad q(j) = \begin{cases} 1, & j = 0,\\ 2^\xi, & j > 0. \end{cases}
\end{aligned} \tag{8}$$

Сформулируем утверждения об оценке вычислительной сложности реализации расчёта σ_t и τ_t в виде двух лемм.

Лемма 1. Временная сложность алгоритма двоичного деления для вычисления σ_t на машине Шёнхаге ограничена сверху $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m))$; емкостная сложность — $\mathrm{O}(m)$.

Лемма 2. Временная сложность алгоритма двоичного деления для вычисления τ_t на машине Шёнхаге ограничена сверху $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m))$; емкостная сложность — $\mathrm{O}(m)$.

Доказательство лемм 1 и 2 аналогично доказательству подобных лемм из [2, 3].

Рассчитывать приближённые значения $P\left(\gamma^{(\varsigma)}\right)^*$ с точностью $2^{-(m+1)}$ по формуле (7) будем в соответствии со следующим итеративным процессом:

$$\begin{aligned} &h_1(m_1) = \sigma^*_{k_1},\\ &\widehat{h}_i(m_1) = \sigma^*_{k_1-i+1} + \tau^*_{k_1-i+2}h_{i-1}, \quad i = 1,\ldots,k_1,\\ &h_i(m_1) = \widehat{h}_i(m_1) + \varepsilon_i; \end{aligned} \tag{9}$$

при $i = k_1$ полагаем $P\left(\gamma^{(\varsigma)}\right)^* = h_{k_1}(m_1)$. Здесь $m_1 \geqslant m$, σ^*_i — приближения σ_i с точностью 2^{-m_1}, τ^*_i — приближения τ_i с точностью 2^{-m_1}. Величины $h_i(m_1)$ получаются отбрасыванием битов $q_{m_1+1}q_{m_1+2}\ldots q_{m_1+j}$ чисел $\widehat{h}_i(m_1)$ после двоичной точки, начиная с (m_1+1)-го, то есть

$$|\varepsilon_i| = |h_i(m_1) - \widehat{h}_i(m_1)| = 0{,}0\ldots0q_{m_1+1}q_{m_1+2}\ldots q_{m_1+j}, \tag{10}$$

а знак ε_i совпадает со знаком $\widehat{h}_i(m_1)$ (ясно, что $|\varepsilon_i| < 2^{-m_1}$).

Лемма 3. Для любого $i \in \{1,\ldots,k_1\}$ справедлива оценка

$$|h_i(m_1)| < 4. \tag{11}$$

Доказательство. Применим математическую индукцию по j для $h_j(m_1)$, используя оценки

$$|\sigma_i| < 1, \quad |\tau_i| = \left(\gamma^{(\nu)}\right)^{r_1} < \frac{1}{2}.$$

База индукции при j, равном 1: $|h_1(m_1)| \leqslant \sigma_{k_1} + 2^{-m_1} < 2$. Индукционный переход для $(j+1) \geqslant 2$:

$$|h_{j+1}(m_1)| = |\sigma^*_{k_1-(j+1)+1} + \tau^*_{k_1-(j+1)+2}h_j + \varepsilon_{j+1}| < 1 + \left[\frac{1}{2} + 2^{-m_1}\right]4 + 2^{-m_1} < 4.$$

Лемма доказана. ■

Лемма 4. Погрешность вычисления $h_{k_1}(m_1)$ по схеме (9) оценивается как

$$\Delta(k_1, m_1) < 2^{-m_1+k_1+3}.$$

Доказательство. Обозначим

$$H_1 = \sigma_{k_1}, \quad H_i = \sigma_{k_1-i+1} + \tau_{k_1-i+2}H_{i-1}, \quad \eta(i, m_1) = |h_i(m_1) - H_i|.$$

Воспользуемся методом математической индукции для $\eta(j, m_1)$ по j. База индукции при j, равном 1:

$$\eta(1, m_1) = |h_1(m_1) - H_1| = |\sigma^*_{k_1} - \sigma_{k_1}| < 2^{-m_1+4}.$$

Индукционный переход для $(j+1) \geqslant 2$:

$$\eta(j+1, m_1) = |\sigma^*_{k_1-(j+)+1} + \tau^*_{k_1-(j+1)+2} h_j(m_1) + \varepsilon_{j+1} - \sigma_{k_1-(j+1)+1} - \tau_{k_1-(j+1)+2} H_j| <$$
$$< |\tau^*_v h_j(m_1) - \tau_v h_j(m_1) + \tau_v h_j(m_1) - \tau_v H_j| + 2 \cdot 2^{-m_1} \leqslant 2^{-m_1} h_j(m_1) + 2^{-1}\eta(j, m_1) + 2 \cdot 2^{-m_1}.$$

Так как из (11) $|h_j(m_1)| < 4$ и, по индукционному предположению, $\eta(j, m_1) < 2^{-m_1+j+3}$, то

$$\eta(j+1, m_1) < 4 \cdot 2^{-m_1} + 2^{-1} 2^{-m_1+j+3} + 2 \cdot 2^{-m_1} < 2^{-m_1+(j+1)+3}.$$

Из $\Delta(k_1, m_1) = \eta(k_1, m_1)$ получаем искомое неравенство. ■

Из леммы 4 следует, что достаточно взять $m_1 = 2m + 3$, чтобы вычислять $P\left(\gamma^{(\varsigma)}\right)$ с точностью $2^{-(m+1)}$.

Обозначим алгоритм расчета частичной суммы ряда (6), использующий схему (9), через *ModifBinSplit* (modified binary splitting, алгоритм 3).

Алгоритм 3. *ModifBinSplit*. Приближенное значение ряда Тейлора логарифмической функции

Вход: Запись 0^m точности вычисления 2^{-m}
Выход: Приближённое значение ряда (6) с точностью 2^{-m}

1: $m_1 := 2m + 3$
2: $h := \sigma^*_{k_1}$ (с помощью обычного алгоритма двоичного деления с точностью 2^{-m_1})
3: **Для всех** $i = 2, 3, \ldots, k_1$
4: рассчитываем $a := \sigma^*_{k_1-i+1}$ с точностью 2^{-m_1} с помощью обычного алгоритма двоичного деления и $b := \tau^*_{k_1-i+2}$ с точностью 2^{-m_1};
5: вычисляем выражение $\widehat{h} := a + b \cdot h$;
6: h присваиваем величину $\widehat{h}$, округленную в соответствии с (10);
7: на выход записываем h.

Оценим временную вычислительную сложность алгоритма 3 при расчёте на машине Шёнхаге:

— $\mathrm{O}(\log(m))$ вычислений σ_t дают $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m)^2)$;
— $\mathrm{O}(\log(m))$ вычислений τ_t дают $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m)^2)$;
— $\mathrm{O}(\log(m))$ умножений чисел длины $\mathrm{O}(m)$ дают $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m))$;

итого получаем $\mathrm{O}(M_{\mathrm{Sch}}(m)\log(m)^2)$. Емкостная сложность модифицированного алгоритма двоичного деления *ModifBinSplit* — это $\mathrm{O}(m)$, так как во всех вычислениях в данном алгоритме фигурируют числа длины $\mathrm{O}(m)$.

Утверждение 1. Модифицированный алгоритм *ModifBinSplit* двоичного деления для вычисления ряда Тейлора логарифмической функции принадлежит классу алгоритмов Sch(FQLIN-TIME//LIN-SPACE).

6. Быстрое вычисление функции $\ln(1+x)$

Пусть $u = 1 + x$, $\sigma^{(x)}(5)$ и $\sigma^{(u)}(5)$ — полуинтервалы $[2^{-5}, 1 - 2^{-5})$ и $[1 + 2^{-5}, 2 - 2^{-5})$ на вещественной прямой соответственно.

Построим метод вычисления функции $\ln(1 + x)$ на полуинтервале $\sigma^{(x)}(5)$, аналогичный быстрому методу вычисления экспоненциальной функции, и назовём его, по аналогии с методом быстрого вычисления экспоненты, быстрым методом вычисления логарифмической функции.

Будем вычислять приближённое значение $\ln(u_0)^*$ с точностью 2^{-n}, где $u_0 = 1 + x_0$, $u_0 \in \sigma^{(u)}(5)$, $n \geqslant 2^4$. Возьмём натуральные k и m, такие, что

$$k \geqslant \min\{j : n \leqslant 2^j\},\ m = 2^{k+1}. \tag{12}$$

Пусть u_m — двоично-рациональное приближение аргумента u_0 с точностью 2^{-m}, то есть $|u_m - u_0| \leqslant 2^{-m}$. Так как $u_m \in [1, 2)$, то двоично-рациональная запись числа u_m есть $u_m = 1{,}\alpha_1^{(1)}\alpha_2^{(1)} \ldots \alpha_m^{(1)}\alpha_{m+1}^{(1)}$.

Обозначим u_m через $u^{(1)}$; возьмём $v^{(1)} = 1{,}\alpha_1^{(1)}$ и вычислим $u^{(2)} = \dfrac{u^{(1)}}{v^{(1)}}$ с точностью 2^{-m}, то есть $u^{(2)} = \dfrac{u^{(1)}}{v^{(1)}} + \xi^{(2)}$, где $|\xi^{(2)}| \leqslant 2^{-m}$. Легко проверить, что первый символ в двоичной записи величины $u^{(2)}$ — это 0, то есть $u^{(2)} = 1{,}0\alpha_2^{(2)}\alpha_3^{(2)} \ldots \alpha_m^{(2)}\alpha_{m+1}^{(2)}$. В силу соотношения $\ln(u) = \ln\left(\dfrac{u}{v} \cdot v\right) = \ln\left(\dfrac{u}{v}\right) + \ln(v)$ имеем следующее равенство:

$$\ln\left(u^{(1)}\right) = \ln\left(\frac{u^{(1)}}{v^{(1)}}v^{(1)}\right) = \ln\left(u^{(2)} - \xi^{(2)}\right) + \ln\left(v^{(1)}\right) = \ln\left(1 - \frac{\xi^{(2)}}{u^{(2)}}\right) + \ln\left(u^{(2)}\right) + \ln\left(v^{(1)}\right).$$

Далее, возьмём $v^{(2)} = 1{,}0\alpha_2^{(2)}\alpha_3^{(2)}$ и поделим $u^{(2)}$ на $v^{(2)}$ с точностью 2^{-m}. В результате получим соотношение

$$\ln\left(u^{(1)}\right) = \ln\left(1 - \frac{\xi^{(2)}}{u^{(2)}}\right) + \ln\left(1 - \frac{\xi^{(3)}}{u^{(3)}}\right) + \ln\left(u^{(3)}\right) + \ln\left(v^{(1)}\right) + \ln\left(v^{(2)}\right).$$

Затем берём $v^{(3)} = 1{,}000\alpha_4^{(3)}\alpha_5^{(3)}\alpha_6^{(3)}\alpha_7^{(3)}$ и выводим аналогичную формулу для $\ln\left(u^{(1)}\right)$. Возьмём натуральное число $p = k+2$. На p-м шаге данного процесса, начиная с шага 2, получим следующую формулу для $\ln\left(u^{(1)}\right)$:

$$\ln\left(u^{(1)}\right) = \sum_{i=2}^{p} \ln\left(1 + \frac{\xi^{(i)}}{u^{(i)}}\right) + \ln\left(u^{(p)}\right) + \sum_{i=1}^{p-1} \ln\left(v^{(i)}\right) = \sum_{i=2}^{p} F_i + G + \sum_{i=1}^{p-1} H_i.$$

Оценим сверху величины F_i и G, используя неравенство $|\ln(1 + x)| < |x|$ (данное неравенство следует из свойств знакопеременных рядов [10]). Так как $|u^{(i)}| \geqslant 1$, то $\dfrac{\xi^{(i)}}{u^{(i)}} \leqslant 2^{-m}$ и $F_i < 2^{-m}$. Далее, так как $u^{(p)} = 1 + \zeta^{(p)}$, где $|\zeta^{(p)}| \leqslant 2^{-m}$, то $G < 2^{-m}$.

Получаем, что если вычислять величины H_i с точностью 2^{-m}, то погрешность вычисления $\ln\left(u^{(1)}\right)$ оценивается следующим образом:

$$\varepsilon_1(m) = \left|\ln\left(u^{(1)}\right)^* - \ln\left(u^{(1)}\right)\right| < 4p \cdot 2^{-m} = (\log(m) + 2)2^{-m+2}.$$

Если взять $m \geqslant 2n$ (что согласуется с (12)), то $\varepsilon_1(m) < 2^{-(n+2)}$. Далее, так как

$$\varepsilon_2 = \left|\ln\left(u^{(1)}\right) - \ln(u_0)\right| = \left|\ln\left(u_0 + \zeta^{(1)}\right) - \ln(u_0)\right| = \left|\ln\left(1 + \frac{\zeta^{(1)}}{u_0}\right)\right| < 2^{-m},$$

где $|\zeta^{(1)}| \leqslant 2^{-m}$, то получаем оценку погрешности вычисления функции $\ln(u)$ в точке u_0:

$$\left|\ln\left(u^{(1)}\right)^* - \ln\left(u_0\right)\right| \leqslant \varepsilon_1 + \varepsilon_2 < 2^{-(n+1)}.$$

Вычислять приближения величин H_ς, то есть величины $\ln\left(v^{(\varsigma)}\right)^*$, будем с помощью алгоритма *ModifBinSplit* для ряда (6); для этого запишем $v^{(\varsigma)}$ для $1 \leqslant \varsigma \leqslant (p-1)$ в виде $1+\gamma^{(\varsigma)}$, где

$$\begin{aligned}
&\gamma^{(\varsigma)} = \beta^{(\varsigma)} 2^{-(2^\varsigma - 1)},\\
&\beta^{(1)} = \alpha_1^{(1)},\\
&\beta^{(2)} = \alpha_2^{(2)}\alpha_3^{(2)},\\
&\beta^{(3)} = \alpha_4^{(3)}\alpha_5^{(3)}\alpha_6^{(3)}\alpha_7^{(3)},\\
&\dots
\end{aligned}$$

Здесь $\beta^{(\varsigma)}$ — $2^{(\varsigma-1)}$-значное целое число.

Алгоритм 4. *FLE* (fast logarithm evaluation). Быстрое вычисление логарифмической функции вещественного аргумента

Вход: Запись 0^n точности вычисления 2^{-n}. **Оракульные функции:** φ_x аргумента x
Выход: Приближённое значение $\ln(1+x)$ с точностью 2^{-n} на полуинтервале $[2^{-5}, 1-2^{-5})$

1: Возьмём k и m так, чтобы выполнялось (12);
2: $p := k+2$;
3: $u^{(1)} := \varphi_x(m)$;
4: $S := 0$ (S — двоично-рациональное число).
5: **Для всех** $i = 2, 3, \dots, p$
6: вычисляем $u^{(i)} = \dfrac{u^{(i-1)}}{v^{(i-1)}}$;
7: рассчитываем $H^{(i)} := \ln\left(1+\gamma^{(i)}\right)^*$ с помощью алгоритма *ModifBinSplit* с точностью 2^{-m};
8: $S := S + H^{(i)}$;
9: на выход записываем значение S, округленное до точности 2^{-n}.

Вычислительная сложность алгоритма 4 при расчёте на машине Шёнхаге следующая: временная сложность — $\mathrm{O}(M_{\mathrm{Sch}}(n)\log(n)^3)$, так как алгоритм расчёта гипергеометрических рядов *ModifBinSplit* использует $\mathrm{O}(M_{\mathrm{Sch}}(n)\log(n)^2)$ операций, а в алгоритме *FLE* используется $\mathrm{O}(\log(n))$ таких вычислений и $\mathrm{O}(\log(n))$ сложений чисел длины $\mathrm{O}(n)$; емкостная сложность — $\mathrm{O}(n)$, так как во всех вычислениях фигурируют числа такой длины.

Утверждение 2. Алгоритм *FLE* быстрого вычисления логарифмической функции вещественного аргумента принадлежит классу вычислительной сложности на машине Шёнхаге Sch(FQLIN-TIME//LIN-SPACE).

Заключение

Алгоритм *FLE* расчета логарифмической функции можно применять в информатике как основу Sch(FQLIN-TIME//LIN-SPACE)-алгоритмической функции $\ln(1+x)$,

заданной на подмножестве множества Sch(FQLIN-TIME//LIN-SPACE) алгоритмических вещественных чисел.

Отметим, что если для умножения использовать рекурсивный метод Карацубы [17] с временной сложностью $\mathrm{O}(n^{\log 3})$, то временная сложность алгоритма расчёта логарифмической функции *FLE* равна $\mathrm{O}(n^{\log 3}\log^3 n)$, то есть она полиномиальна степени меньше чем 2.

Для построения алгоритма расчёта логарифмической функции на произвольном промежутке можно использовать мультипликативную редукцию интервала [18] по аналогии с тем, как осуществляется мультипликативная редукция интервала для логарифмической функции в [3].

ЛИТЕРАТУРА

1. *Яхонтов С. В.* Вычисление гипергеометрических рядов с квазилинейной временной и линейной емкостной сложностью // Вестник Самарского государственного технического университета. Сер. Физико-математические науки. 2011. Вып. 2 (17). С. 239–249.
2. *Яхонтов С. В.* Эффективное по времени и по памяти вычисление экспоненциальной функции комплексного аргумента на машине Шёнхаге // Вестник С.-Петербург. ун-та. Сер. 10. Прикладная математика. Информатика. Процессы управления. 2011. Вып. 4. С. 97–110.
3. *Яхонтов С. В., Косовский Н. К., Косовская Т. М.* Эффективные по времени и памяти алгоритмические приближения чисел и функций: учеб. пособие. СПб., 2012. 256 с.
4. *Schonhage A., Grotefeld A. F. W, and Vetter E.* Fast algorithms. A multitape Turing machine implementation. Leipzig: Brockhaus AG, 1994. 298 p.
5. *Ko K.* Complexity theory of real functions. Boston: Birkhauser, 1991. 310 p.
6. *Карацуба Е. А.* Быстрое вычисление $\exp(x)$ // 17-я Всесоюз. школа по теории информации и её приложениям. Проблемы передачи информации. 1990. Т. 26. № 3. С. 109.
7. *Карацуба Е. А.* Быстрые вычисления трансцендентных функций // Проблемы передачи информации. 1991. Т. 27. Вып. 4. С. 76–99.
8. *Карацуба Е. А.* Fast evaluation of hypergeometric function by FEE // Proc. 3rd CMFT conference on computational methods and function theory, Nicosia, Cyprus, October 13–17, 1997. Singapore: World Scientific, 1999. Ser. Approx. Decompos. V. 11. P. 303–314.
9. *Haible B. and Papanikolaou T.* Fast multiple-presicion evaluation of series of rational numbers // Proc. of the Third Intern. Symposium on Algorithmic Number Theory, Portland, Oregon, USA. June 21–25, 1998. P. 338–350.
10. *Карацуба Е. А.* Быстрое вычисление $\zeta(3)$ // Проблемы передачи информации. 1993. Т. 29. № 1. С. 68–73.
11. *Karatsuba C. A.* Fast evaluation of Bessel functions // Integral Transforms and Special Functions. 1993. V. 1. No. 4. P. 269–276.
12. *Карацуба Е. А.* Быстрое вычисление дзета-функции Римана $\zeta(s)$ для целых значений аргумента s // Проблемы передачи информации. 1995. Т. 31. No. 4. С. 69–80.
13. *Карацуба Е. А.* Быстрое вычисление дзета-функции Гурвица и L-рядов Дирихле // Проблемы передачи информации. 1998. Т. 34. № 4. С. 342–353.
14. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с.
15. *Косовская Т. М., Косовский Н. К.* Принадлежность классу FP дважды полиномиальных паскалевидных функций над подпрограммами из FP // Компьютерные инструменты в образовании. 2010. № 3. С. 3–7.

16. *Фихтенгольц Г. М.* Курс дифференциального и интегрального исчисления. М.: Физматлит, 2003. Т. 2.
17. *Карацуба А. А., Офман Ю. П.* Умножение многозначных чисел на автоматах // Доклады АН СССР. 1962. Т. 145. № 2. С. 293–294.
18. *Muller J.-M.* Elementary functions. Algorithms and implementation. Boston: Birkhauser, 1997. 204 p.

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

DOI 10.17223/20710410/20/12

УДК 519.8

ИНВЕСТИЦИОННАЯ БУЛЕВА ЗАДАЧА МАРКОВИЦА В УСЛОВИЯХ НЕОПРЕДЕЛЁННОСТИ, МНОГОКРИТЕРИАЛЬНОСТИ И РИСКА[1]

В. А. Емеличев, Р. П. Шацов

Белорусский государственный университет, г. Минск, Беларусь

E-mail: emelichev@bsu.by, roshats@gmail.com

Получены нижняя и верхняя оценки радиуса устойчивости парето-оптимального решения многокритериального варианта задачи Марковица с минимальными критериями рисков Сэвиджа в случае, когда в пространстве портфелей задана произвольная метрика Гельдера l_p, $1 \leqslant p \leqslant \infty$, а в пространствах рисков и состояний рынка — метрика Чебышева.

Ключевые слова: *многокритериальная инвестиционная задача, парето-оптимальный портфель, критерий риска Сэвиджа, радиус устойчивости портфеля, метрика Гельдера.*

Введение

В работе [1] на основе теории Марковица сформулирована многокритериальная булева задача выбора инвестиционных портфелей с максиминимальными критериями эффективности (доходности) Вальда и паретовским принципом оптимальности. Проведён анализ устойчивости парето-оптимального портфеля, который позволит учесть неопределённость исходной информации путём указания пределов надёжности инвестиционных решений в случае, когда в пространствах портфелей и состояний рынка задана линейная метрика l_1, а в критериальном пространстве — метрика Гельдера l_p.

В настоящей работе в рамках той же модели Марковица рассматривается многокритериальная инвестиционная булева задача формирования портфеля инвестора с другими целевыми функциями, а именно с критериями минимизации рисков упущенных возможностей, которым подвергается инвестор при выборе инвестиционных проектов. При этом перманентная хрупкость и непредсказуемость состояний рынка учитывается путём использования минимаксных критериев Сэвиджа (критериев крайнего пессимизма). В результате проведённого параметрического анализа найдены нижняя и верхняя оценки радиуса устойчивости портфеля, оптимального по Парето, при условии, что в пространстве решений задана произвольная метрика Гельдера l_p, $1 \leqslant p \leqslant \infty$, а в критериальном пространстве рисков и пространстве состояний финансового рынка — чебышевская метрика l_∞. Полученные результаты обобщают некоторые ранее известные оценки.

[1]Работа поддержана грантом Белорусского республиканского фонда фундаментальных исследований № Ф11К-095.

1. Постановка задачи и определения

Рассмотрим многокритериальный дискретный вариант задачи Марковица управления инвестициями [2 – 5], основанной на диверсификации как методе снижения риска. Для этого введём ряд обозначений.

Пусть i — номер возможного состояния финансового рынка (рыночной ситуации, сценария развития), $i \in N_m = \{1, 2, \ldots, m\}$; $j \in N_n$ — номер альтернативного инвестиционного проекта (актива); $k \in N_s$ — вид риска; r_{ijk} — мера экономического риска вида $k \in N_s$, которому подвергается инвестор, выбирая проект номер $j \in N_n$ в предположении, что рынок находится в состоянии номер $i \in N_m$; $R = [r_{ijk}] \in \mathbb{R}^{m\times n\times s}$; $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{E}^n$ — инвестиционный портфель, где $\mathbb{E} = \{0, 1\}$, $x_j = 1$, если инвестор выбирает проект с номером j, $x_j = 0$ в противном случае; $X \subset \mathbb{E}^n$ — множество всех допустимых инвестиционных портфелей, т. е. тех, реализация которых обеспечивает инвестору ожидаемый суммарный доход и не превосходит имеющегося у него начального капитала.

Фактор риска — неотъемлемый атрибут функционирования финансового рынка. Методам количественной оценки экономических рисков, их классификации и характеристике посвящена обширная литература (см., например, [6 – 9]). Последнее время эксперты предлагают осуществлять квантификацию рисков через призму пяти «R»: Robustness (прочность), Redundancy (избыточность), Resourcefulness (изобретательность), Response (реагирование) и Recovery (восстановление). Разнообразие видов рисков приводит к постановке многокритериальной инвестиционной задачи.

Пусть эффективность выбираемого портфеля (булева вектора) $x \in X$, $|X| \geqslant 2$, оценивается векторной целевой функцией

$$f(x, R) = (f_1(x, R_1), f_2(x, R_2), \ldots, f_s(x, R_s)),$$

компонентами которой являются минимаксимальные критерии рисков Сэвиджа [10]

$$f_k(x, R_k) = \max_{i\in N_m} R_{ik}x = \max_{i\in N_m} \sum_{j\in N_n} r_{ijk}x_j \to \min_{x\in X}, \qquad k \in N_s,$$

где $R_k \in \mathbb{R}^{m\times n}$ — k-е сечение матрицы R со строками $R_{ik} = (r_{i1k}, r_{i2k}, \ldots, r_{ink}) \in \mathbb{R}^n$, $i \in N_m$.

Тем самым, следуя критерию Сэвиджа (критерию «узкого места»), инвестор в условиях экономической нестабильности проявляет крайнюю осторожность, оптимизируя суммарный риск портфеля в самом невыгодном для него состоянии, а именно когда риск максимальный. Подобная осторожность уместна, поскольку инвестиции — это обмен определённой сегодняшней стоимости на возможно неопределённый будущий доход. Очевидно, что такой подход может быть продиктован лишь крайним пессимизмом и его использование целесообразно только тогда, когда речь идет о необходимости достижения гарантированного результата. Отметим, что в теории оптимизации задачи с минимаксимальными и максиминимальными критериями занимают видные места [11 – 14]. К таким задачам, в частности, относится и задача о наилучшем равномерном приближении функций многочленами, поставленная П. Л. Чебышевым.

Под векторной (s-критериальной) инвестиционной булевой задачей $Z^s(R)$, $s \in \mathbb{N}$, с критериями Сэвиджа будем понимать задачу поиска множества парето-оптимальных (эффективных) инвестиционных портфелей (множества Парето)

$$P^s(R) = \{x \in X : \nexists x' \in X \ \ (g(x, x', R) \geqslant 0_{(s)} \ \ \& \ \ g(x, x', R) \neq 0_{(s)})\},$$

где

$$g(x,x',R)=(g_1(x,x',R_1),g_2(x,x',R_2),\ \ldots,g_s(x,x',r_s)),$$
$$g_k(x,x',R_k)=f_k(x,R_k)-f_k(x',R_k)=\min_{i'\in N_m}\max_{i\in N_m}(R_{ik}x-R_{i'k}x'),\quad k\in N_s,$$
$$0_{(s)}=(0,0,\ldots,0)\in\mathbb{R}^s.$$

Легко видеть, что в частном случае при $m=1$ векторная инвестиционная задача $Z^s(R)$ превращается в s-критериальную задачу линейного булева программирования.

Для всякого натурального числа d в действительном пространстве $\mathbb{R}^d$ определим метрику Гельдера l_p, $1\leqslant p<\infty$, т. е. нормой вектора $a=(a_1,a_2,\ldots,a_d)\in\mathbb{R}^d$ считаем число

$$\|a\|_p=\left(\sum_{j\in N_d}|a_j|^p\right)^{1/p}.$$

Будем также использовать метрику Чебышева l_∞:

$$\|a\|_\infty=\max\{|a_j|:\ j\in N_d\}.$$

Известно, что метрика l_p, заданная в пространстве $\mathbb{R}^d$, порождает метрику l_q в сопряжённом пространстве $(\mathbb{R}^d)^*$, причём числа p и q связаны условиями

$$\frac{1}{p}+\frac{1}{q}=1,\qquad 1<p<\infty. \tag{1}$$

Как обычно, полагаем $q=1$, если $p=\infty$, и $q=\infty$, если $p=1$. Поэтому в дальнейшем считаем, что областью изменений чисел p и q является интервал $[1,\infty]$, а сами числа связаны условием (1).

Легко видеть, что из (1) вытекает равенство

$$\|z\|_p\|z\|_q=\|z\|_1\ \text{ при }\ z\in\{-1,0,1\}^n,\ p\in[1,\infty]. \tag{2}$$

В пространстве портфелей $\mathbb{R}^n$ зададим метрику l_p, $p\in[1,\infty]$, а в пространстве состояний рынка $\mathbb{R}^m$ и критериальном пространстве рисков $\mathbb{R}^s$ — чебышевскую метрику l_∞. Тем самым под нормой k-го сечения $R_k\in\mathbb{R}^{m\times n}$ матрицы $R=[r_{ijk}]\in\mathbb{R}^{m\times n\times s}$ будем понимать число

$$\|R_k\|_{p\infty}=\|(\|R_{1k}\|_p,\|R_{2k}\|_p,\ldots,\|R_{mk}\|_p)\|_\infty,\quad k\in N_s,$$

а под нормой самой матрицы R — число

$$\|R\|_{p\infty\infty}=\|(\|R_1\|_{p\infty},\|R_2\|_{p\infty},\ldots,\|R_s\|_{p\infty})\|_\infty=\max_{k\in N_s}\|R_k\|_{p\infty}.$$

Очевидно, что

$$\|R_{ik}\|_p\leqslant\|R_k\|_{p\infty}\leqslant\|R\|_{p\infty\infty},\quad i\in N_m,\ k\in N_s.$$

Кроме того, в силу известного неравенства Гельдера

$$ab\leqslant\|a\|_p\|b\|_q,$$

где $a=(a_1,a_2,\ldots,a_s)\in\mathbb{R}^s$, $b=(b_1,b_2,\ldots,b_s)^T\in\mathbb{R}^s$, для любых $x,x^0\in X$ получаем

$$\begin{aligned}R_{ik}x-R_{i'k}x^0&\geqslant-(\|R_{ik}\|_p\|x\|_q+\|R_{i'k}\|_p\|x^0\|_q)\geqslant\\&\geqslant-\|R_k\|_{p\infty}(\|x\|_q+\|x^0\|_q),i,i'\in N_m,\ k\in N_s.\end{aligned} \tag{3}$$

Следуя [1, 15 – 19], радиусом устойчивости парето-оптимального инвестиционного портфеля $x^0 \in P^s(R)$ к изменениям параметров задачи $Z^s(R)$ назовём число

$$\rho = \rho^s(x^0, p, m) = \begin{cases} \sup\ \Xi_p, & \text{если } \Xi_p \neq \varnothing, \\ 0, & \text{если } \Xi_p = \varnothing, \end{cases}$$

где

$$\Xi_p = \{\varepsilon > 0 :\ \forall R' \in \Omega_p(\varepsilon) \quad (x^0 \in P^s(R + R'))\};$$
$$\Omega_p(\varepsilon) = \{R' \in \mathbb{R}^{m\times n\times s} :\ \|R'\|_{p\infty\infty} < \varepsilon\}$$

— множество возмущающих матриц; $P^s(R + R')$ — множество Парето возмущённой задачи $Z^s(R + R')$. Тем самым радиус устойчивости задаёт предельный уровень возмущений исходных данных задачи (элементов матрицы R), сохраняющих парето-оптимальность портфеля.

Методом от противного легко доказать следующую лемму.

Лемма 1. Пусть $x^0 \in P^s(R)$, $\gamma > 0$ и $1 \leqslant p \leqslant \infty$. Если при любом портфеле $x \in X \setminus \{x^0\}$ и каждой возмущающей матрице $R' \in \Omega_p(\gamma)$ найдётся такой индекс $l \in N_s$, что справедливо неравенство $g_l(x, x^0, R_l + R'_l) > 0$, то x^0 является парето-оптимальным портфелем любой возмущённой задачи $Z^s(R + R')$, т. е. $x^0 \in P^s(R + R')$ при $R' \in \Omega_p(\gamma)$.

2. Основной результат

Введём в рассмотрение оператор проектирования вектора $a = (a_1, a_2, \ldots, a_s) \in \mathbb{R}^s$ на неотрицательный ортант:

$$[a]^+ = (a_1^+, a_2^+, \ldots, a_s^+),$$

где знак $+$ над вектором означает положительную срезку этого вектора, т. е. $a_k^+ = \max\{0, a_k\}$, $k \in N_s$.

Для портфеля x^0 задачи $Z^s(R)$ введём обозначения

$$\varphi = \varphi^s(x^0, p, m) = \min_{x\in X\setminus\{x^0\}} \frac{\|[g(x, x^0, R)]^+\|_\infty}{\|x\|_q + \|x^0\|_q},$$
$$\psi = \psi^s(x^0, p, m) = \min_{x\in X\setminus\{x^0\}} \frac{\|[g(x, x^0, R)]^+\|_\infty}{\|x - x^0\|_q}.$$

Очевидно, что $\psi \geqslant \varphi \geqslant 0$.

Теорема 1. При любых $m, s \in \mathbb{N}$ и $1 \leqslant p \leqslant \infty$ для радиуса устойчивости $\rho^s(x^0, p, m)$ парето-оптимального инвестиционного портфеля $x^0 \in P^s(R)$ задачи $Z^s(R)$ справедливы следующие оценки:

$$\varphi^s(x^0, p, m) \leqslant \rho^s(x^0, p, m) \leqslant \psi^s(x^0, p, m).$$

Доказательство. Пусть $x^0 \in P^s(R)$. Сначала докажем справедливость неравенства $\rho \geqslant \varphi$, которое очевидно при $\varphi = 0$. Пусть $\varphi > 0$. Согласно определению числа φ, для любого портфеля $x \in X \setminus \{x^0\}$ верно неравенство

$$\|[g(x, x^0, R)]^+\|_\infty \geqslant \varphi(\|x\|_q + \|x^0\|_q). \tag{4}$$

Далее методом от противного докажем, что справедлива формула

$$\forall R' \in \Omega_p(\varphi) \;\; \exists l \in N_s \quad (g_l(x, x^0, R'_l) > 0).$$

Пусть, напротив, существует такая возмущающая матрица $R^0 \in \Omega_p(\varphi)$ с сечениями R^0_k, $k \in N_s$, что $g_k(x, x^0, R_k + R^0_k) \leqslant 0$, $k \in N_s$. Тогда, согласно неравенствам (3), для любого индекса $k \in N_s$ получаем

$$0 \geqslant g_k(x, x^0, R_k + R^0_k) = \max_{i \in N_m}(R_{ik} + R^0_{ik})x - \max_{i \in N_m}(R_{ik} + R^0_{ik})x^0 =$$
$$= \min_{i' \in N_m} \max_{i \in N_m}(R_{ik}x - R_{i'k}x^0 + R^0_{ik}x - R^0_{i'k}x^0) \geqslant g_k(x, x^0, R_k) - \|R^0_k\|_{p\infty}(\|x\|_q + \|x^0\|_q) \geqslant$$
$$\geqslant g_k(x, x^0, R_k) - \|R^0\|_{p\infty\infty}(\|x\|_q + \|x^0\|_q) > g_k(x, x^0, R_k) - \varphi(\|x\|_q + \|x^0\|_q).$$

Отсюда выводим неравенство $\|[g(x, x^0, R)]^+\|_\infty < \varphi(\|x\|_q + \|x^0\|_q)$, которое противоречит (4).

Наконец, применяя лемму 1, убеждаемся, что портфель $x^0 \in P^s(R + R')$ при любой возмущающей матрице $R' \in \Omega_p(\varphi)$. Следовательно, $\rho \geqslant \varphi$.

Теперь докажем неравенство $\rho \leqslant \psi$. Согласно определению числа $\psi > 0$, существует такой портфель $x^* \in X \setminus \{x^0\}$, что справедливы соотношения

$$g_k(x^*, x^0, R_k) \leqslant [g_k(x^*, x^0, R_k)]^+ \leqslant \|[g(x^*, x^0, R)]^+\|_\infty = \psi\|x^* - x^0\|_q, \qquad k \in N_s. \tag{5}$$

Полагая $\varepsilon > \psi$, рассмотрим возмущающую матрицу $R^0 = [r^0_{ijk}] \in \mathbb{R}^{m \times n \times s}$, элементы которой зададим формулой

$$r^0_{ijk} = \delta \, \frac{x^0_j - x^*_j}{\|x^* - x^0\|_p}, \quad i \in N_m, \;\; j \in N_n, \;\; k \in N_s,$$

где $\psi < \delta < \varepsilon$. Так как все строки R^0_{ik}, $i \in N_m$, любого сечения $R^0_k \in \mathbb{R}^{m \times n}$, $k \in N_s$, одинаковы, то, обозначив такую строку через A, имеем

$$A = \delta \frac{(x^0 - x^*)^T}{\|x^* - x^0\|_p}. \tag{6}$$

Поэтому $\|R^0\|_{p\infty\infty} = \|R^0_k\|_{p\infty} = \|R^0_{ik}\|_p = \|A\|_p = \delta$, $i \in N_m$, $k \in N_s$, и, следовательно, $R^0 \in \Omega_p(\varepsilon)$ для любого $\varepsilon > \delta$.

Далее, благодаря (2) и (6), для любого числа $p \in [1, \infty]$ справедлива цепочка равенств

$$A(x^* - x^0) = -\delta \frac{\|x^* - x^0\|_1}{\|x^* - x^0\|_p} = -\delta\|x^* - x^0\|_q.$$

Наконец, используя эти равенства и соотношения (5), заключаем, что для любого индекса $k \in N_s$ справедливы соотношения

$$g_k(x^*, x^0, R_k + R^0_k) = \max_{i \in N_m}(R_{ik} + A)x^* - \max_{i \in N_m}(R_{ik} + A)x^0 = g_k(x^*, x^0, R_k) + A(x^* - x^0) =$$
$$= g_k(x^*, x^0, R_k) - \delta\|x^* - x^0\|_q < g_k(x^*, x^0, R_k) - \psi\|x^* - x^0\|_q \leqslant 0.$$

Поэтому $x^0 \notin P^s(R + R^0)$. Следовательно, $\rho \leqslant \psi$. ■

При $m = 1$, как уже отмечалось выше, задача $Z^s(R)$ превращается в s-критериальную задачу булева программирования с линейными критериями. Запишем её в удобном виде:

$$Z^s_B(R): \quad Rx \to \min_{x \in X},$$

где $X \subseteq \mathbb{R}^n$; $R = [r_{kj}] \in \mathbb{R}^{s\times n}$ — матрица со строками $R_k = (r_{k1}, r_{k2}, \ldots, r_{kn}) \in \mathbb{R}^n$, $k \in N_s$. Такой случай ($m = 1$) можно интерпретировать как ситуацию, при которой состояние финансового рынка стабильно и не вызывает сомнений. В этом контексте, как и прежде, считаем, что в пространстве портфелей $\mathbb{R}^n$ задана метрика Гельдера, а в критериальном пространстве рисков $\mathbb{R}^s$ — метрика Чебышева.

Следующий известный результат свидетельствует о том, что верхняя оценка радиуса устойчивости инвестиционного портфеля $x^0 \in P^s(R)$ задачи $Z^s(R)$ является достижимой при $m = 1$.

Теорема 2 [19]. Для радиуса устойчивости инвестиционного портфеля $x^0 \in P^s(R)$ задачи линейного булева программирования $Z^s_B(R)$, $R \in \mathbb{R}^{s\times n}$, при любых $p \in [1, \infty]$ и $s \in \mathbb{N}$ справедлива формула

$$\rho^s(x^0, p, 1) = \psi^s(x^0, p, 1) = \min_{x\in X\setminus\{x^0\}} \frac{\|[R(x - x^0)]^+\|_\infty}{\|x - x^0\|_q}.$$

3. Следствия

Все приведенные ниже следствия, вытекающие из теоремы 1, очевидны и верны при любом числе критериев $s \in \mathbb{N}$.

Следствие 1 [20]. При любом числе $m \in \mathbb{N}$ справедливы оценки

$$\min_{x\in X\setminus\{x^0\}} \frac{\|[g(x, x^0, R)]^+\|_\infty}{\|x + x^0\|_1} \leqslant \rho^s(x^0, \infty, m) \leqslant \min_{x\in X\setminus\{x^0\}} \frac{\|[g(x, x^0, R)]^+\|_\infty}{\|x - x^0\|_1}. \tag{7}$$

Отметим, что в [20] построены конкретные классы многокритериальных инвестиционных задач с критериями Сэвиджа, для которых эти оценки достигаются, т. е. становятся формулами. О достижимости оценок (7) свидетельствует также следующее утверждение.

Следствие 2. Если для всякого инвестиционного портфеля $x \neq x^0$ множество $\{j \in N_n : \ x^0_j = x_j = 1\}$ пусто, то при любом $m \in \mathbb{N}$ справедлива формула

$$\rho^s(x^0, \infty, m) = \varphi^s(x^0, \infty, m) = \psi^s(x^0, \infty, m).$$

Следствие 3 [21]. При любом числе $m \in \mathbb{N}$ справедливы следующие оценки:

$$\frac{1}{2} \min_{x\in X\setminus\{x^0\}} \|[g(x, x^0, R)]^+\|_\infty \leqslant \rho^s(x^0, 1, m) \leqslant \min_{x\in X\setminus\{x^0\}} \|[g(x, x^0, R)]^+\|_\infty.$$

В [21] доказана достижимость этих оценок путём построения соответствующих классов задач, для которых эти оценки превращаются в формулы.

Заключение

Желания инвестора минимизировать различные виды рисков при выборе инвестиционного портфеля служат веским основанием необходимости применения многокритериального подхода к экономико-математическому моделированию рисков. Этот подход позволяет использовать методы теории выбора и принятия решений в условиях многокритериальности [22, 23]. В данной работе при построении многокритериальной модели управления инвестиционными рисками использованы целевые функции «узкого места» Сэвиджа, которые в условиях нестабильности и неопределённости финансового рынка позволяют выбирать тот портфель, при котором величина суммарного

риска принимает минимальное значение в самой неблагополучной ситуации, а именно тогда, когда риск максимален. Другой тип неопределённости появляется в связи с применением в качестве исходной информации статистических и экспертных оценок различных видов рисков, точность и достоверность которых всегда вызывает сомнение. В этой связи возникает потребность в проведении постоптимального анализа, состоящего в выявлении радиуса устойчивости, т. е. предельного уровня изменений (возмущений) параметров исходной задачи, сохраняющих оптимальность выбранного портфеля. В работе получены нижняя и верхняя оценки радиуса устойчивости парето-оптимального портфеля в случае произвольной метрики Гельдера, заданной в пространстве решений задачи. Результаты могут быть использованы при конструировании оптимальных портфелей инвестора, желающего учитывать неопределённость различных видов рисков при наличии возможных (прогнозных) состояний финансового рынка.

ЛИТЕРАТУРА

1. *Емеличев В. А., Коротков В. В.* Исследование устойчивости решений векторной инвестиционной булевой задачи в случае метрики Гельдера в критериальном пространстве // Прикладная дискретная математика. 2012. № 4. С. 61–72.
2. *Markowitz H.* Portfolio selection // J. Finance. 1952. V. 7. No. 1. P. 77–91.
3. *Markowitz H. M.* Portfolio selection: efficient diversification of investments. New York: Willey, 1991. 400 p.
4. *Шарп У. Ф., Александер Г. Дж., Бейли Д. В.* Инвестиции. М.: Инфра-М, 2003. 1028 с.
5. Portfolio decision analysis: improved methods for resource allocation (International Series in Operations Research and Management Science) / eds. A. Salo, J. Keisler, A. Morton. New York: Springer, 2011. 424 p.
6. *Тэпман Л. Н.* Риски в экономике. М.: ЮНИТИ-ДАНА, 2002. 380 с.
7. *Шапкин А. С.* Экономические и финансовые риски. М.: Дашков и К°, 2003. 544 с.
8. *Бронштейн Е. М., Качкаева М. М., Тулупова Е. В.* Управление портфелем ценных бумаг на основе комплексных квантильных мер риска // Известия РАН. Теория и системы управления. 2011. № 1. С. 178–183.
9. *Деревянко П. М.* Оценка проектов в условиях неопределенности // Корпоративный менеджмент [Электронный ресурс]. 2006. `http://www.cfin.ru/finanalysis/invest/fuzzy_analysis.shtml/`. Дата доступа: 06.02.13.
10. *Savage L. J.* The foundations of statistics. New York: Dover Publ., 1972. 310 p.
11. *Демьянов В. Ф., Малоземов В. Н.* Введение в минимакс. М.: Наука, 1972. 368 с.
12. *Федоров В. В.* Численные методы максимина. М.: Наука, 1979. 280 с.
13. Minimax and applications / eds. D.-Z. Du, P. M. Pardalos. Dordrecht: Kluwer Acad. Publ., 1995. 308 p.
14. *Сухарев А. Г.* Минимаксные алгоритмы в задачах численного анализа. М.: Либроком, 2009. 304 с.
15. *Емеличев В. А., Кузьмин К. Г.* О радиусе устойчивости эффективного решения векторной задачи целочисленного линейного программирования в метрике Гёльдера // Кибернетика и системный анализ. 2006. № 4. С. 175–181.
16. *Емеличев В. А., Коротков В. В., Кузьмин К. Г.* Многокритериальная инвестиционная задача в условиях неопределенности и риска // Известия РАН. Теория и системы управления. 2011. № 6. С. 157–164.

17. *Емеличев В. А., Коротков В. В.* О радиусе устойчивости эффективного решения многокритериальной задачи портфельной оптимизации с критериями Сэвиджа // Дискретная математика. 2011. Т. 23. Вып. 4. С. 33–38.
18. *Емеличев В. А., Коротков В. В.* Постоптимальный анализ векторной инвестиционной задачи с максиминными критериями Вальда // Дискретный анализ и исследование операций. 2012. Т. 19. № 6. С. 23–36.
19. *Емеличев В. А., Кузьмин К. Г.* Общий подход к исследованию устойчивости парето-оптимального решения векторной задачи целочисленного линейного программирования // Дискретная математика. 2007. Т. 19. Вып. 3. С. 79–83.
20. *Emelichev V., Korotkov V., and Kuzmin K.* On stability of a Pareto-optimal solution of a portfolio optimization problem with Savage's minimax risk criteria // Bulletin of the Academy of Sciences of Moldova. Mathematics. 2010. No. 3 (64). P. 35–44.
21. *Емеличев В. А., Коротков В. В.* Об устойчивости эффективного решения векторной инвестиционной булевой задачи с минимаксными критериями Сэвиджа // Труды Института математики НАН Беларуси. 2010. Т. 18. № 2. С. 3–10.
22. *Ларичев О. И.* Теория и методы принятия решений. М.: Логос, 2002. 392 с.
23. *Ногин В. Д.* Принятие решений в многокритериальной среде: количественный подход. М.: Физматлит, 2002. 144 с.

СВЕДЕНИЯ ОБ АВТОРАХ

АЛЕКСЕЙЧУК Антон Николаевич — доктор технических наук, профессор Института специальной связи и защиты информации Национального технического университета Украины «КПИ», г. Киев. E-mail: **alex-crypto@mail.ru**

ГОРШКОВ Сергей Павлович — доктор физико-математических наук, доцент Института криптографии, связи и информатики, г. Москва. E-mail: **spg54@bk.ru**

ГОЦУЛЕНКО Владимир Владимирович — старший научный сотрудник, кандидат технических наук, старший научный сотрудник отдела теплофизических основ энергосберегающих теплотехнологий Института технической теплофизики НАН Украины, г. Киев. E-mail: **gosul@ukr.net**

ГРЯЗНУХИН Александр Юрьевич — аспирант Института специальной связи и защиты информации Национального технического университета Украины «КПИ», г. Киев. E-mail: **x67mail@gmail.com**

ДВИНЯНИНОВ Артём Васильевич — сотрудник лаборатории ТВП, г. Москва. E-mail: **artem_msk@pisem.net**

ЕМЕЛИЧЕВ Владимир Алексеевич — профессор, доктор физико-математических наук, профессор кафедры математической кибернетики Белорусского государственного университета, г. Минск. E-mail: **emelichev@bsu.by**

ЗУБОВ Анатолий Юрьевич — кандидат физико-математических наук, доцент, старший научный сотрудник Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **Zubovanatoly@yandex.ru**

КОСТЮК Юрий Леонидович — профессор, доктор технических наук, профессор Национального исследовательского Томского государственного университета, г. Томск. E-mail: **kostyuk_y_l@sibmail.com**

МУРИН Дмитрий Михайлович — аспирант Ярославского государственного университета им. П. Г. Демидова, г. Ярославль. E-mail: **nirum87@mail.ru**

СМЫШЛЯЕВ Станислав Витальевич — кандидат физико-математических наук, Московский государственный университет им. М. В. Ломоносова, г. Москва.
E-mail: **smyshsv@gmail.com**

ТАРКОВ Михаил Сергеевич — кандидат технических наук, старший научный сотрудник Института физики полупроводников СО РАН, г. Новосибирск.
E-mail: **tarkov@isp.nsc.ru**

ЧЕРЕМУШКИН Александр Васильевич — доктор физико-математических наук, член-корреспондент Академии криптографии РФ, заведующий кафедрой Института криптографии, связи и информатики, г. Москва. E-mail: **avc238@mail.ru**

ШАЦОВ Роман Павлович — студент кафедры математической кибернетики Белорусского государственного университета, г. Минск. E-mail: **roshats@gmail.com**

ЯХОНТОВ Сергей Викторович — кандидат физико-математических наук, доцент кафедры информатики математико-механического факультета Санкт-Петербургского государственного университета, г. Санкт-Петербург.
E-mail: **SergeyV.Yakhontov@gmail.com**

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Gorshkov S. P. **ABOUT SOME PROPERTIES OF HORN AND ANTI-HORN FUNCTIONS.** Some properties of weakly positive (anti-Horn) and weakly-negative (Horn) Boolean functions are investigated. Particularly, estimates are given for the complexity of constructing reduced form and for the possible lengths of expressions of considered functions, and it is shown that there are no limits for the weight of such functions.
Keywords: *weakly positive* (*anti-Horn*) *Boolean function, weakly negative* (*Horn*) *Boolean function, computing complexity.*

Gorshkov S. P., Dvinyaninov A. V. **LOWER AND UPPER BOUNDS FOR THE AFFINITY ORDER OF TRANSFORMATIONS OF BOOLEAN VECTOR SPACES.** Let Φ_n be the set of all transformations of the Boolean vector space V_n. Affinity order of a mapping $F \in \Phi_n$ is the least order of the set V_n partition with the property: for every its block there exists an affine mapping $A : V_n \to V_n$ being equivalent to F on this block. Affinity order of Φ_n is the greatest order of $F \in \Phi_n$. Upper and lower bounds for the affinity order of Φ_n are given in the article. These results can be used for estimating complexity of some techniques in Boolean equations resolving.
Keywords: *transformation of Boolean vector space, affine mapping, solution complexity of Boolean equations.*

Smyshlyaev S. V. **CONNECTIONS BETWEEN SOME PARAMETERS OF PERFECTLY BALANCED BOOLEAN FUNCTIONS.** For the perfectly balanced Boolean functions with barriers, some relations between their parameters are proved. Particularly, general results are obtained concerning the properties of the polynomials of these functions. They can be used to get new bounds for the number of inverse functions for an arbitrary function with barrier.
Keywords: *perfectly balanced functions, functions with barriers, cryptography.*

Cheremushkin A. V. **AN ADDITIVE APPROACH TO NONLINEARITY DEGREE OF DISCRETE FUNCTIONS ON A PRIMARY CYCLIC GROUP.** An additive approach to the definition of nonlinearity degree for a discrete function on a cyclic group is proposed. For elementary abelian groups, this notion is equivalent to the ordinary "multiplicative" one. For polynomial functions on the ring of integers $\bmod p^n$, this notion is equivalent to the minimal degree of a polynomial. It is proved that the nonlinearity degree on a cyclic group is a finite number if and only if the order of the group is a power of a prime. An upper bound for the nonlinearity degree of functions on a cyclic group of order p^n is given.
Keywords: *discrete functions, nonlinearity degree.*

Zubov A. U. **AUTHENTICATION CODE WITH SECRECY BASED ON PROJECTIVE GEOMETRY.** An authentication code with the secrecy based on the projective geometry is proposed. It ensures the secure authentication and confidentiality for the equiprobable source of information.
Keywords: *authentication code with secrecy, perfect cipher, projective geometry.*

Tarkov M. S. **MAPPING PARALLEL PROGRAMS ONTO MULTICORE COMPUTERS BY RECURRENT NEURAL NETWORKS.** The problem of mapping a parallel program with weighted vertices (processes) and edges (interprocess exchanges) onto a weighted graph of the distributed computer system is considered. An algorithm for solving this problem based on the use of Hopfield networks is proposed. The algorithm has been tested on mapping a number of graphs of parallel programs onto multicore computer. Experiments have shown that the proposed algorithm provides well-balanced sub-optimal mappings.

Keywords: *graphs of parallel programs, multicore systems, neural networks, Hopfield networks.*

Alekseychuk A. N., Gryaznukhin A. Yu. **FAST ALGORITHM FOR RECOVERING THE TRUE SOLUTION WITH FIXED WEIGHT OF A SYSTEM OF LINEAR BOOLEAN EQUATIONS WITH NOISED RIGHT-HAND SIDE.** A system of linear Boolean equations with noised right-hand side is considered. It is assumed that the true solution of the system has a predetermined weight that is independent of the number of variables in the system. A probabilistic algorithm for finding this solution is proposed. The time complexity of our algorithm is less than the time complexity of maximum likelihood method, and in contrast to the known algorithms with this property, our algorithm uses only comparisons and (bitwise and arithmetic) additions of binary integers. The proposed algorithm can be used in practice in some cases when other algorithms are less efficient.

Keywords: *system of Boolean linear equations with noised right-hand side, probabilistic algorithm.*

Gotsulenko V. V. **A FORMULA FOR THE NUMBER OF COMBINATIONS WITH CONSTRAINED REPETITIONS AND ITS APPLICATION.** Various generalizations of the concept of combination with repetitions are obtained. Formulas for the calculation of the considered combinatorial numbers are found, and various problems that are solved with their application are considered.

Keywords: *combinations with constrained repetitions, Diophantine equations, formal polynomial, generating functions, multisets.*

Kostyuk Yu. L. **EFFECTIVE IMPLEMENTATION OF ALGORITHM FOR SOLVING THE TRAVELLING SALESMAN PROBLEM BY BRANCH-AND-BOUND METHOD.** The modification of Little's algorithm for solving the well-known travelling salesman problem by branch-and-bound method is proposed. At each intermediate stage of the algorithm execution, a more exact lower bound is evaluated for all variants of the route which may be built on the basis of the current partial solution. Thanks to this, the rejection of unperspective variants becomes, as a rule, much more effective, especially when applied to the random asymmetrical distance matrix. The implementations of this modified algorithm are described with the depth-first and breadth-first search, and also with the depth-first search when an approximate route with the inaccuracy prescribed arbitrarily is searched. In a computing experiment, for each algorithm implementation, the values of constants a and c have been evaluated for the complexity function $U(n) = a \cdot c^n$ that is the number of distance matrixes (decision tree nodes) processing by the algorithm. In any case the time of each node processing increases by 1,5–2 times while the time of processing the whole decision tree by the algorithm is significantly decreased.

Keywords: *travelling salesman problem, branch-and-bound method, computing experiment.*

Murin D. M. **MODIFICATION OF THE LAGARIAS — ODLYZHKO METHOD FOR SOLVING THE GENERALIZED KNAPSACK PROBLEM AND THE SYSTEMS OF KNAPSACK PROBLEMS.** In 1985, to solve the computational knapsack problem, J. C. Lagarias and A. M. Odlyzhko have offered a method based on the reduction of this problem to the search of one of the short vector in the lattice of a special kind. This way allows to solve "almost all" knapsack problems that have a low density. In this paper the Lagarias — Odlyzhko method is modified to be applicable for solving the generalized knapsack problem and the systems of generalized knapsack problems. The system of knapsack problems is introduced hear as the finite set of the individual knapsack problems that have a common solution. Besides, for the generalized knapsack problem and for the systems of knapsack problems, some sets of density values are defined so that the modified methods allow to solve "almost all" generalized knapsack problems and the systems of knapsacks problems with the densities from these sets.
Keywords: *Lagarias — Odlyzhko method, knapsack problem, generalized knapsack problem, systems of knapsack problems.*

Yakhontov S. V. **TIME- AND SPACE-EFFICIENT EVALUATION OF THE REAL LOGARITHMIC FUNCTION ON SCHONHAGE MACHINE.** In the present paper, an algorithm *FLE* is constructed for the fast evaluation of the real logarithmic function $\ln(1+x)$ on interval $[2^{-5}, 1-2^{-5})$ on Schonhage machine. An upper bound of the time and space complexity of this algorithm is given. The algorithm *FLE* is based on Taylor series expansion and is similar to the algorithm for the fast evaluation of the exponential function *FEE*. A modified binary splitting algorithm *ModifBinSplit* for hypergeometric series is constructed to use in algorithm *FLE*. It is proved that the time and space complexity of algorithms *ModifBinSplit* and *FLE* are quasi-linear and linear respectively if they are implemented on Schonhage machine; therefore it is proved that these algorithms are in class Sch(FQLIN-TIME//LIN-SPACE). Multiple interval reduction is used to compute the logarithmic function on an arbitrary interval.
Keywords: *logarithmic function, algorithmic real functions, quasi-linear time complexity, linear space complexity.*

Emelichev V. A., Shatsov R. P. **MARKOWITZ INVESTMENT BOOLEAN PROBLEM IN CASE OF UNCERTAINTY, MULTICRITERIA AND RISK.** Lower and upper bounds are obtained for the stability radius of a Pareto optimal portfolio of multicriteria variant of Markowitz problem with Savage minimax risk criteria in the case of any Hölder metric l_p, $1 \leqslant p \leqslant \infty$, in the portfolio space and Chebyshev metric in the risk and market state spaces.
Keywords: *multicriteria investment problem, Pareto optimal portfolio, Savage risk criteria, stability radius of portfolio, Hölder metric.*

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, а также в перечень журналов, рекомендованных УМО в области информационной безопасности РФ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте vestnik.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также и правила подготовки рукописей статей в журнал.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надёжности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*